
# Collaborator

## User's and Administrator's Guide

# Collaborator v13.6.13601 Manual



Printed June 2021 in Austin, TX, USA.

# Table of Contents

# 1 General Information

Collaborator is a code and document review tool that helps you deliver higher quality code to QA as well as to your customer.

## Getting Started

- Tutorial (For Developers) 42
- Video Lessons
- Performing Reviews 332

## General Information

- Overview 1
- Review Workflow and Phases 17
- Collaborator Components 13

## Using Collaborator

- Web Client 312
- Desktop Clients 484
- Version Control Integration 620

## Server Administration

- Tutorial (For Administrators) 35
- Collaborator Server 56
- Server Upgrades 83

## For Existing Users

- What's New in Collaborator 13 982
- Techniques & Best Practices 967

## Support & Resources

- Community
- Contact Support
- Additional Resources 32

## 1.1 About Collaborator

### What is Code Review?

Code Review, or Peer Code Review, is the act of consciously and systematically convening with one's fellow programmers to check each other's code for mistakes, and has been repeatedly shown to accelerate and streamline the process of software development like few other practices can.

Code Review is an integral process of software development that helps identify bugs and defects before the testing phase. Code review is often overlooked as an ongoing practice during the development phase, but countless studies show it's the most effective quality assurance strategy. Meetings end up taking more time than intentionally planned. Not having a set process in place means you don't actually know if your code reviews are effective or are even happening.

Collaborator is a code review and document review tool that helps development, testing and management teams work together to produce high quality code. It allows teams to peer review code, user stories and test plans in a transparent, collaborative framework — instantly keeping the entire team up to speed on changes made to the code. By enabling team members to work together to review their work, Collaborator can help you catch bugs before your software hits the market.

## Main features of Collaborator:

- Implements various review styles 16
- Multiple, simultaneous reviewers
- Workflow supporting reviewers/authors separated by many time zones
- Version control integration 620: Git, Perforce, Subversion, Mercurial, Team Foundation Server, ClearCase and more
- Cross-platform clients: web client 312, GUI clients 484, command-line clients 506 and IDE plug-ins
- Defect-tracking 436 with severity, type, classification, checklists, and external issue-tracker integration
- Full-featured metrics 972, reports 467, and data-export 472
- Project- and role-based rules 246 and reporting
- Command-line and web-service API 907 for integrations, extensions, automations, and triggers

## Supported formats of review materials:

- Text files 377 (including source code and any other text-based formats)
- Word processing documents 383 (.DOC, .DOCX, .RTF, .DOCM, .DOT, .DOTM, .DOTX, .ODT and .OTT)
- Spreadsheets 389 (.XLS, .XLSX, .XLSB, .XLSM, .XLTM, .XLTX and .ODS)
- Presentations 395 (.PPT, .PPTX, .POT, .POTM, .POTX, .PPS, .PPSM, .PPSX, .PPTM and .ODP)

- Visio vector graphics 400 (.VDW, .VDX, .VSD, .VSDM, .VSDX, .VSS, .VSSM, .VSSX, .VST, .VSTM, .VSTX, .VSX and .VTX)
- Simulink models 404 (web view archives)
- Images 410 (.JPG, .JPEG, .PNG, .GIF and .BMP)
- PDF documents 414 (.PDF, and any other document types converted to PDF format)
- URLs 419 (HTTP and HTTPS)

## 1.2 Collaborator Editions

Starting from version 10.0 Collaborator is distributed in three editions: Collaborator Community, Collaborator Team and Collaborator Enterprise.

**Collaborator Community** is the lightweight edition of Collaborator, designed especially for small development groups with simpler needs. Collaborator Community is offered free of charge and includes the basic features necessary for code review.

**Collaborator Team** is designed for medium development groups with moderate needs. Collaborator Team includes the majority of features necessary for code review without the extra cost of the enterprise level features found in Collaborator Enterprise.

**Collaborator Enterprise** is the full featured edition of Collaborator, designed especially for large development groups. In order to review Simulink models 404, you should have Collaborator Enterprise edition license, obtain Simulink integration licenses and assign them to particular users.

The remainder of this section will describe the differences between the three products.

### Set Up and Licensing

All three editions of Collaborator are shipped within the same installer that can be found on our downloads page. The products are differentiated only by the license codes installed. During a trial installation, the product option will be given in the First-Run Initialization 36 phase. After installation, you can easily switch from one edition to another, by changing out the license code with a new one provided by your account manager.

### Collaborator Feature Summary by Edition

The table below will help you assess the functional differences between Collaborator editions:

| System Administration | | | |
|---|---|---|---|
| **Feature** | **Collaborator Community** | **Collaborator Team** | **Collaborator Enterprise** |

| | | | |
|---|---|---|---|
| **Licenses** 90 | Fixed licenses only | Fixed licenses only | Fixed and concurrent licenses available |
| **License Model** 90 | Perpetual | Subscription | Subscription or Perpetual |
| **Database Back-Ends** | Embedded 60 | Embedded 60, MySQL 60 | Embedded 60, MySQL 60, SQL Server 64, Oracle 67 |
| **LDAP / ActiveDirectory Authentication** 119 | Not Supported | Supported | Supported |
| **Single Sign-On** 130 | Not Supported | Not Supported | Supported |
| **Number of users** | Up to 10 | Up to 25 | Unlimited - Based on Licenses Owned |
| **Technical support** | No | Yes | Yes |

| **File Type Support** | | | |
|---|---|---|---|
| **Type** | **Collaborator Community** | **Collaborator Team** | **Collaborator Enterprise** |
| **Word Processing Documents** 383 | Not Supported | Not Supported | Supported |
| **Spreadsheets** 389 | Not Supported | Not Supported | Supported |
| **Presentations** 395 | Not Supported | Not Supported | Supported |
| **Visio Documents** 400 | Not Supported | Not Supported | Supported |
| **PDF Documents** 414 | Not Supported | Not Supported | Supported |
| **Image Files** 410 | Not Supported | Supported | Supported |
| **Live URLs** 419 | Not Supported | Not Supported | Supported |

| | | | |
|---|---|---|---|
| **Simulink models** 404 | Not Supported | Not Supported | Supported (with Simulink Integration license) |

| Server Configuration Options | | | |
|---|---|---|---|
| **Features** | **Collaborator Community** | **Collaborator Team** | **Collaborator Enterprise** |
| **Subscriptions** 187 | Supported | Supported | Supported |
| **Bug Tracking Integration** 194 | Supported | Supported | Supported |
| **User Statistics** 219 | Supported | Supported | Supported |
| **Email Notifications** 199 | Supported | Supported | Supported |
| **Server-Side Version Control Integrations** 204 | Supported | Supported | Supported |
| **System Status** 212 | Supported | Supported | Supported |
| **Groups** 226 | Not Supported | Not Supported | Supported |
| **Review and Uploads Access Restrictions** 188 | Not Supported | Not Supported | Supported |
| **Review Custom Fields** 256 | Not Supported | Limited. Only 2 fields. | Unlimited |
| **Participant Custom Fields** 258 | Not Supported | Not Supported | Supported |
| **Defect Custom Fields** 266 | Not Supported | Limited. Only 2 fields. | Unlimited |
| **Multiple Role Configurations** 279 | Not Supported | Not Supported | Supported |
| **Configurable Workflow** 246 | Not Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Server-Side Triggers** 206 | Not Supported | Not Supported | Supported |
| **Customizable Notifications** 288 | Not Supported | Not Supported | Supported |
| **Automatic Links** 286 | Not Supported | Not Supported | Supported |
| **Archiving Reviews** 165 | Not Supported | Not Supported | Supported |
| **Review Checklists** 272 | Not Supported | Supported | Supported |
| **Electronic Signatures** 193 | Not Supported | Not Supported | Supported |
| **Server Branding** 168 | Not Supported | Not Supported | Supported |

| Clients and Version Control Integrations | | | |
|---|---|---|---|
| **Client/VCS** | **Collaborator Community** | **Collaborator Team** | **Collaborator Enterprise** |
| **Web GUI** 315 | Supported | Supported | Supported |
| **GUI Client** 496 | Supported | Supported | Supported |
| **External Diff Viewer Launcher** 615 | Supported | Supported | Supported |
| **Eclipse Plugin** 525 | Supported | Supported | Supported |
| **Visual Studio Plugin** 566 | Supported | Supported | Supported |
| **Command Line Client** 506 | Supported | Supported | Supported |
| **Tray Notifier** 613 | Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Version Control Integration** | Git 645, Subversion 799, GitHub, Bitbucket, GitLab 826 | CVS 635, Git 645, Mercurial 730, Perforce 764, Subversion 799, Team Foundation Server 738, GitHub, Bitbucket, GitLab 826 | CVS 635, Git 645, Mercurial 730, Perforce 764, Subversion 799, AccuRev 622, PTC Integrity 752, ClearCase 666, Rational Synergy 691, Rational Team Concert 700, Team Foundation Server 738, GitHub, Bitbucket, GitLab 826 |

## Reports

The table below compares the fields and metrics available in different editions of Collaborator.

| **Reports** | | | |
|---|---|---|---|
| **Column/Field** | **Collaborator Community** | **Collaborator Team** | **Collaborator Enterprise** |
| ***Customizable Review Reports*** | | | |
| **ID** | Not Supported | Supported | Supported |
| **Review Title** | Not Supported | Supported | Supported |
| **Review Creation Date** | Not Supported | Supported | Supported |
| **Review Completion Date** | Not Supported | Supported | Supported |
| **Workflow** | Not Supported | Supported | Supported |
| **Phase** | Not Supported | Supported | Supported |
| **Review is Private** | Not Supported | Supported | Supported |
| **Creator Login** | Not Supported | Supported | Supported |
| **Creator Full Name** | Not Supported | Supported | Supported |
| **Author Login** | Not Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Author Full Name** | Not Supported | Supported | Supported |
| **Moderator Login** | Not Supported | Supported | Supported |
| **Moderator Full Name** | Not Supported | Supported | Supported |
| **Reviewer Login** | Not Supported | Supported | Supported |
| **Reviewer Full Name** | Not Supported | Supported | Supported |
| **Observer Login** | Not Supported | Supported | Supported |
| **Observer Full Name** | Not Supported | Supported | Supported |
| **Defect Count** | Not Supported | Supported | Supported |
| **Open Defect Count** | Not Supported | Supported | Supported |
| **Overview** | Not Supported | Supported | Supported |
| **Group** | Not Supported | Supported | Supported |
| **Defects per Hour** | Not Supported | Supported | Supported |
| **Comment Count** | Not Supported | Supported | Supported |
| **Last Comment** | Not Supported | Supported | Supported |
| **Idle for** | Not Supported | Supported | Supported |
| **File Count** | Not Supported | Supported | Supported |
| **LOC** | Not Supported | Supported | Supported |
| **LOC Changed** | Not Supported | Supported | Supported |
| **LOC Added** | Not Supported | Supported | Supported |
| **LOC Removed** | Not Supported | Supported | Supported |
| **LOC Modified** | Not Supported | Supported | Supported |
| **LOC Delta** | Not Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Review Wall-Clock Duration** | Not Supported | Supported | Supported |
| **Total Person-Time** | Not Supported | Supported | Supported |
| **Reviewer Time** | Not Supported | Supported | Supported |
| **Author Time** | Not Supported | Supported | Supported |
| **Number of Participants** | Not Supported | Supported | Supported |
| **Average Participant Time** | Not Supported | Supported | Supported |
| **Custom Fields** | Not Supported | Supported | Supported |
| ***Review Detail Reports*** | | | |
| *Overview* | | | |
| **ID** | Supported | Supported | Supported |
| **Status** | Supported | Supported | Supported |
| **Title** | Supported | Supported | Supported |
| **Creator** | Supported | Supported | Supported |
| **Created On** | Supported | Supported | Supported |
| **Finished On** | Supported | Supported | Supported |
| **Wall-Clock Time** | Supported | Supported | Supported |
| **Number of Defects** | Supported | Supported | Supported |
| **Number of Comments** | Supported | Supported | Supported |
| **Overview** | Supported | Supported | Supported |
| **Total Person Time** | Not Supported | Supported | Supported |
| **Total Reviewer Time** | Not Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **LOC All Version (Uploaded/ Changed)** | Not Supported | Supported | Supported |
| **LOC Final Version (Uploaded/ Changed)** | Not Supported | Supported | Supported |
| **Document Pages (Final)** | Not Supported | Supported | Supported |
| **Custom Fields** | Not Supported | Supported | Supported |
| *Participants* | | | |
| **Name** | Supported | Supported | Supported |
| **Role** | Supported | Supported | Supported |
| **Person-Hours** | Not Supported | Supported | Supported |
| **Per Participant Custom Fields** | Not Supported | Not Supported | Supported |
| *Defect Log* | | | |
| **ID** | Not Supported | Supported | Supported |
| **Creator** | Not Supported | Supported | Supported |
| **File** | Not Supported | Supported | Supported |
| **Location** | Not Supported | Supported | Supported |
| **Text** | Not Supported | Supported | Supported |
| **Defect Custom Fields** | Not Supported | Supported | Supported |
| *Materials Summary* | | | |
| **Num Files** | Supported | Supported | Supported |
| **Num Code Files** | Supported | Supported | Supported |
| **Num Images** | Supported | Supported | Supported |
| **Num Other Binaries** | Supported | Supported | Supported |

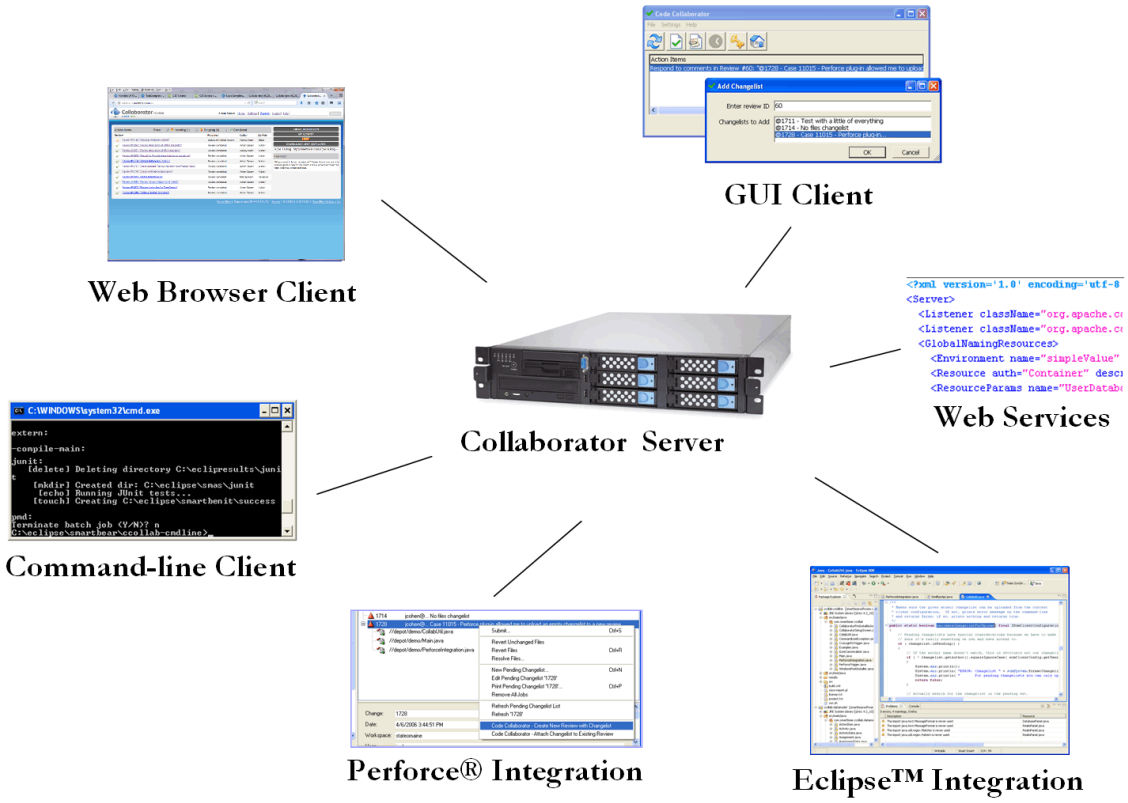
| | | | |
|---|---|---|---|
| **Num Changelists** | Supported | Supported | Supported |
| **LOC (Uploaded)** | Supported | Supported | Supported |
| **LOC (Changed)** | Supported | Supported | Supported |
| **Num Documents** | Not Supported | Supported | Supported |
| **Num URL's** | Not Supported | Supported | Supported |
| **Document Pages** | Not Supported | Supported | Supported |
| **Inspection Rate (Uploaded)** | Not Supported | Supported | Supported |
| **Inspection Rate (Changed)** | Not Supported | Supported | Supported |
| **Document Inspection Rate** | Not Supported | Supported | Supported |
| **Image Inspection Rate** | Not Supported | Supported | Supported |
| **URL Inspection Rate** | Not Supported | Supported | Supported |
| **Binary Inspection Rate** | Not Supported | Supported | Supported |
| **Code Defect Density (Uploaded)** | Not Supported | Supported | Supported |
| **Code Defect Density (Changed)** | Not Supported | Supported | Supported |
| **Document Defect Density** | Not Supported | Supported | Supported |
| **Image Defect Density** | Not Supported | Supported | Supported |
| *Materials Detail* | | | |
| **Path** | Supported | Supported | Supported |
| **Version** | Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Annotation** | Supported | Supported | Supported |
| **LOC** | Supported | Supported | Supported |
| **LOC Changed** | Supported | Supported | Supported |
| **# Comments** | Supported | Supported | Supported |
| **Defects** | Supported | Supported | Supported |
| *Overall Review Conversation* | | | |
| **Speaker** | Supported | Supported | Supported |
| **Text** | Supported | Supported | Supported |
| ***Customizable Defect Reports*** | | | |
| **Defect ID** | Not Supported | Supported | Supported |
| **Review ID** | Not Supported | Supported | Supported |
| **State** | Not Supported | Supported | Supported |
| **Review Title** | Not Supported | Supported | Supported |
| **Review Creation Date** | Not Supported | Supported | Supported |
| **Review Completion Date** | Not Supported | Supported | Supported |
| **Created** | Not Supported | Supported | Supported |
| **Creator Login** | Not Supported | Supported | Supported |
| **Creator Full Name** | Not Supported | Supported | Supported |
| **File** | Not Supported | Supported | Supported |
| **File Version** | Not Supported | Supported | Supported |
| **Changelist ID** | Not Supported | Supported | Supported |
| **Changelist Date** | Not Supported | Supported | Supported |
| **Changelist Author** | Not Supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **Changelist Comment** | Not Supported | Supported | Supported |
| **SCM Type** | Not Supported | Supported | Supported |
| **Location** | Not Supported | Supported | Supported |
| **Comment** | Not Supported | Supported | Supported |
| **Review Group** | Not Supported | Supported | Supported |
| **Defect Custom Fields** | Not Supported | Supported | Supported |
| ***User Reports*** | | | |
| **ID** | Not Supported | Supported | Supported |
| **Login** | Not Supported | Supported | Supported |
| **Name** | Not Supported | Supported | Supported |
| **E-mail** | Not Supported | Supported | Supported |
| **Phone** | Not Supported | Supported | Supported |
| **Last Login** | Not Supported | Supported | Supported |
| **Last Activity** | Not Supported | Supported | Supported |
| **Last Logout** | Not Supported | Supported | Supported |
| **System Admin** | Not Supported | Supported | Supported |
| **Enabled** | Not Supported | Supported | Supported |
| **User Activity** | Not Supported | Supported | Supported |

## 1.3 Components

Collaborator includes a server component, and a variety of clients, 3rd-party integrations, and client protocols.




## Collaborator Server 56

As with most enterprise-class software systems, a server process acts as the hub, manager, and controller of information. The server has a web-based user interface where users and administrators can do everything — create and perform reviews, configure personal and system-wide settings and run reports. The server uses a database to store all data and configuration.

## Web Client 313

The web browser client is used by entering an external URL in a web browser to connect to the Collaborator server. This is where most of the review will take place, where you edit, view, configure the reviews. The Web Client supports any modern browser.

## Command-line Client 506

Developers will typically install the cross-platform client. This tool includes a Command-Line Client that lets you upload local files (and file-changes) into new and existing reviews. The Command-Line Client also includes scripting commands for implementing custom behavior and integrating with external systems.

There are many reasons why you might want to integrate Collaborator with other systems. An issue-tracker integration point might let you synchronize Collaborator "defects" with issue-tracker "issues," or you might want to mirror review data (metrics/comments/file-differences) into the associated ticket. A reporting integration point might let you mirror Collaborator metrics into your existing reporting system (examples: defects/kLOC, defects/man-hour, kLOC/man-hour, number of defects found of different types or severities, etc).

## GUI Client 496

The cross-platform client install includes a graphical client to complement the web-based user interface already provided by the server. The GUI client is a cross-platform client available for Windows, Mac, and Unix/Linux users.

## Eclipse Integration 525

The Eclipse Plug-in gives users the ability to upload files and perform reviews from within any Eclipse-based application. A variety of applications are supported.

## Visual Studio Integration 566

The Visual Studio Extension gives users the ability to upload files and perform reviews from within Microsoft Visual Studio.

## Version Control Integration 620

Collaborator integrates with many popular Version Control Systems. Integrations include command-line, graphical, and Eclipse plug-in clients and Version Control triggers. Server integration is supported for selected Version Control Systems and requires that a properly configured client be configured on the Collaborator server.

## Perforce Integration 764

Perforce users will probably want to install the Perforce Client Integration tools. These are included in the client installer 486.

Integration with P4V and P4Win lets users upload changelists into new or existing reviews just by right-clicking on the changelist. This works on both "pending" and "submitted" changelists.

We also supply a special tool for use as a Perforce server trigger. For example, you can use this to enforce a rule like "Every submit on this branch requires a review". You can also use this to automatically upload all submitted changelists into Collaborator so that you can review files after they have been checked in. This can be especially useful with off-shore development groups.

### Tray Notifier 613

In addition to uploading files, the GUI Client gives you a taskbar icon that updates to show you whether you have any pending tasks in Collaborator. This is called the Tray Notifier and is available to Windows and Unix/Linux users. This allows users to easily access pertinent reviews and alerts the user of any new activity in reviews without being actively involved in the web browser client.

### External Integrations 907

Besides that, you can integrate Collaborator with any other external tool by using scripts or JSON API web services.

## 1.4 Review Styles

### How we view different styles of review

Below are some guidelines for how to structure different types of reviews as well as some settings to consider.

**Quick & Easy** – Generally, these are the types of reviews done when not a lot of formality is required. The review process typically includes:

- Very few participants. Minimally, one Author 279 and at least one Reviewer 279;
- Limited number of custom fields 256;
- Participants not required to 'Mark Read' all comments 284;
- No Moderators (special persons who lead and control the review).

**Formal Inspection** – Formal inspections require specific processes to be followed. Often this is driven by regulations or the need for an audit trail. Formal Inspections typically include:

- Moderators 279 (special persons who lead and control the review);
- Multiple review participants including Reviewers and Observers 279 (subject matter experts);
- Custom fields 256 for capturing additional details;
- Requirements for Moderators and Authors to 'Mark Read' all comments 284.

**Document Review** – Reviewing documents can be as informal or formal as is needed. Most often, the type of document is what drives the formality of the review. Here are some key Collaborator functionality items to consider:

- Should all Reviewers be required 285 to participate or not;
- Remember, that the file name must not change. Do not append anything like _v2 to the next revision of the file;
- Create a unique template that contains Roles 279, Custom Fields 256 and Automatic links 286 that are important for document review;
- Learn how to use the zoom, scroll and lock 367 options.

**Safety Critical** – This if the most formal of all the review processes. Basically, build a template based on the Formal Inspection template.

- Additional custom fields 256 could be required for capturing data;
- Enable Electronic signatures 193 for final signoff;
- Require that all participants 'Mark Read' all comments 284;
- Enable Checklists 272 so that nothing is overlooked;
- Archive the review using 'Save to ZIP 165'.

## 1.5 Review Workflow and Phases

This section describes a typical review and its phases.

### Review Creation

Any of the following actions will create a review:

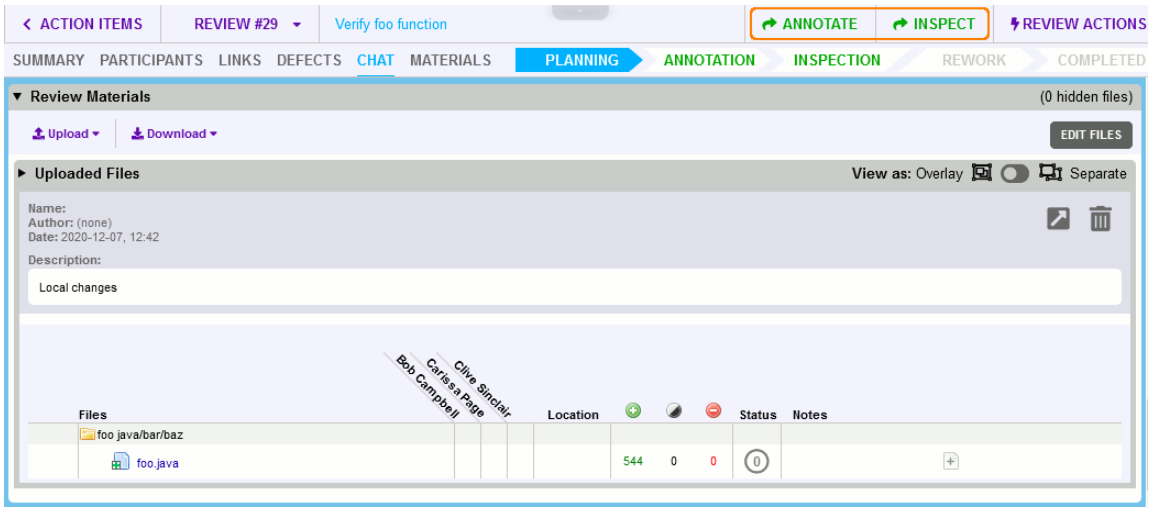
- Clicking the New Review 336 button in the Web Client Home page.
- Adding new materials to a new review through the GUI Client 496, Command-Line Client 506, Eclipse Plug-in 545, Visual Studio Extension 566, or Perforce Plugins 786.
- Using the ccollab admin review create command in the Command-Line Client.
- Triggering a script 926 that creates a new review.

Once a review is created it goes into the Planning phase.

## Planning Phase


In the Planning phase, only the review creator is active to customize the review using the Review Creation Wizard 336. Here the creator enters review attributes including the title, custom fields 256, attaches review materials 340, and assigns participants 338 to the review.

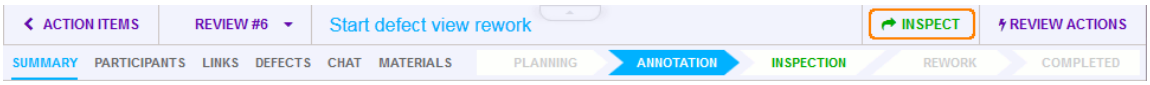
Using the "Annotate" and "Inspect" buttons on the review header toolbar 346, you can save your current information and move the review to another phase.




## Annotating Phase


The Annotating phase is an optional phase that an author may select before the Inspection begins. It provides the opportunity for other participants to add comments 429, review materials 340 or open defects 436 prior to the Inspection phase. When a review is moved to Annotating, Collaborator will send notifications to all selected participants 338 inviting them to participate in the review. Like Planning, Collaborator will not automatically move the review from the Annotating phase.

The review header toolbar includes only the "Inspect" button for moving the review to the Inspection phase.




## Inspection Phase

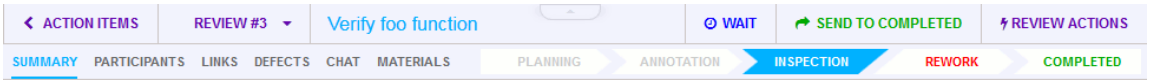

Once the review has begun, Collaborator activates all roles and notifies the participants that their actions are required in the review. Participants are presented 345 with the files, file-differences 379, and other review material. Everyone can make comments on individual lines of a file or on entire documents.

Comments 429 work a bit like "instant message" chat and a bit like "newsgroups". If everyone is chatting at the same time, you have a real-time "instant message" environment so the review can progress swiftly. If one or more participants are separated by many time zones or just are not currently at the computer, the chat looks like a newsgroup where you post comments and receive e-mails when someone responds. This means Collaborator works equally well no matter where your developers or reviewers are located.

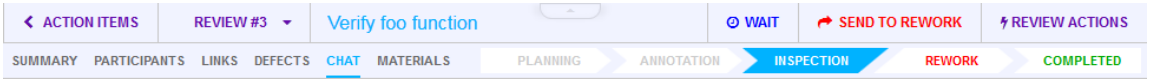
If reviewers find problems, they open defects 436 right from the Web Client, associated with the file and line number if applicable. Defects are tracked through Collaborator and can optionally be mirrored or exported 442 to an external issue-tracking system.

Once all required 285 participants have indicated that they are finished with the current phase, the review moves to the next phase, which will depend on whether open defects 436 remain in the review.

If there are no defects or all defects have been marked external 442 or fixed 438, the review moves to the Completed phase.

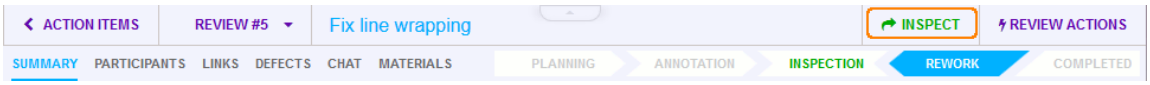



If the review contains any open defects, the review moves on to the Rework phase.



## Rework or Fix Defects Phase


In the Rework phase, the authors are the only roles active so that they may fix the defects found in the Inspection phase. When the authors are satisfied with their fixes, the fixes are uploaded into the review. The authors then indicate they are ready for the fixes to be validated by clicking the "Send to Inspection" button.



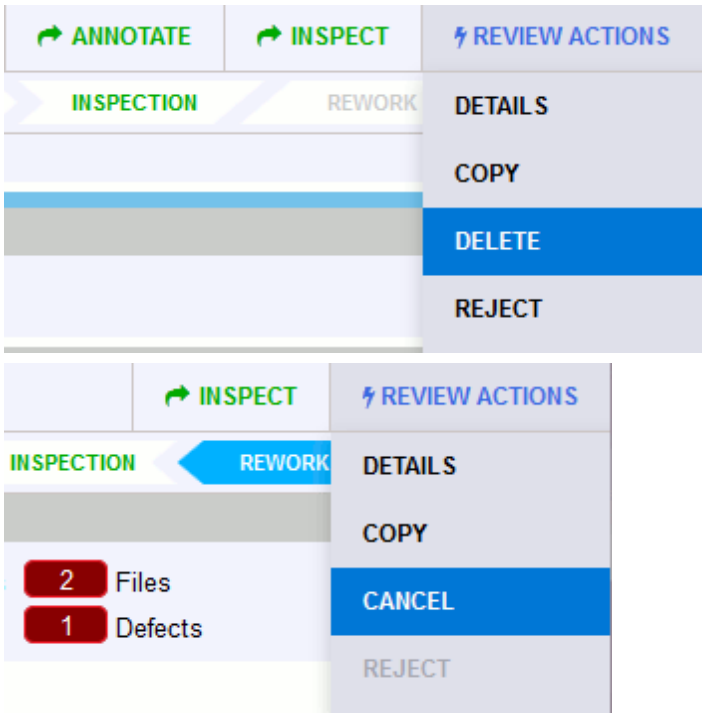

For validation, the review moves back to the Inspection phase, where reviewers have access to the new changes as well as the original changes and comments.

## Completed Phase


Only when no open defects remain and all required 285 participants have approved the review during inspection, then the review goes to the Completed phase.

## Cancelled Phase


Depending on the Allow Deleting/Cancelling Reviews 191 administrative setting, there may be an option for the author/creator to delete 346 or cancel 347 a review. In this case, the **Delete** action will become enabled for planned reviews and **Cancel** actions will become enabled for in-progress reviews:

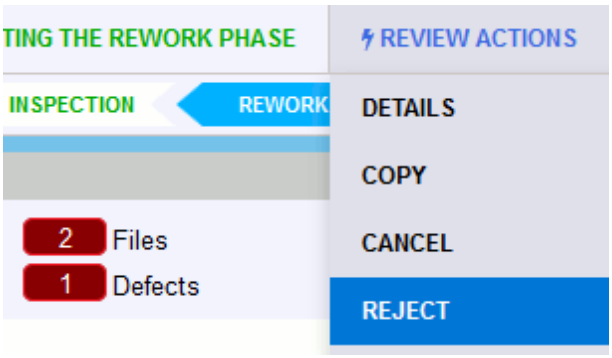



Clicking the "Delete" button completely removes the review from the database. A deleted review can never be restored.

Clicking the "Cancel" button moves the review to the Cancelled phase. A cancelled review can only be re-activated by an administrator.

## Rejected Phase


Depending on the Allow Reject Review 191 administrative setting, there may be an option for reviewers to reject 347 a review. In this case, the **Reject** action will become enabled for in-progress reviews:




Clicking the "Reject" button invokes a dialog where you can specify reject reason and then moves the review to the Rejected phase. Rejected review cannot be reactivated even by administrator.

## 1.6 How Do I?

This section is included as a quick help to get you introduced to some basic tasks in Collaborator. This section is formatted in a series of questions and answers focused on describing how to use Collaborator. While the rest of the manual will serve as the ultimate reference manual, explaining each part or component of Collaborator in detail, this section will be useful if you are new to Collaborator and want to learn how to perform tasks in a quick and easy way. The following questions may not include all your questions, so if you find yourself in need of help, please contact our support team whose information is listed in the Contact Us 32 section.

### 1.6.1 How Do I Get Started?



#### See a typical code review in action?

To get introduced to Collaborator, we recommend viewing a demo to see how it works and how a typical review is done.

You can also attend a webinar (either live or recorded), where we go through a more in-depth demo and also answer your individual questions in real time!

## Get started with Collaborator?

If you are new to Collaborator, we have heaps of reference materials that help you use and understand the product.

If you want to start evaluating Collaborator or get a quick start with the basics, we suggest the Getting Started guides. Getting Started with Collaborator Server helps system administrators install and configure the main Collaborator server, while Getting Started with the Collaborator Client helps developers get started conducting code reviews with Collaborator.

System Administrators 35

Developers 42

We also have the User's manual that serves as a reference manual for everything you want to know about Collaborator. It is divided into sections for installation and server administration, web reviews and the web browser client, the specific integrations, and external integrations.

And if you still have questions, you can always contact us 32. We would be happy to assist you with any specific questions about Collaborator.

## Experiment with Collaborator without installing the product?

We have a demo server waiting for you to try! The demo server is an already set up Collaborator server, so you do not have to hassle with administrating a server.

1) Visit our demo server at http://demo.smartbear.com.

2) Create a new account by supplying a login name and password of your choice under "Create a new User" and jump right into Collaborator.

**Note:** The demo server is publicly available to everyone. Do NOT post confidential information and/or materials.

For more detailed information on how to get started with the demo server, visit our blog entry here.

3) Read our Getting Started Guide for Developers 42 to quickly get caught up with the basic code review steps, or explore around by clicking random links to learn by trial and error.

## Make sure code reviews are worth the effort without taking too much time out of my work day to try them out?

We think code reviews are awesome, but we understand you might be skeptical. And why invest in something you are not sure of? We put together some tips on how to minimize your time evaluating the code review process. This document, Is Code Review for You?, explains how to try code review for a week, how to see if it delivers solid, measurable results for your team, and how to get your team on board.

## Perform my first review?

So you have been assigned your first review, and you are not sure what the next steps are. We'll guide you through so you can jump right in. Just read our blog post here.

## 1.6.2 How Do I Do Reviews?



### Know what parts I really need? What are all these parts?

Every organization needs a Collaborator Server installed and set up. (Read the Getting Started for Server Administrators 35 for instructions.) The server process acts as the hub, manager, and controller of information. (Download the Server Installer for this component.)

To connect to the server, you use the Web Browser Client[313] by entering a URL (provided by your Collaborator server administrator) into your web browser. The Web Browser Client is where users and administrators can do everything — create and perform reviews, configure personal and system-wide settings and run reports.

The following clients are all optional but very useful. Which one works best for you depends on your version control system, code review environment, and/or personal preferences. Just read the descriptions, and choose what you need or want to use.

The Command-Line Client[506] is a cross-platform client interface to the Collaborator server. It can be used for uploading files, integrating with version control, and querying the server, or as a part of an automated script in a sophisticated ALM / build system.

The GUI Client[496] is a cross-platform graphical client to complement the web-based user interface already provided by the server. Use it to upload materials to a new or existing review easily by integrating with any or a combination of the following Version Control Systems: AccuRev, ClearCase, CVS, PTC Integrity, Perforce, Rational Synergy, Rational Team Concert, Subversion, and Team Foundation Server. (If you use a different VCS, you can always use the Command-Line Client.)

The Tray Notifier[613] is a taskbar icon that shows whether you have any pending tasks in Collaborator. This icon allows you to easily access pertinent reviews and alerts you of any new activity in reviews without requiring active involvement in the web browser client.

The Perforce P4V/P4Win plug-ins[786] integrate with the Perforce GUI clients and allows you to upload changelists into new or existing reviews just by right-clicking on the changelist.

The Eclipse Plug-in[525] lets you upload files and create reviews from within any Eclipse-based application.

The Visual Studio Plug-in[566] gives you the ability to upload files and create reviews from within Visual Studio.

## Create a new review?

In the Web Browser Client[313], creating a new review is as simple as clicking on the **Create New Review** button at the top right of the action items section of the Collaborator home page and filling out the information in the next screen.

Wait, your **Create New Review** link is disabled? Congratulations, you are the Collaborator administrator. By default, we do not allow admins to participate in reviews, so the link at the top right does not show up. You can change this setting by going to the **Admin** link on the menu bar at the top of the screen. Once you are at the Admin menu, scroll down to the "Review Process Options" and set the *Allow System Administrator to Perform Reviews* setting to *Yes*.

In the Command-Line Client, use the command: `ccollab admin review create` to create a new review.

You can also create a new review when adding review materials from the GUI Client, the Eclipse Plug-in, the Visual Studio plugins, and the P4V plugins.. The client will give the option to add the material to a new review or an existing review. Be sure to select *Create New Review* if you want a new review.

## Get around having to *Read* all comments?

In the default configuration, you must *Read* all comments. But you can change this! As the system administrator, you can change this setting; go to the **Admin** menu, look under the sub-category **Roles**, select a Role "set", and change *Required to Read All Comments* to *No*. This setting is configurable for each role, so you can ensure that all comments are read by someone, but not everyone.

When this setting is configured to *Yes*, a review participant cannot click the Finish button if the review contains unread comments. When set to *No* for a role, this setting allows a review participant to click the Finish button even if there are unread comments. Also, you will not be brought back to a review if some other participant has commented after your clicked the Finished button. Read our extended blog entry about this feature here.

But my system administrator will not change this setting...

You'll have to mark all comments read, but you can do this easily by clicking the super **Mark All Comments as Read** button at the top left of the Diff Viewer. You will still have to click it for every file (hey, we have to make sure you are looking at the files) but it saves you clicks in the end.

## Figure out what "Accept" means?

We get this question often, and the short answer is: "Accept" means whatever you want it to mean, so you do not even have to use it if you do not want to. "Accept" does not have an explicit meaning in Collaborator; it is used differently in different environments. So, you can use this feature however you think best suits your review environment. Once this button is clicked, a green checkmark appears to notify participants that a comment has been Accepted.

## Know why (or when) was (or will I be) I brought back to a review?

You may have been brought back into a review because of your *Required to Read All Comments* setting (See above). Read more about this here.

## Put a review on pause?

If you cannot continue or finish reviewing because you are waiting for more information (For example, an answer to a question you posed in a comment), you can put the review on hold by using the **Wait** button in the Next Steps section. "Wait" keeps you deactivated from the review until the action option you have selected occurs. So the **Wait** button is for when you want to come back to a review at a later time. If you don't return and the selected activity does not occur, the review will not be marked finished, so to prevent a review from stalling, be sure to come back and finish it.

## Finish a review/Clear an action item?

So you want to finish your part of the review. If you are in a role that has the ability to mark reviews finished, it is pretty simple. First, defects must be marked resolved: either *fixed*, *deleted* or *tracked externally*. Then, under the "Next Steps" section, select **Send to Completed**. If you are not the only reviewer, a list of actions will appear. You can select an action to indicate that you are finished with the review *unless* the selected action occurs. If you are required to read all comments, make sure all comments have been marked read before clicking **Send to Completed**. You will also be brought back to the review if another comment is made after you have approved a review.

## Finish a stalled review? Perhaps someone left for vacation, and now the review cannot be completed.

Several things may prevent a review from being marked *Complete*. If an open defect is keeping you from closing a review, have a system administrator resolve the defect, and you will be able to finish the review. If the absent participant is the only person who has the ability to complete reviews, simply change that participant's role in the review so that they are assigned to a role that is not required to complete the review. For more detailed information, read the blog post, "Ending a Review Now", here.

## Indicate file review order?

You can use the "General Chat" and the "Notes" areas to annotate files and give instructions on suggested file review order. For detailed step-by-step instructions, read our blog post here.

## Remove a participant from a review?

Removing a participant from a review simply requires clicking the red 'X' next to their name.

## Change my notification level?

You can change the frequency at which you receive notification emails by going to your **Preferences** menu and then setting the *Notification Level* in the **Notifications** tab. When your individual preferences are set to the default setting, *Minimal*, you only get notifications resulting from other users' actions. You can also change the setting to *None* or *All*.

## Remove a file?

Once a review has started, removing a file is only possible if there are no comments or defects linked to the file. To remove a file, click the **Edit** button on the "Review Materials" section of the review summary screen. You should see a **[Delete]** link to the left of the file or changelist, in which case click the link to delete the file. You may need to set the "View As:" option to "Separate" in order to be able to see the specific version of the file you wish to remove.

If you do not see the link, it indicates that comments or defects are linked to the file or changelist; you will not be able to delete the file because it would orphan comments.

## Compare files or choose which version of a file to look at?

To view different versions of the file you are reviewing in the Diff viewer, click on **Compare** at the top of the Diff viewer. Collaborator shows you a list of all available versions that you can choose from. Use the radio buttons for the right and left side of the Diff Viewer to select the versions. To view only one version, set both left and right sides to the same version.

You can also compare a current version to the latest Accepted version, which is detailed here.

## Change the Diff Viewer? I don't like the Diff Viewer.

You cannot change the aesthetics of our Diff Viewer, but if you prefer other diff viewer software, we do support an External Diff Viewer Launcher, which you can use to review diffs in a separate diff viewer of your preference. To configure this capability, please visit the owner's manual here 615.

## Find out how much time I have spent on each review?

Beginning with Collaborator v6.0, user oriented reports allow you to view your own statistics even if you are not an administrator. View our blog post here.

## Do code review in Eclipse?

With the Eclipse Plug-in, you can now review code within Eclipse. To get caught up with the new changes, visit our blog here.

## Add files to a review from Visual Studio?

Beginning with Collaborator v6.0, the Visual Studio Add-in offers support for adding files to Collaborator from Visual Studio. To learn more about this add-in and feature, visit our blog here.

## Review Office Documents?

Collaborator has native support for Microsoft Office Word (.doc and .docx) 383 and Excel (.xls and .xlsx) 389 documents. To review other types of Microsoft Office Documents you will need to convert them to PDF format 414 first.

## Delete/redact a comment?

While we still don't allow the deletion of comments, we do provide a way to indicate that a comment is now irrelevant and should not be considered for the review. Detailed steps and screenshots are available here.

### Compare ClearCase files that are on two different branches?

To compare file versions that are on two different branches in ClearCase you have to specify the specific ClearCase version identifiers. Beginning with v6.0 of Collaborator you can use our GUI Client to do that; detailed steps are available here.

## 1.6.3 How Do I Do Setup or Administrate the Server?



### Switch from internal authentication to LDAP?

The easiest way to change authentication options is to re-run the installer again. There is no need to uninstall, just run the installer over the existing installation. From there, you can choose "LDAP Authentication" when asked and provide your LDAP credentials.

### Reset my password?

Sorry, you will have to ask your Collaborator system administrator about this one.

### Upgrade to a new version of the Collaborator server?

1) Back up first! 99

2) Note your Support and Upgrades date by going to **Admin>Licensing** . In the Current Licensing box, take note of the date next to the *Upgrades Expire On:* field. Make sure the version you want to upgrade to was released before this date. You can find this information by looking at the dates next to the version numbers in our Version History 982.

3) Run the new version installer over your existing installation and choose *Existing Configurations* when prompted.

*Note: Be sure to fully read the Server Upgrades 83 section.*

## Change administrative control (especially in LDAP)?

Administrative Access can be given to individual users by an existing administrator. Just have an administrator go to **Admin>Users** and edit the account you would like to give administrative control to. On the **Editing** page, set the **Is Administrator** field to Yes. Administrative Access allows a user to access to all reviews, reports, and also edit any review.

The Special Administrator Account is a privileged account that is always allowed to login regardless of how many licenses are in use. With Internal Authentication, the special administrator account is always "admin". This cannot be changed.

With LDAP, there is one user assigned as the "System Administrator". If you would like to change this assignment, go to the ROOT.xml configuration file at `$INSTALLDIR/tomcat/conf/Catalina/localhost/ROOT.xml`. Near the bottom of that file is a line:

```
<Parameter description="The name of the Collaborator system
administrator who is always allowed to log in". name="system-
administrator" override="false" value="admin"/>
```

Change the value to the login name you would like to set the account to:

```
<Parameter description="The name of the Collaborator system
administrator who is always allowed to log in". name="system-
administrator" override="false" value="XXXXXXXXXXXX"/>
```

where XXXXXXXXXXXX is the login name of the System Administrator. Then stop and restart the service.

## How do I view metrics and reports?

Collaborator provides built-in reports that allow you to view detailed metrics by reviews 468, defects 475, and users 478. To view these reports, you must have access to reports by enabling the Reports Access 187 setting in the **Admin** menu. Once you have access, you may view built-in reports by clicking the **Reports** button that will appear at the top right menu of the Collaborator screen.

## Collect data from each review participant?

Participant Custom Fields 258 allow administrators to collect data. To learn about enabling and utilizing this feature, please view our blog post here.

### Find metrics for non-text documents?

Additional metrics for binary files, such as images and PDFs, can be viewed in v6.0 and newer. To view these metrics for a single review, visit the Reports[467] section (provided you have access) and view a Review Detail Report[468]. Our blog post here includes a helpful view of the new metrics to look for.

Many of our customers use our custom reporting[908] features to obtain reports tailored to their needs. Learn how to create custom SQL reports here[917] or learn about our database schema here[908].

### Define groups of users?

You can either set up user groups manually or by syncing them automatically to externally defined groups.

To set up user groups manually, view our owner's manual here[228].

To learn how to sync, view the Group page here[239].

We have also provided some examples of how to use groups that may help you in deciding how to fully utilize this feature here[227].

## 1.7 Sunset Announcements

This section announces plans to deprecate some of Collaborator functionality:

- Collaborator's Bitbucket integration will no longer be able to support Mercurial starting on June 1, 2020, when Atlassian will officially remove Mercurial features and repositories from Bitbucket and its API.
- Oracle's extended support for the Oracle 11g database will end on 31st of December 2020 and Collaborator will no longer be able to support it.
- Single sign-on integration using Java servlet for Tomcat is deprecated and will be removed in future releases. Please consider implementing single sign-on using SAML[132] or Atlassian Crowd OpenID[145] instead.

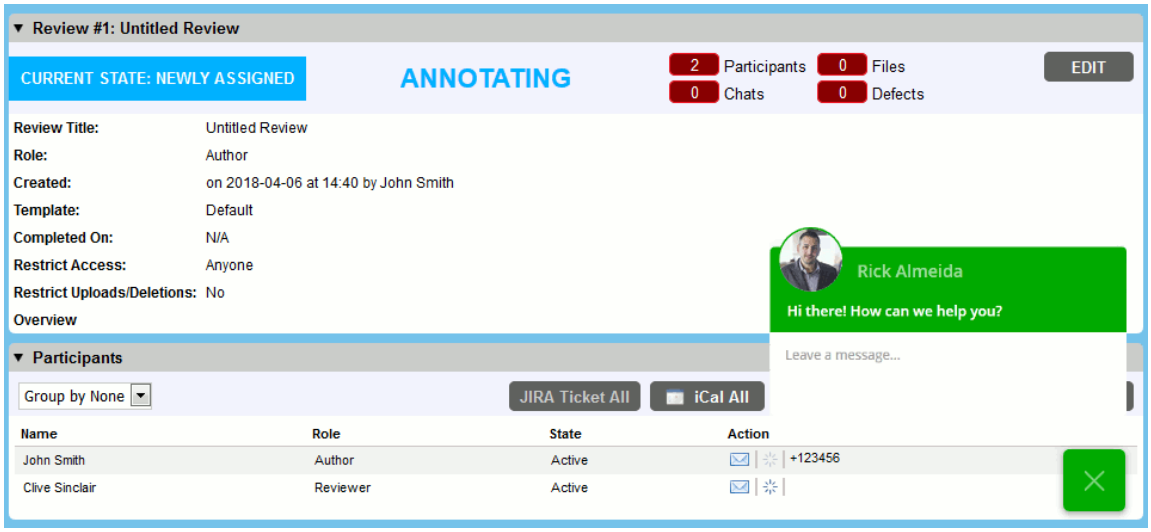
## 1.8 Contact SmartBear

If you have questions, problems or just need help with Collaborator, you can either contact our Support Teams or try to search for the needed information using the help resources located on our web site (community, blogs, technical papers).

# Contacting the Support Team

## Chat With Us (for Trial users only)

To help trial users have a better experience of using Collaborator and solve common issues, the trial version of Collaborator includes a built-in chat. You can use the chat to contact our team directly from the product. To ask for assistance, click the Chat button in the bottom-right corner of Collaborator Web Interface and enter your question. You will get a response from our Customer Care team.




## Leave a Request on the Website

If you are having issues with Collaborator and need technical assistance, open the Contact Support Form on our web site and fill in the required fields:

https://support.smartbear.com/message/?prod=Collaborator

The Support team will answer you via email. All further communication will be made via email. However, to start a conversation, please use the Contact Support Form on our web site.

For information on our support policies, please visit our web site http://support.smartbear.com/about.

# More Resources

- To get acquainted with the product faster, watch **video tutorials** and **screencasts** on our web site:
  http://support.smartbear.com/screencasts/collaborator/

- You can ask questions, search for answers, exchange comments and suggestions on our **community**:
  http://community.smartbear.com

- Learn more about using Collaborator from **technical papers** and **blogs** published at:
  http://support.smartbear.com/articles/collaborator/
  http://blog.smartbear.com

- Make sure you regularly visit our web site, http://smartbear.com, where you will find -

  - News
  - Downloads
  - Articles, Case Studies, Webinars
  - Updated support options

## 1.9 Copyright Notice



### SmartBear Terms of Use (EULA)

To read the End-User License Agreement (EULA), please refer to the following link:

https://smartbear.com/terms-of-use/

### Third-Party Libraries

Collaborator includes third-party open source software modules that are subject to their respective licenses. A list of third-party open source software and their licenses is available on our website.

# 2 Getting Started

This section will briefly guide you through the steps of getting started with Collaborator. It is intended for users who want to jump right into Collaborator and start using the basic features. If you would like more detailed instructions on how to use Collaborator, please visit the appropriate sections of this manual.

We have divided this section into several guides.

The Getting Started For System Administrators 35 shows you how to install and configure the Collaborator server.

The Getting Started For Developers 42 describes how to work with Collaborator as a regular user. It consists of three stages:

- Creating Reviews 42 tells how to create reviews as an author.
- Participating in Reviews 48 tells how to take part in reviews as a reviewer or observer.
- Reworking Defects 53 tells how to get notified about the defects found by other participants and how to make corrections to your reviews.

## 2.1 Getting Started For System Administrators

In order to use Collaborator, you must first install and configure the server side of the Collaborator.

### 1. Download the Collaborator server

Navigate to SmartBear Downloads Center: http://support.smartbear.com/downloads/collaborator/ and download the "Server Installer" that corresponds to a platform of your server.

### 2. Install the Collaborator server

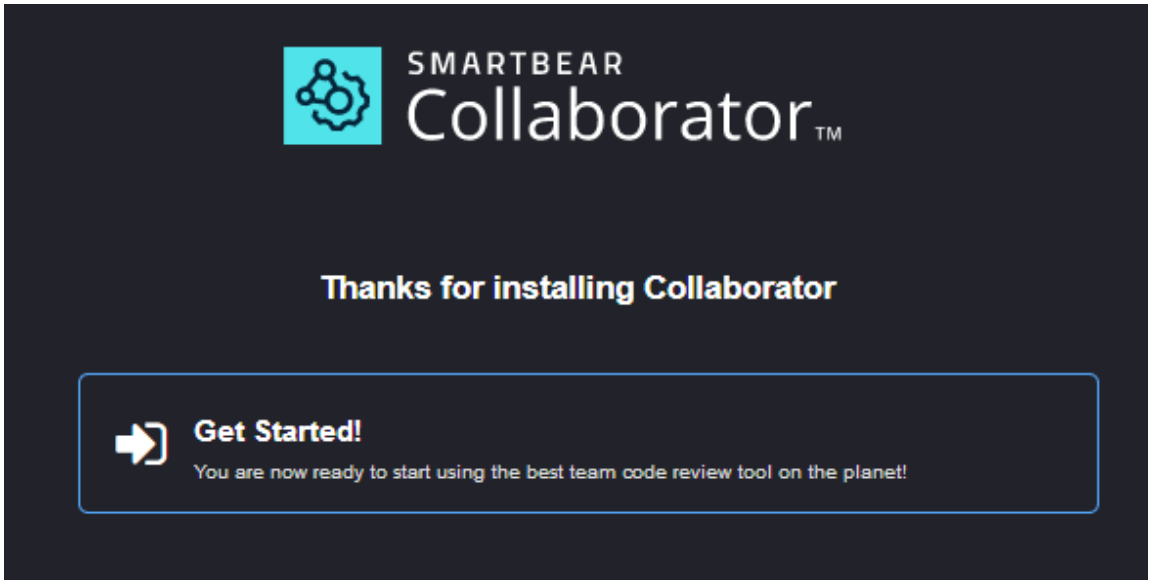
Launch server installer and go through the wizard 70 to complete the installation.

> **Note:** During the installation you may configure advanced settings for the Collaborator server: server port, authentication type, database type and so on. For first-time and evaluation installations we suggest that you leave the default values for these settings.

Once the installation is finished, your default browser will start and open the Collaborator's Web User Interface. The rest of the configuration and administration process is done using the Web User Interface.

### 3. Initialize database

When opening the Web User Interface for the first time after installing the Collaborator server, the database initialization screen is displayed.

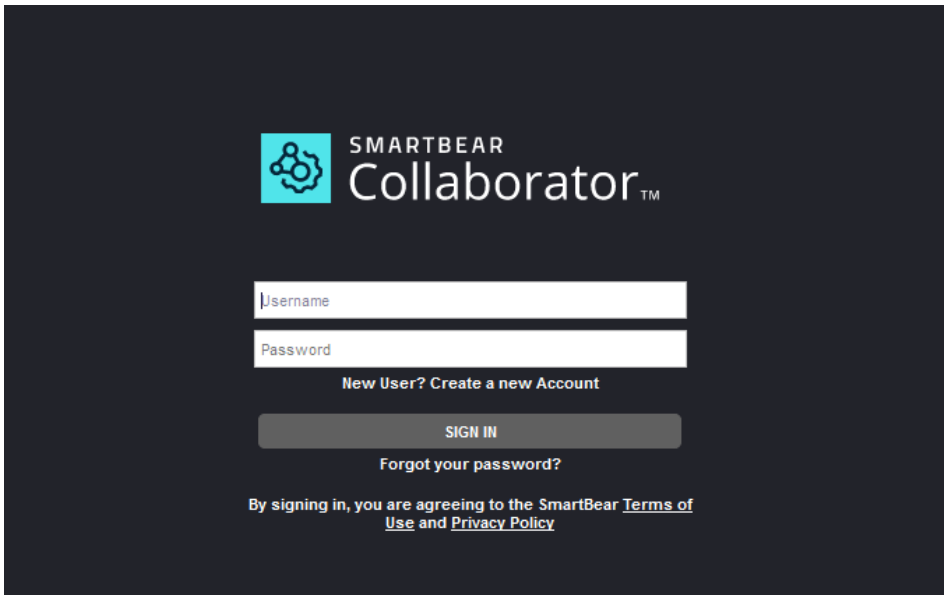



Click **Get Started** to continue.

### 4. Perform first-run initialization

If you have chosen the default values for advanced settings or chosen "*Internal Authentication*" during the installation process, then specify the same username and password that you have provided during the installation.

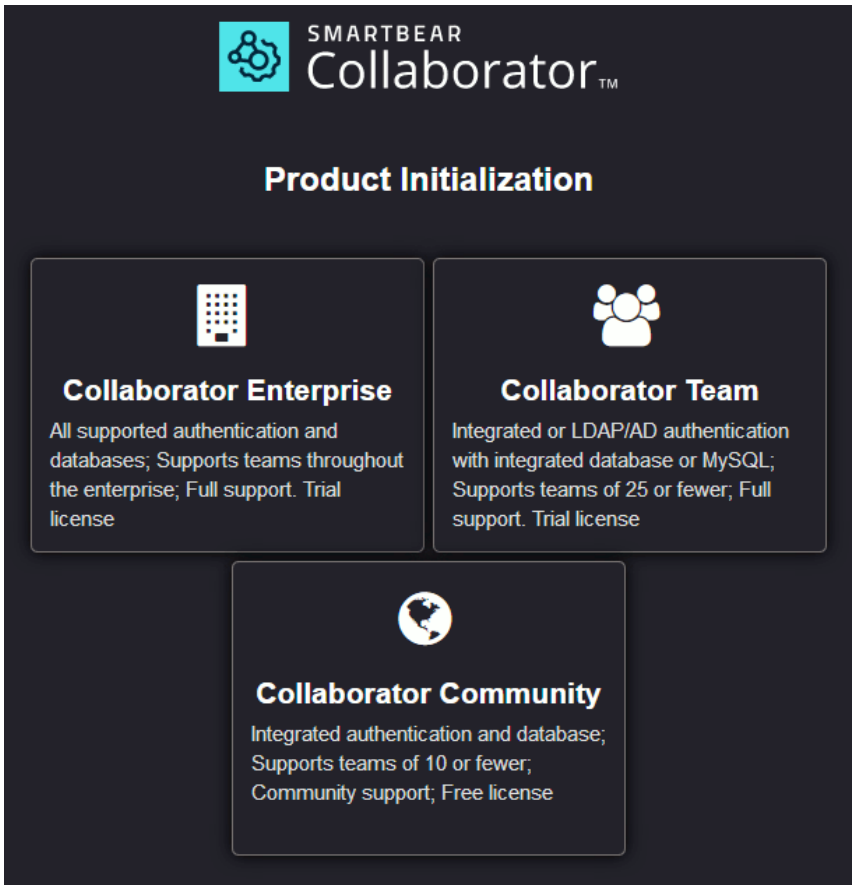
If you have chosen the "*LDAP Authentication*" during the installation process, then specify the same username and password that you have provided when configuring your LDAP integration. Please see LDAP Authentication 119 for more information.




Click **Sign In**.

Once you have logged in as the admin, you will be redirected to the First-Run Initialization page. This page is displayed only once after the installation and asks you to provide necessary information to set up the Collaborator server.

In the Product Initialization section you will need to choose which product you would like to use: Collaborator Community, Collaborator Team or Collaborator Enterprise.



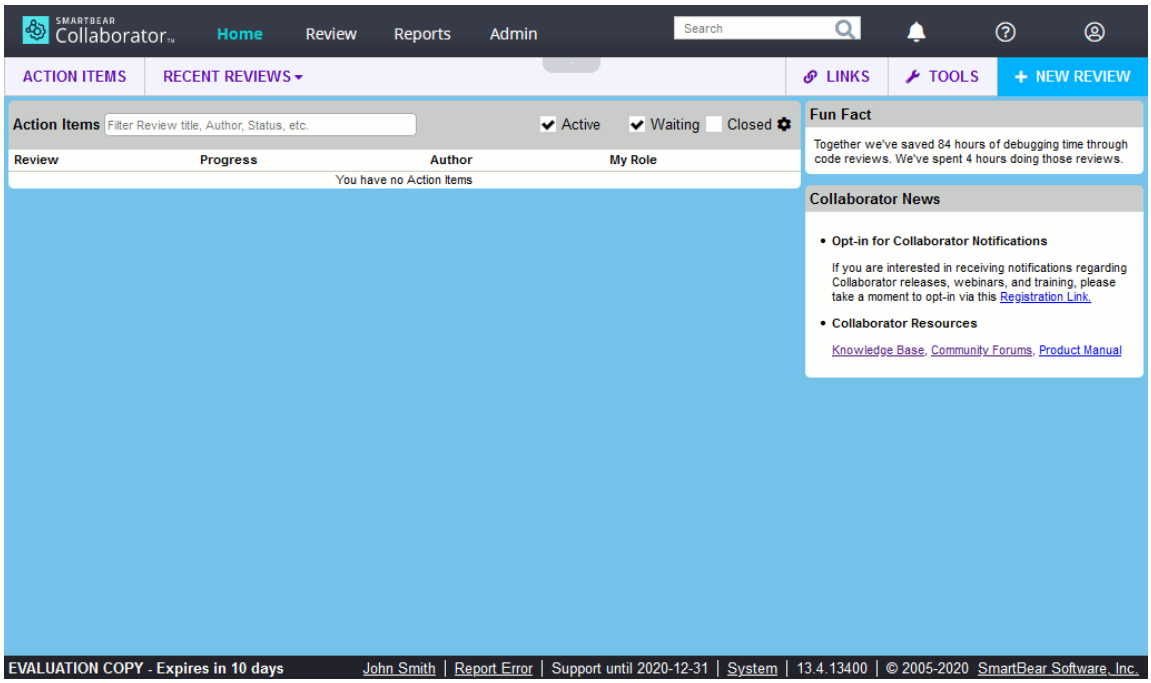

All products have the same goal, to make your peer review process easier and more effective, but provide different sets of features. Collaborator Community is intended for small developer groups, Collaborator Team is for medium groups and Collaborator Enterprise is for large enterprise level companies. To view all differences between Collaborator editions, please read this page 3.

**Note:** Depending on certain features that were chosen during the installation 70, the Product Initialization section may show a smaller set of available products. For example, if Oracle database was selected during the installation, then only Collaborator Enterprise will be available.

## 5. Configure basic system settings

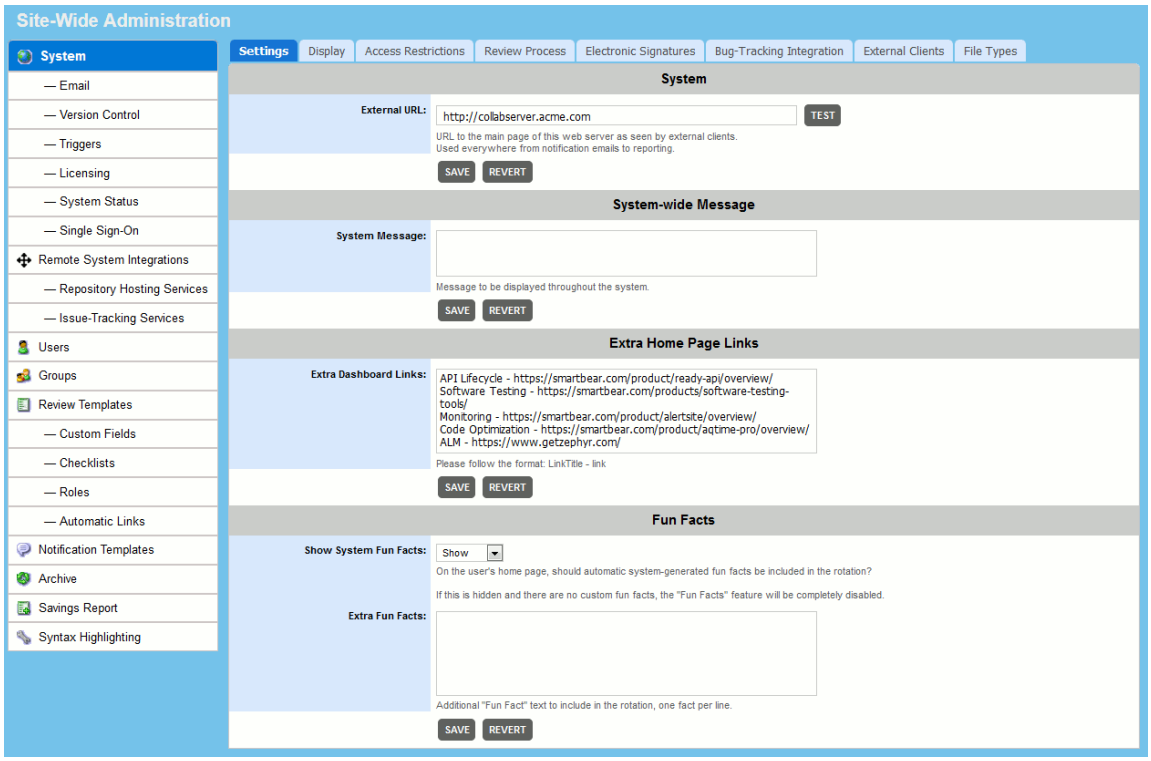
After the first-run initialization, you are redirected to the Home page. This will be the starting page for administrating and working with Collaborator. Later on, after logging in you will start your work with Collaborator from the Home page.

This page looks much the same as your average Collaborator user's home page. For the administrators the page additionally displays the Admin section on the top toolbar, the System link in page footer.




As the Collaborator administrator, you would like to familiarize yourself with the administration settings, so click the **Admin** section in header menu.

The Site-Wide Administration Settings section is where you can configure the settings of your Collaborator server.




The administration settings are grouped into a number of categories: General, Users, Groups and so on. You can read the detailed description of each setting in the Collaborator Settings 178 section.

When configuring a Collaborator server for the first time, pay your attention to the following settings from the System category:

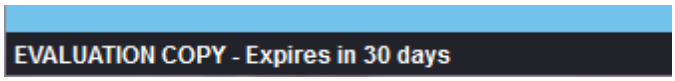
- *External URL* – Enter the full URL to the main page of the web server on which Collaborator was installed, as it would be seen by external clients.
- *Display | Global "Create User"* – When you are starting out, and you do not want to manually create usernames for all users, mark this field as "Show" so the users can create their own username and password. Mark it as "Hide" if you want to control the creation of users and usernames.
- *Access Restrictions* – Control who has access to reviews, reports, subscriptions, and systems information.
- *Review Process* – Control what users can or cannot do during the review process.

Besides, most likely you would need to configure the following categories, as well:

- **Email** - Configure SMTP server settings and email notifications.
- **Remote System Integrations** – Setup integrations with remote repository systems like GitHub, GitLab or Bitbucket, or with issue-trackers like JIRA.

## 6. Activate your licence

If you have an external internet connection, Collaborator will attempt to contact our licensing server so that we can automatically set you up with a **30-day trial license**. Your licensing status is displayed in the bottom-left side of home page footer.




If you do not have an external connection, or if a firewall prevents access to our licensing server, you need to obtain a license code manually.

To request a license, please fill out this form:

https://smartbear.com/product/collaborator/free-trial/

To enter **a permanent license code**, or a trial license code that you obtained manually:

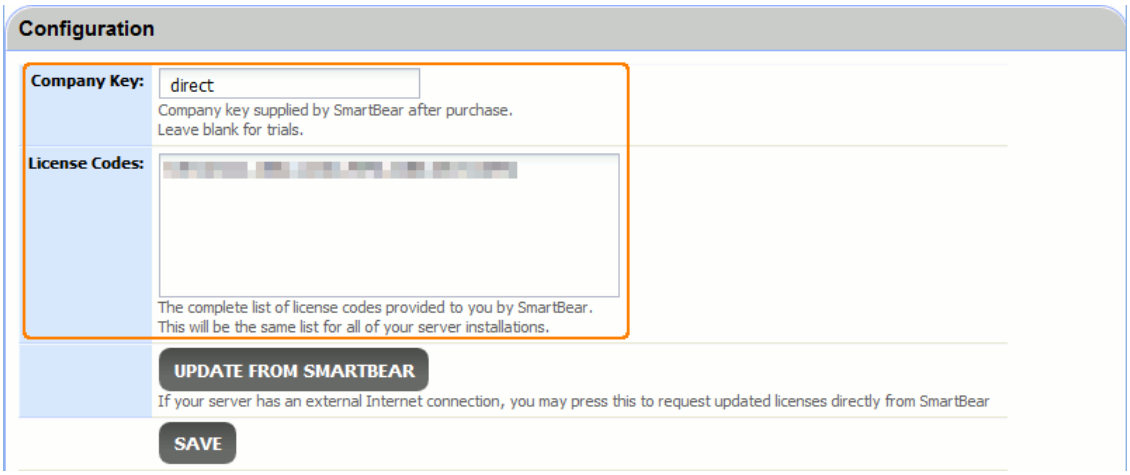
1. Click **Licensing** category of the Admin section.

This will open a page that gives you information regarding the licensing, such as when the license expires, when the support and upgrades expire, how many seats you have and so forth.

2. Find the "Node ID" field in the "Current License" pane. The Node ID is an 8-character value that is unique to every installation. Node ID is tied to every license code we generate.


3. Copy an 8-character value of Node ID and send it to your Account Manager to activate your license. If you do not know who your Account Manager is, just send the Node ID to sales@smartbear.com.

4. After you send us the Node ID, you will receive an e-mail with a company key and license code. The company key is "trial" for temporary licenses and a word or phrase for permanent licenses. The license code will be a 32-character code. Enter the company key and the license code in their respective fields, and click **Save**.




That is all. Collaborator server is installed, configured and ready for operation.

### Where to go next

To get detailed information about the server, its configuration and maintenance, read Server Administration 56 section.

To learn how to upgrade an existing Collaborator server, read Server Upgrades 83 section.

To learn about using Collaborator in a daily work, see Getting Started For Developers 42 and Web User's Guide 313 sections.

## 2.2 Getting Started For Developers

Peer review process includes three major types of activity: creating reviews as an author, participating in someone else's reviews as a reviewer or observer and improving your work according to comments and defects found by another participants.

This guide explains how to use Collaborator for to perform all these steps.

- Creating Reviews 42
- Participating in Reviews 48
- Reworking Defects 53

> These guides assume that your Collaborator server has already been installed and configured 35 by your system administrator.

### 2.2.1 Creating Reviews

You have several options for creating a review: via Collaborator Web Client, via GUI Client or via

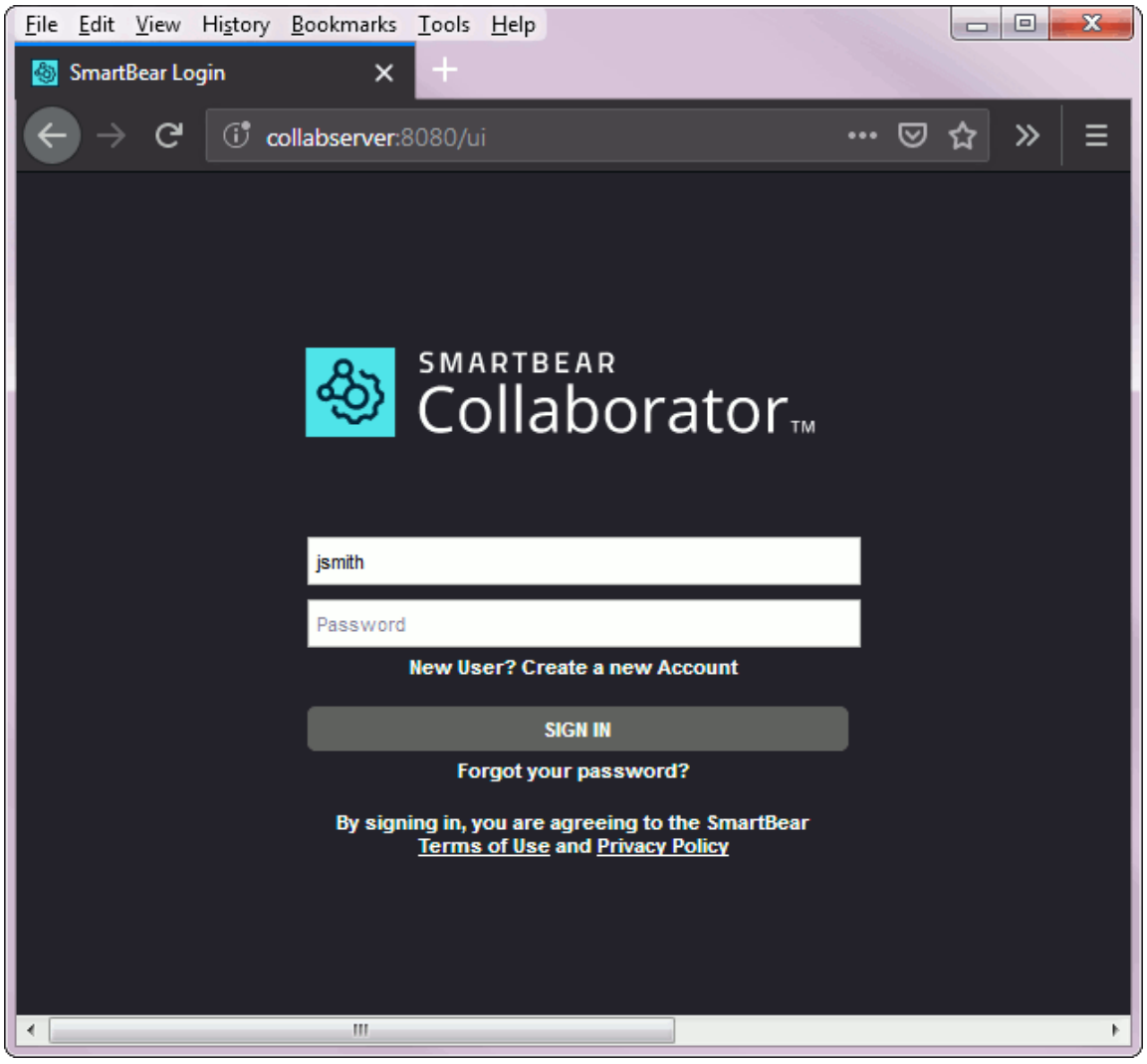
Command-Line Client. Besides you can create reviews directly from Eclipse, Visual Studio or from P4V or P4Win clients for Perforce. To ease you into the process, we will start with the Web Client 313. It does not need any client components except for web browser.

## 1. Open Collaborator Web client

Start your favorite web browser and navigate to the URL of the Collaborator Web Client. If you do not know the URL, ask your Collaborator administrator. The URL is defined by the External URL 180 setting.

## 2. Log into Web client

You will be redirected to the login screen.



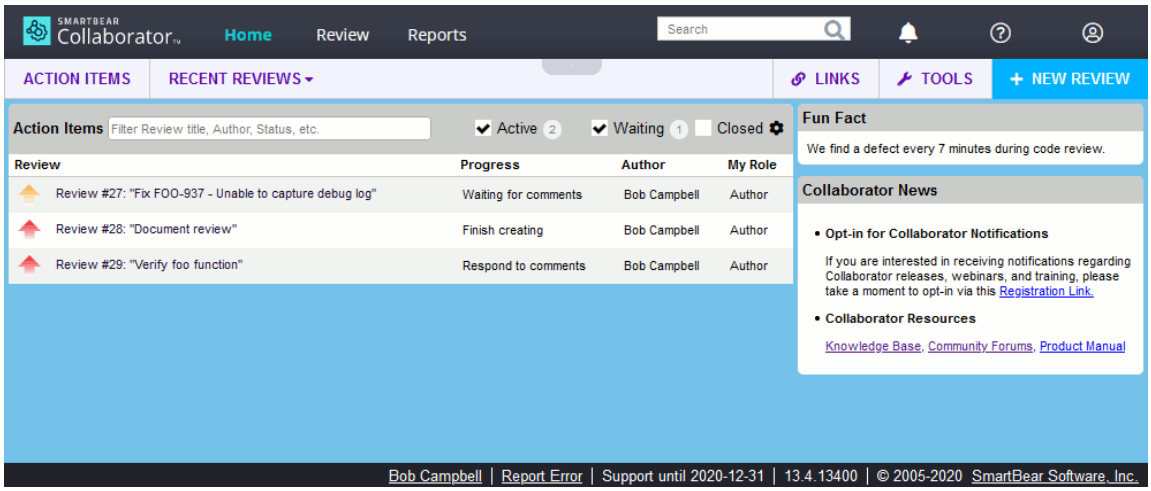
Enter your user name and password and click **Login**. The username and password is typically supplied by the administrator 220.

If you use LDAP or ActiveDirectory 119 for authentication in your company, you can use that username/password with Collaborator and it will automatically create your user account.

If your administrator allows user created logins, a "Create New Account" form appears on the login page where you can create your own login name and password. In this case we recommend that your account in Collaborator be identical to your version control system account.

## 3. Get familiar with Home page

Once you log in, you will be redirected to the Home page. This page is a starting point for most of peer review process.

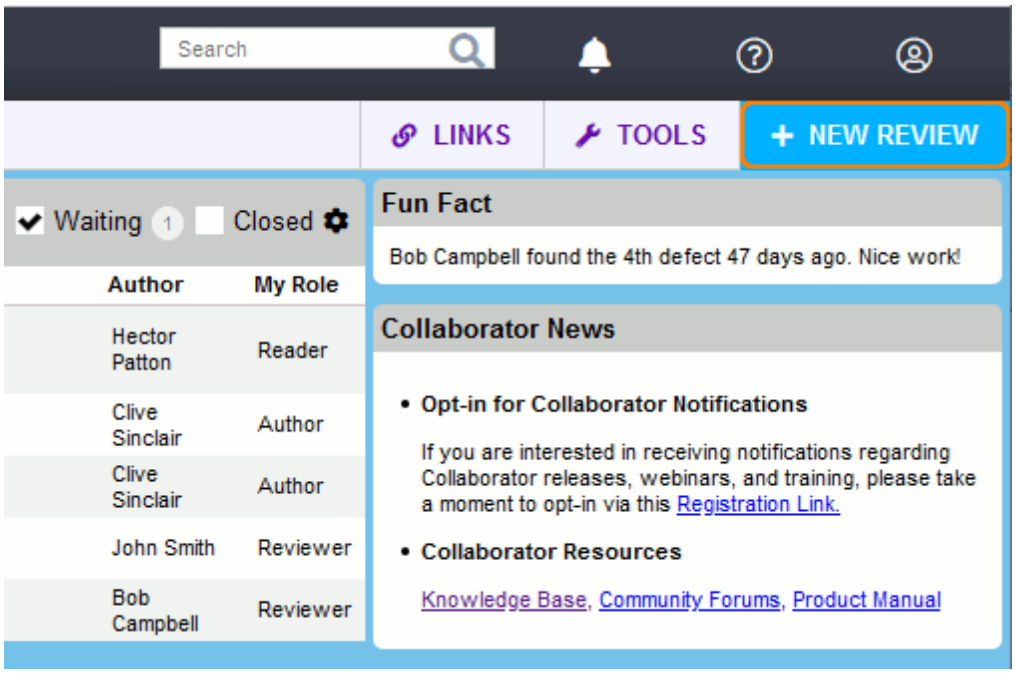



*Collaborator Web Client Home Page*

On Home page you can check your Action Items 333 to see if you have any assigned reviews. You can also edit and change your settings and preferences by clicking Settings 317 in the menu bar on the top right of the screen.

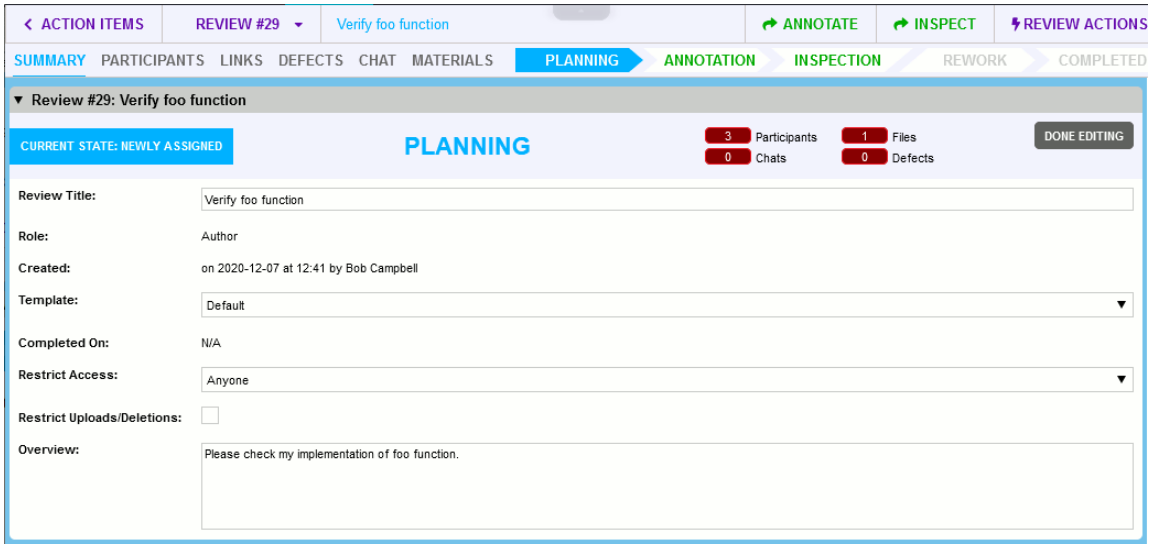
## 4. Create a new review

To create a new review, click **New Review** on the Home screen.




You will see a Create Review Screen[336] where you can specify information about the newly created review.

## 5. Specify general information about a review

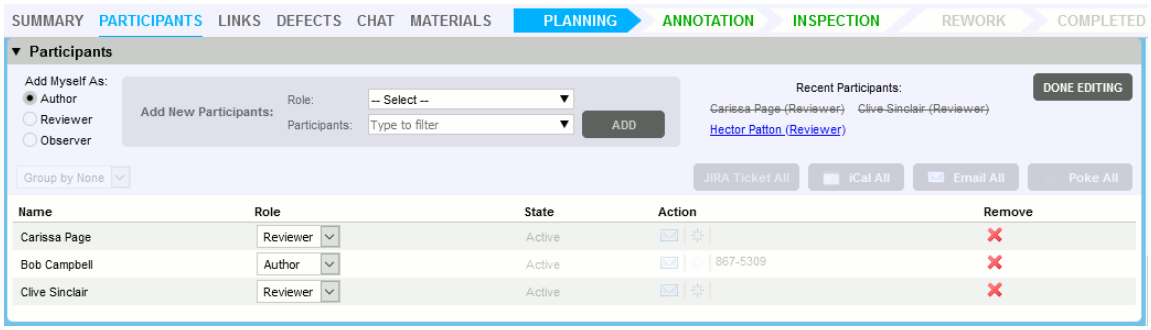



Enter a brief description for the review to the **Review Title** field. The title is used all over the place -- in Action Items, in notification emails, in web page titles, and so forth.

All other fields of the General Information section are optional, so you may skip them.

## 6. Add review participants

The next step in creating a review is to invite people to the review, and give each of them a role to play.




1. Add yourself as Author 279.

2. Add one or more of your colleagues as Reviewers 279.

To add another user as a participant, select the desired role in the **Role** drop-down menu, then select one or more users in the **Participants** drop-down menu, and click **Add**. Start typing the name of the user in the appropriate field to narrow down the user list. User/role combinations you have used recently will also appear under "Recent Participants". This makes it easy and fast to select common combinations.

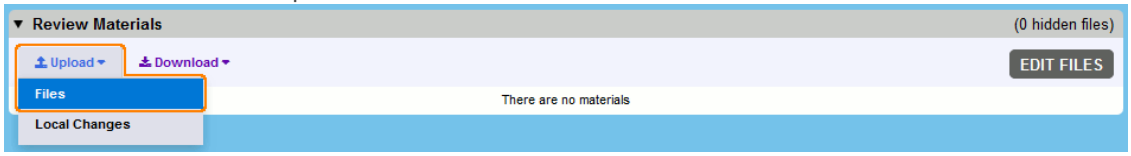
## 7. Add review materials

Review materials is what you are asking to review. From the Web Client you can upload the following types of content:

- Arbitrary files from your computer: text files 377, word processing documents 383, spreadsheets 389, presentations 395, PDF documents 414, Visio graphics 400, images 410 and so on.
- Changes committed to a pre-configured 204 version control system

Let's upload some text file to a review.

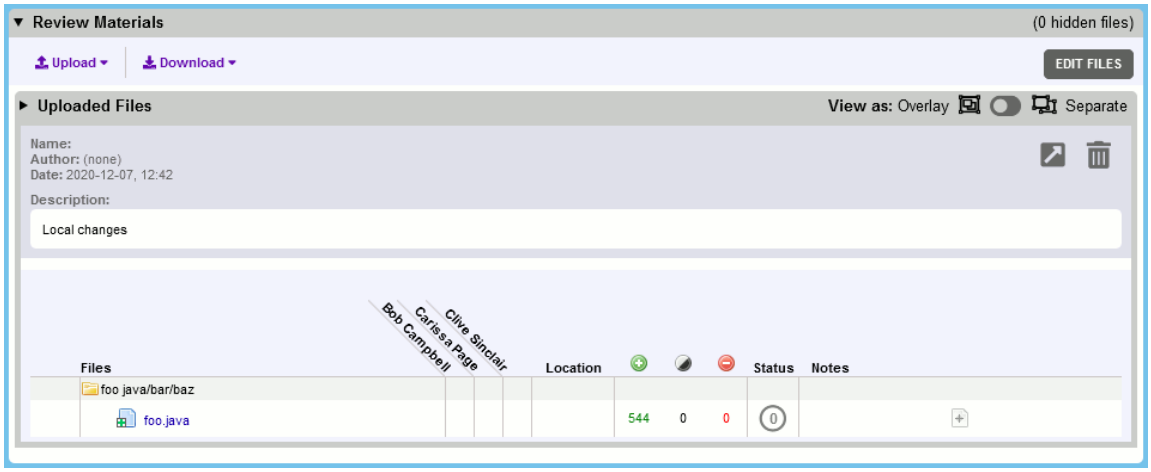
1. Scroll to the Review Materials section.
2. Click the **Upload** button.
3. Select **Files** from the drop-down list.



4. In the ensuing dialog, choose one or more files you want to attach and then click **Upload**.

   **Tip:** Alternatively, you may just drag your local files and drop them to review web page.

Once you have attached anything to a review, it will be listed in the Review Materials section.




Most Collaborator users do not use the Web Client for attaching source files, reserving it for adding supporting documents. Instead, most of our users opt to use the GUI Client, the Command-Line Client, or add source files directly from Eclipse, Visual Studio or from P4V or P4Win clients for Perforce.

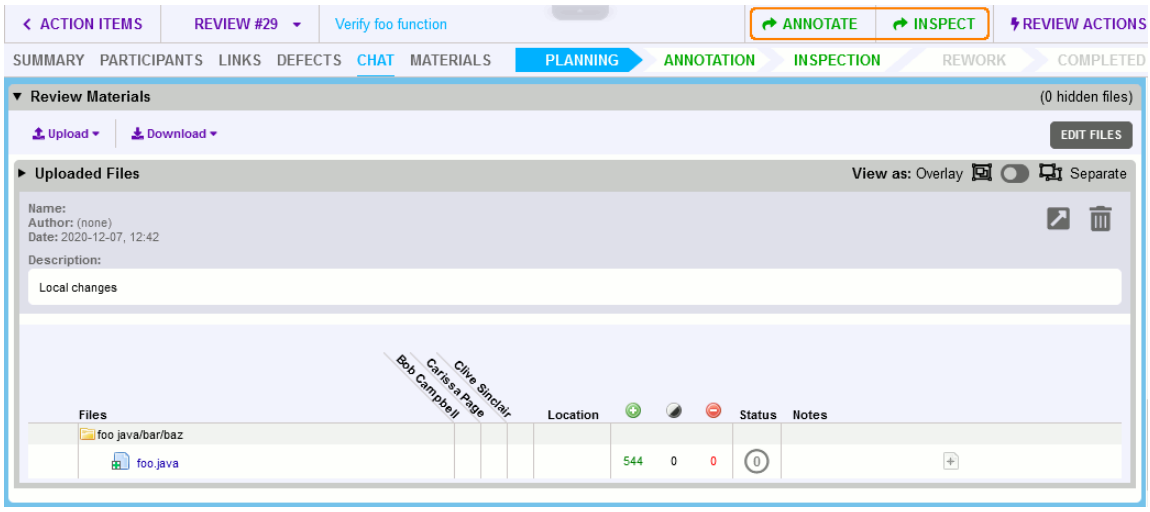
These clients need to be installed on your computer and properly configured, however they provide an extensive integration with Source Control Systems and/or IDEs.

To learn how to add review materials via these clients, please read the respective sections in the manual: GUI Client 496, Command-Line Client 506, Eclipse Plug-in 525, Visual Studio Plug-in 566 or P4V / P4Win Integration 786.

## 8. Start the review

Click **Inspect** button in Review Summary screen header to start a review.

Alternatively, you can click **Annotate** button to apply the current changes, and proceed to optional Annotating phase. In this phase all participants that you have added will be notified about this review and will be invited to add their comments and review materials. The review will not begin until you select the **Inspect** button.




Other participants will receive notifications about your new review, and it will be displayed in their Action Items 333 list.

## Where to Go Next

To learn how to take part in someone else's reviews as a reviewer or observer, see Participating in Reviews 48.

To learn how to get to know about the issues other participants have found in your review, see Reworking Defects 53.

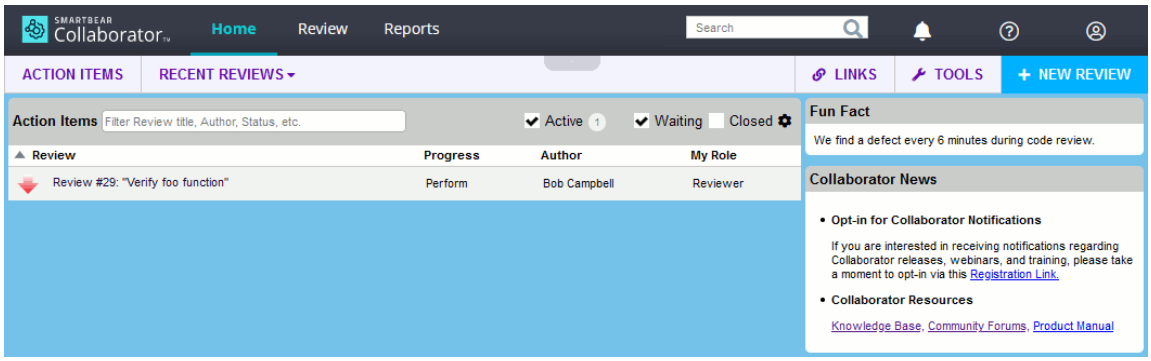
## 2.2.2 Participating in Reviews

When someone invites you to a review, you receive an email notification. To take part in this review you need to open Collaborator Web Client.

### 1. Log into the Web Client

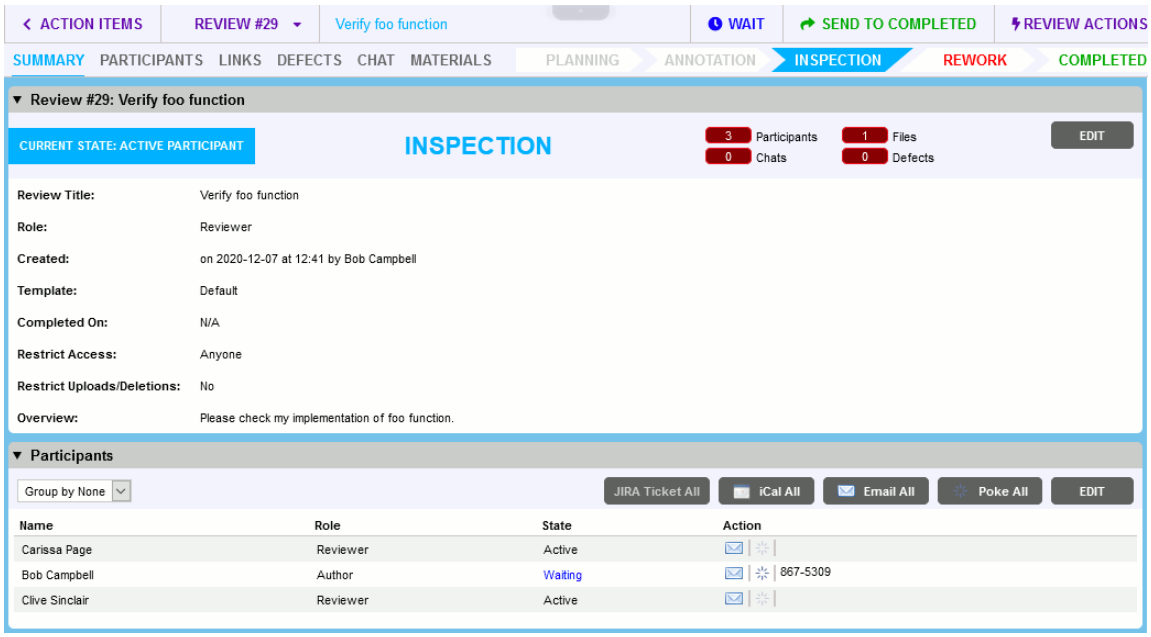
Start your favorite web browser and navigate to Collaborator Web Client URL. If asked, enter your user name and password in the login screen.

In the Action items list on the Home screen, you will see a new incoming review.




## 2. Open Review Summary screen

Click the review in the Action items list to open it in the Review Summary 345 screen where you can see the details about this review.



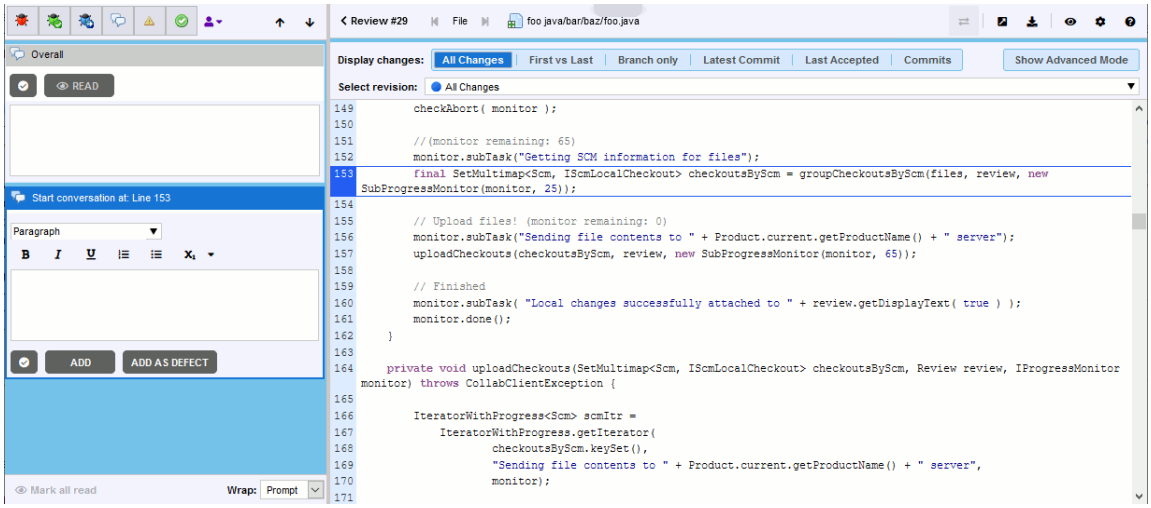

Notice that you are the active participant of this review and the review is in the Inspection phase.

## 3. Inspect review materials

Scroll to the Review Materials section (or click **Materials** in the toolbar) and click the uploaded file (or the first of several uploaded files). The Diff Viewer 364 will open.

Diff Viewer displays the contents of the chosen file and allows to add your comments and defects on this file. The Viewer appearance changes a little depending on a type of the current file.

If a version history is available for the file (for example, when it was added from a source control system or when it was reworked by the author), the Diff Viewer shows the current and previous versions of the file and highlights the differences between them.

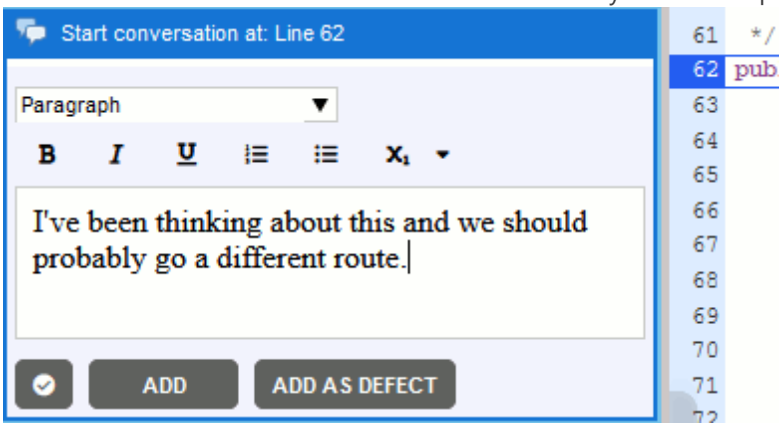



## 4. Add your comments, look for and verify defects

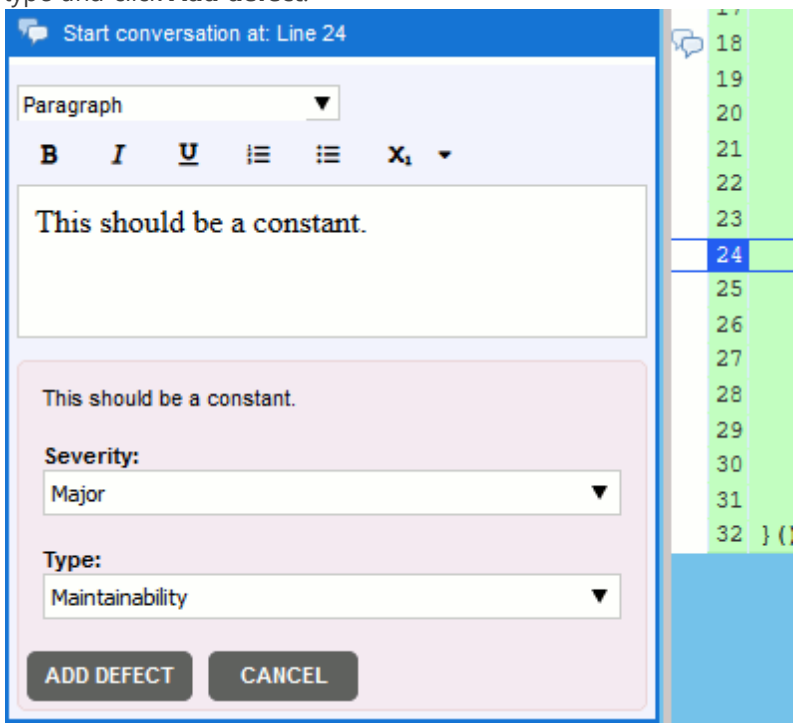
While reviewing, you most likely find some issues with the materials or would like to ask or tell something to the author. For this purpose you can add comments and defects.

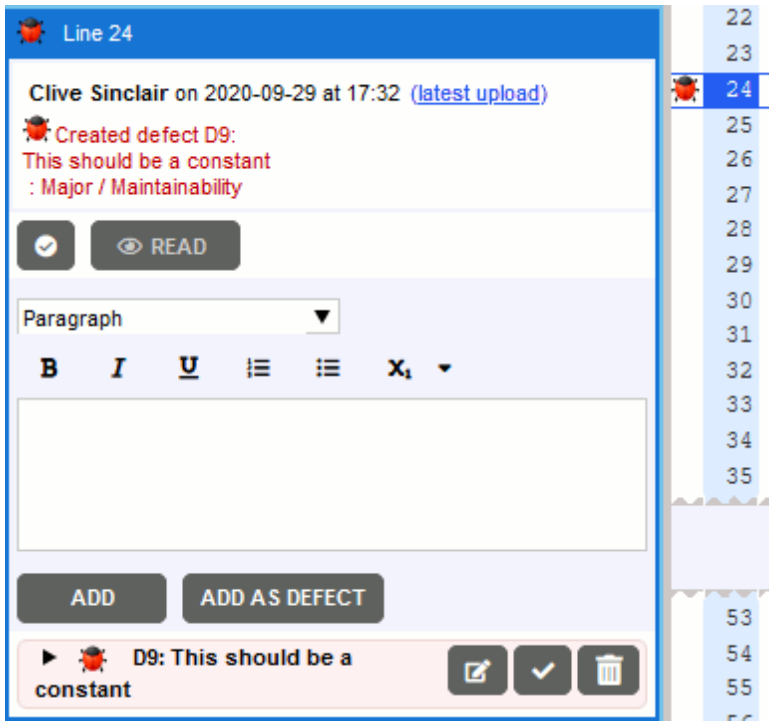
If you are reviewing a file that has already been reworked, then you will need to verify whether the defects have been fixed.

- To add a comment, click on the line (in source code and text files), cell (in spreadsheets) or area (in documents and images) you want to comment on, type your comment and click **Add**. Comments are threaded into a conversation that stays tied to a specific line, cell or area.

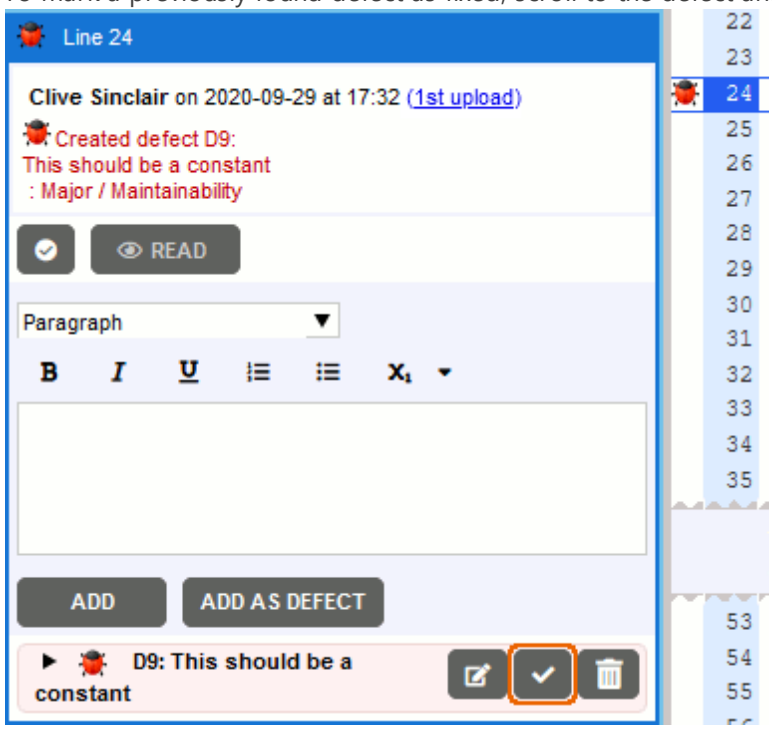


- To add a defect, click on the line, cell or area to which you want to attribute the defect, enter issue description and click **Add as defect**. In the ensuing pane specify defect severity level and type and click **Add defect**.



- To mark a previously found defect as fixed, scroll to the defect and press **Mark fixed**.




Besides creating comments and defects that relate to a specific line or area you can also create comments and defects that relate to entire files (in the Overall section of Diff Viewer) as well as comments and defects that relate to the whole review (in the Chat section of the Review Summary screen). See Review Chats 429 for more information on making conversation during reviews.

When you have examined the current file, you can proceed to another file that was uploaded to the review (if any). To view the next or previous files, you may use the File navigation buttons on the Diff Viewer toolbar 367.

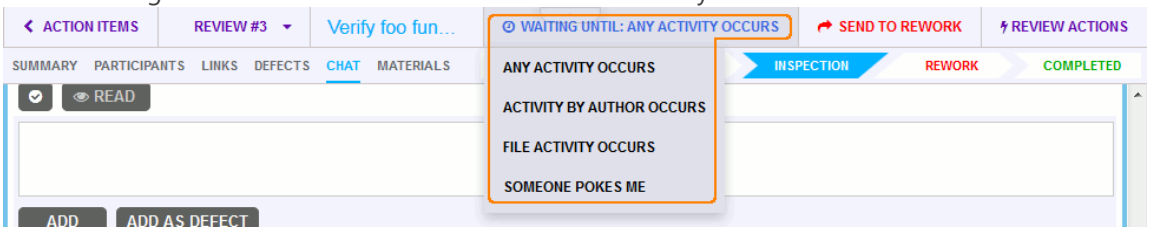

Having inspected all the files that were uploaded to the review, return back to the Review Summary screen.

To return back to Review Summary screen, click a button with review number (Review #29 in image above) on the Diff Viewer toolbar.

## 5. Finish inspection

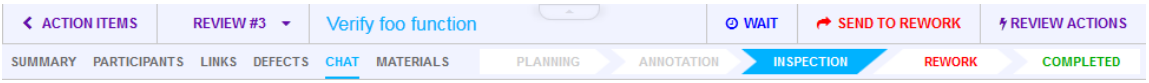
In Review Summary screen header you can find several review action buttons: Wait, Send to rework, (or Send to completed) and so on.

- To signal other participants that you have temporarily stepped out from the review but plan to return, press **Wait.**
  In the ensuing list select when Collaborator should re-invite you to the review.

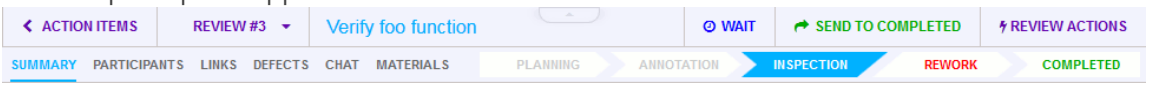



*Wait Options*

- To indicate that you have found some defects that need to be fixed, press **Send to Rework**.
  This will change the review phase to Rework and return it back to authors so that they can fix the found defects.




- To indicate that you approve the changes in the uploaded materials, press **Send to Completed**.
  When all participants approve the review it will be closed.




  A review cannot be closed until all its defects are resolved: either marked as fixed, deleted, or tracked externally[442].

## Where to Go Next

To learn how about correcting the found defects as an author, see Reworking Defects[53].

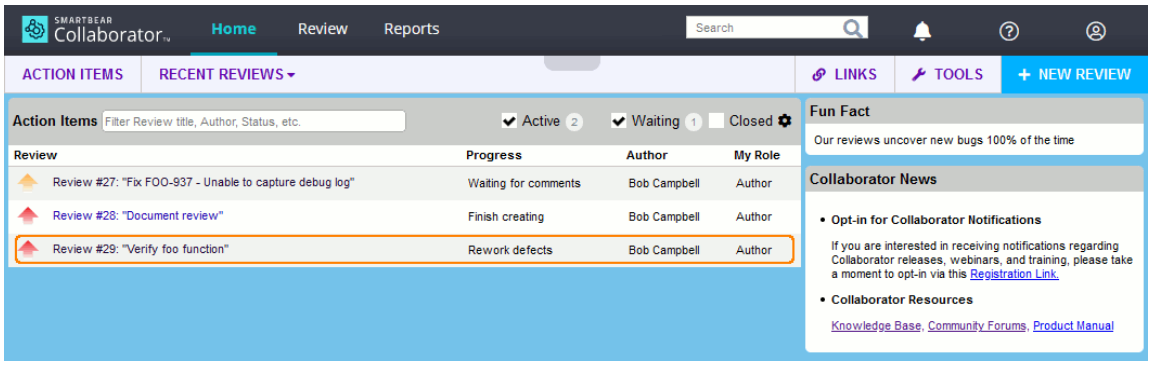
### 2.2.3 Reworking Defects

When one of review participants has noticed some issues with your work and created defects, the review is returned back to you.

#### 1. Log into the Web Client

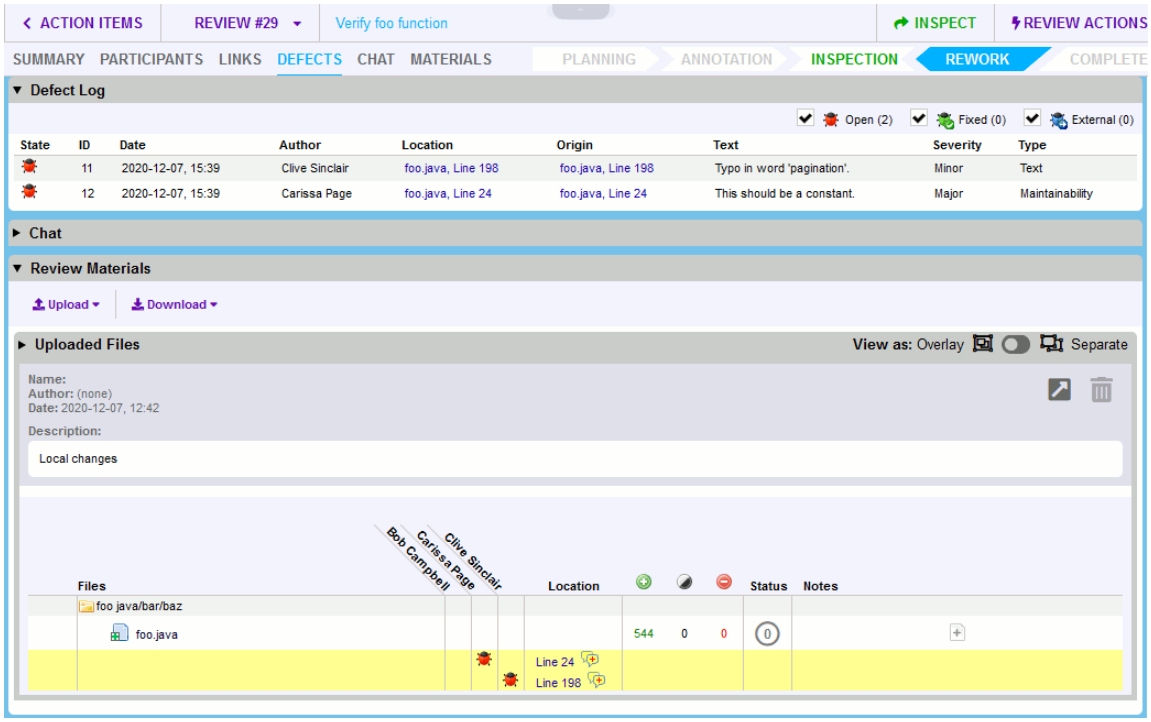
Start your favorite web browser and navigate to Collaborator Web Client URL. If asked, enter your user name and password in the login screen.

In the Action Items list on the Home screen, you will see your review that was returned. It will have the "Rework defects" state. You will need to fix the defects found by other participants.




## 2. Open defect log of the Review Summary screen

Click the review to open the Review Summary screen and scroll to the Defect Log section (or just click Defects in review toolbar).

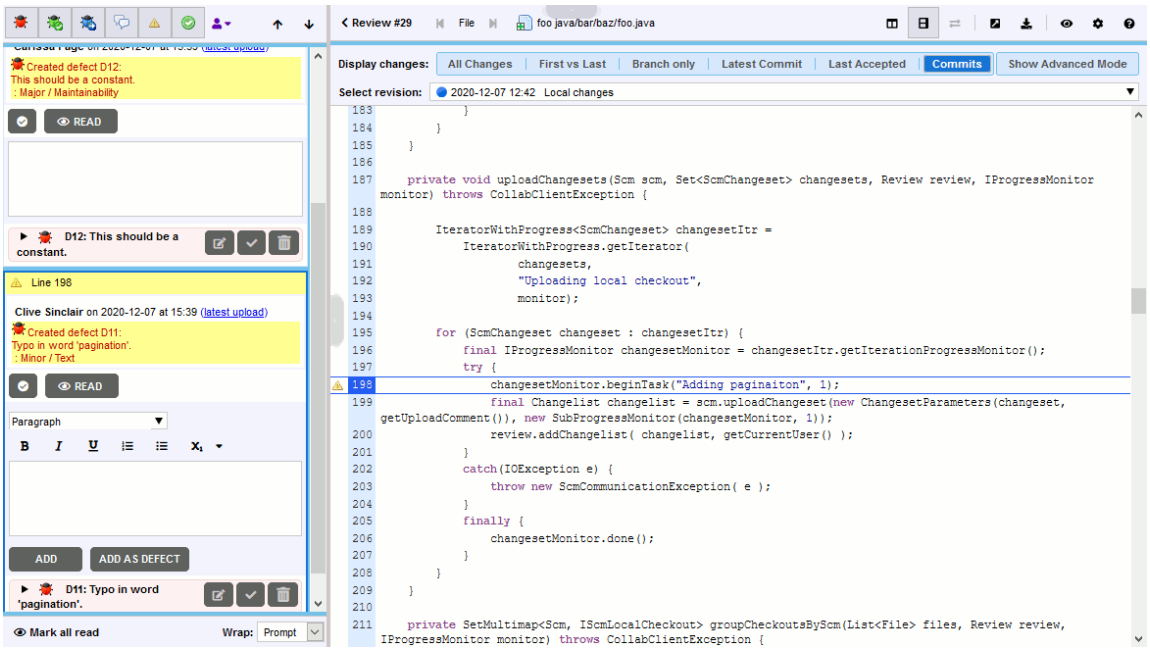



This section lists all found defects, their location and description. The Review Materials section also displays information about defects and comments, however groups it per files.

## 3. Get information about a particular defect

Click the Location link in the Defect Log or in the Review Materials sections.

This will open the Diff Viewer and navigate to the defect.




Read descriptions of the defects and reviewer comments. Reply if needed.

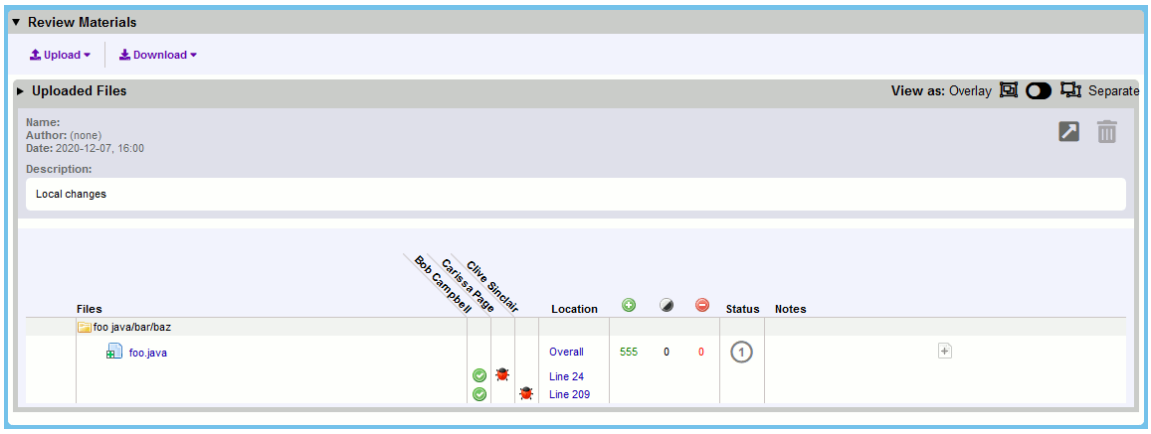
To denote that you agree with the defect or comment, press **Mark Accepted**. To indicate that you have read the comment or defect description, press **Read**.

When you have examined the current file, you can proceed to another file (if any). To view the next or previous files, you may use the File navigation buttons on the Diff Viewer toolbar:


To return back to Review Summary screen, click a button with review number (Review #29 in image above) on the Diff Viewer toolbar.

## 4. Upload the reworked files

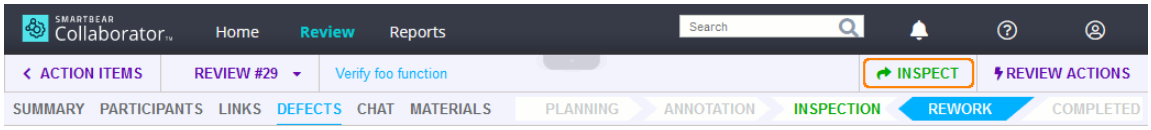
Correct the defects on your computer and then upload the corrected version of the file(s) to the review.




The Status column in the Review Materials for the uploaded file will increase to indicate that the files was reworked.

### 5. Request to verify your corrections

When you have uploaded the corrected versions for all files, press **Inspect** button in Review Summary screen header. Other review participants will be notified that you have reworked the review materials and want them to verify your changes.




A review can go through multiple inspection and rework phases before it is completed.

# 3 Collaborator Server

This chapter describes how to setup and manage the server component of Collaborator. It is useful for system administrators. Regular users of Collaborator should refer to Web Client 312 or Desktop Clients 484 sections instead.

**Collaborator server** process acts as the hub, manager, and controller of information. The server has a web-based user interface where users and administrators can do everything — create and perform reviews, configure personal and system-wide settings and run reports. The server uses a database to store all data and configuration.

## In This Section

- Installation 57
  Covers different aspects of server and database installation, backup and upgrade.

- Server Administration 88
  Covers different aspects of server management and administration.

- Collaborator Settings 178
  Describes how to configure the Collaborator server.

## Related Topics of Interest

- Web Client 312
  Describes the web user interface of Collaborator.

- Desktop Clients 484
  Describes GUI Client, Command-Line Client, plugins for Eclipse, Microsoft Visual Studio and other clients for desktops.

# 3.1 Installation

This chapter covers different aspects of server and database installation, backup and upgrade.

## In This Section

- System Requirements 57
  Describes the minimum and recommended requirements to install and run Collaborator server.

- Database Installation and Configuration 59
  Explains how to install and prepare different databases to work with Collaborator.

- Server Installation 70
  Explains how to install Collaborator server.

- Server Upgrades 83
  Explains how to upgrade an existing Collaborator server.

- Platform-Specific Notes 87
  Contains additional server-related information that is specific to certain operating systems.

### 3.1.1 System Requirements

This section describes the minimum and recommended requirements to install and run Collaborator server, as well as some steps that it is prudent to take before attempting to install Collaborator.

# Server Requirements

Please ensure that you meet or exceed the following requirements to run the current version of Collaborator server.

**Configuration for servers with 20 or fewer concurrent users:**

- Modern processor with 2 cores, 2.5Ghz or better. 64-bit platform is required for document reviews 363.
- 4GB free RAM
- 100GB free storage space
- Java: OpenJDK 11 (recommended), or Oracle JRE/JDK 8, 9, or 11. See Java Compatibility Matrix 1243 for more details.
- Windows, Linux, Solaris or BSD
  (If Windows, *must* be a server edition, not workstation edition) On Windows Server 2008, you need to install Oracle JRE beforehand.
  On *nix platforms, you may need to install Microsoft's TrueType core fonts in order to render Web UI correctly.
- Separate database server 59 is recommended. (Trial users may install on our embedded database.)

**Configuration for servers with 50 or fewer concurrent users:**

- Modern server class processors with at least 4 cores in total, 2.5Ghz or better. 64-bit platform is required for document reviews 363.
- 12GB free RAM. We recommend increasing the memory available to Collaborator server (instructions here 1241)
- 100GB free storage space
- Java: OpenJDK 11 (recommended), or Oracle JRE/JDK 8, 9, or 11. See Java Compatibility Matrix 1243 for more details.
- Windows, Linux, or Solaris
  (If Windows, *must* be a server edition, not workstation edition) On Windows Server 2008, you need to install Oracle JRE beforehand.
  On *nix platforms, you may need to install Microsoft's TrueType core fonts in order to render Web UI correctly.
- Separate database server 59 is required

- Low latency link to database server
- Access from all possible clients to server port without a proxy
- Some kind of drive backup hardware (Example: RAID 0)

For servers with greater than 50 concurrent users, please contact our support team[32] so that they can help with your implementation plan.

## Preparing your Collaborator server

- Install the latest stable build of supported Java version.
- Create a database user with permission to connect via TCP/IP from Collaborator server.
- Create an empty schema on database server in which aforementioned database user can create, alter, read, write, update, and delete tables.
- Make sure that you have access to your intended server port (default 8080) through any firewall you may use.
- If you intend to use LDAP or Active Directory for authentication, please review the relevant section of our manual[119]. You may need to ask your LDAP or Active Directory administrator to create a dedicated user for the Collaborator server.

### 3.1.2 Database Installation

Collaborator supports several databases:

- **Embedded**[60]. A zero-configuration embedded database useful for trials.
- **MySQL**[60] versions 5.6, 5.7, 8.0.
- **Microsoft SQL Server**[64] versions 2012, 2014, 2016, 2017, 2019.
- **Oracle**[67] versions 11gR2, 12c, 18c and 19c.

For trials, the zero-configuration embedded database is the easiest way to get started. MySQL is the best choice if you need a free, open-source database.

Collaborator server must be able to access your database server via TCP/IP. Collaborator requires an empty schema on the database server in which a dedicated database user can create, alter, read, write, update, and delete objects.

A built-in database migration tool[100] allows you to move between any of the supported databases, so you can always get started on one and switch to another later on.

***Note:*** *MySQL database is supported in Collaborator Team and Collaborator Enterprise. Oracle and SQL Server databases are only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page*[3].

For more about the database format, see Database Schema[908].

#### 3.1.2.1 Zero-Configuration

The server installer comes with support for the Hypersonic embedded database. As of version 11.3, we install the HSQL 2.3 database. This in-memory database is perfect for trials where you want to get up-and-running as easily as possible. This database performs automatic daily backups. There is no need to use SQL Server, Oracle or MySQL unless you have specific needs.

Using this database requires no configuration. When the server installer[70] asks for the database type, simply select "Embedded".

If you find that the embedded database is not going to fit your needs, the standard database migration technique[100] works with all databases, so it is possible to move all your data from the embedded database into a "real" database at any time. In addition to using the database migration technique, to move to a "real" database, you will also need to reconfigure the Collaborator server by running the Server Installation program[70].

#### 3.1.2.2 MySQL

MySQL is a popular enterprise-class, open-source database. This is your best choice if you need a free, open-source database.

## Supported MySQL Versions

Collaborator supports MySQL versions 5.6, 5.7, and 8.0.

***Note:*** *MySQL database is supported in Collaborator Team and Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page*[3].

## Download MySQL Components

**Required components**

- MySQL Community Server - database server,
- Connector/J - database driver for Java platforms. Supported driver versions are 5.1.x - 8.0.22

When upgrading to Connector/J 8.0.x be aware that it performs time offset adjustments, so your server should either use one of canonical time zones or use the `serverTimezone` property to specify time zone.

**Optional components**

- MySQL Workbench - GUI tool for graphical server administration. A GUI tool alternative to MySQL Command Line Client, which is a part of part of the database server.

For downloads and documentation for all platforms, see the MySQL Developer Zone web site.

## Install and Configure Database

First, install the Database Server component. Once you have installed the server, the last screen will give you the option to Configure the MySQL Server. If you select this, it will open a new wizard for configuration. Two parts in the configuration that are especially important: **Port Number** and **Root Password**. You will need to know both while getting your database setup and while configuring Collaborator to work with your database. For information on configuring your database server, go to the MySQL Documentation and select your MySQL version.

It is important to make sure you are using InnoDB tables. Using InnoDB tables will scale better for multiple users than MyISAM. For questions regarding configuring you MySQL database tables, see https://dev.mysql.com/doc/mysql-windows-excerpt/8.0/en/mysql-installer-workflow-server.html.

Collaborator does not support the non-default values of the auto_increment_increment and auto_increment_offset MySQL system variables.

For Collaborator, you have to create a separate database, since the MySQL Server install will not create a database for you. This is a manual step that you need to do.

You can create a database using the MySQL Command Line Client, or using MySQL Workbench. If you plan to use the MySQL GUI client, it is important to know that the MySQL GUI Tools Bundle is reaching end of life. The instructions given in this documentation use the functionality in the current GUI client, MySQL Workbench.

The steps below for creating a database were written using MySQL Workbench 5.2.16 and are subject to change. As always, the most reliable source for database creation steps is MySQL Workbench Documentation.

1. Open MySQL Workbench.
2. Create a connection to your database server.
   a) From the Home screen, under Server Administration, select New Server Instance.
   b) Follow the steps in the New Server Instance Wizard.
3. Create the database.

a) Go back to the Home tab.

b) Under SQL Development, select the database server connection you provided in step 2.

c) In the Object Explorer, right-click and select Create Schema.

d) A screen will appear and prompt you to provide a schema name. Once you have done that, click Apply.

e) You will be shown the command used to create the schema, and given an option to edit it.

f) Once the create command is as you want it, click Apply.

g) You will be returned to the screen you saw in step 3d. Click Finish.

We recommend that you create a user specifically for your Collaborator database rather than using the super-user, root. To do this in MySQL Workbench, see the documentation at MySQL Schema Privileges.

All of these changes go into effect immediately. You do not have to restart the MySQL server for changes to take effect.

When installing 79 Collaborator on a server, the installation wizard will prompt for the MySQL server host name, TCP/IP port, database name, user name, password and path to JDBC driver. The installer will report any connectivity errors. When you visit the web page for Collaborator it will detect that you have a new database and will create all tables, indexes, and views for you automatically.

The following SQL script can be used to create a database and a database user for Collaborator and configure the required permissions for that user (be sure to change the database name, user login and password as appropriate):

```
CREATE DATABASE IF NOT EXISTS ccollabdb CHARACTER SET utf8 COLLATE
utf8_unicode_ci;
CREATE USER 'collabuser' IDENTIFIED BY 'password';
GRANT
 ALTER,
 CREATE,
 CREATE TEMPORARY TABLES,
 CREATE VIEW,
 DELETE,
 DROP,
 EXECUTE,
 INDEX,
```

```
 INSERT,
 LOCK TABLES,
 SELECT,
 UPDATE
ON TABLE ccollabdb.*
TO collabuser;
FLUSH PRIVILEGES;
```

## Technical details and limitations

- The MySQL native restore feature does not automatically clear the database before restoring from backup. You can use the MySQL native restore feature, but you **must** manually drop all tables in the database before you run the restore. By doing this, you will ensure an exact reproduction of your backed-up database. Option two is to use the alternative method 99 provided by Collaborator.

- Connector/J have different driver class names in versions 5.x and 8.x. During installation 79 the wizard assigns the appropriate name based on the JDBC driver you have specified. To change the driver name afterwards, you may either re-run the installation wizard, or modify the Collaborator's ROOT.xml manually, as described below:

    1. Download the Connector/J driver.

    2. Stop the Collaborator server, if it is running.

    3. Open <Collaborator Server>/tomcat/conf/Catalina/localhost/ROOT.xml file and locate the `Resource` element.

    4a. For **Connector/J 5.x** specify `driverClassName="com.mysql.jdbc.Driver"`:

```
<Resource driverClassName="com.mysql.jdbc.Driver" factory="org.
apache.tomcat.jdbc.pool.DataSourceFactory" maxIdle="20"
maxActive="100" maxWait="10000" name="/jdbc/collabserver"
password="<DB_password>" scope="Sharable" testOnBorrow="true"
type="javax.sql.DataSource" url="jdbc:mysql://localhost:3306/
<DB_schema>?useServerPrepStmts=false&amp;useUnicode=true&amp;
characterEncoding=UTF-8&amp;autoReconnect=true" username="<DB_user>"
validationQuery="SELECT 1"/>
```

    4b. For **Connector/J 8.x** specify `driverClassName="com.mysql.cj.jdbc.Driver"` and `serverTimezone` property:

```
<Resource driverClassName="com.mysql.cj.jdbc.Driver" factory="org.
apache.tomcat.jdbc.pool.DataSourceFactory" maxIdle="20"
maxActive="100" maxWait="10000" name="/jdbc/collabserver"
password="<DB_password>" scope="Sharable" testOnBorrow="true"
type="javax.sql.DataSource" url="jdbc:mysql://localhost:3306/
<DB_schema>?useServerPrepStmts=false&amp;useUnicode=true&amp;
characterEncoding=UTF-8&amp;autoReconnect=true&amp;serverTimezone=UTC
" username="<DB_user>" validationQuery="SELECT 1"/>
```

**Tips:** The ampersand (&), needs to be SGML encoded to "&" because the configuration file is an XML document.

The Connector's `serverTimezone` value should coinside with the timezone in which your MySQL server is running, and it should be a valid Java timezone identifier.

5. Go to the <Collaborator Server>/tomcat/lib folder.

6. Copy the mysql-connector-java-*.jar file that comes with Connector/J to that folder.

7. Start the Collaborator server.

#### 3.1.2.3 SQL Server

## Supported SQL Server Versions

Collaborator supports Microsoft SQL Server 2012, 2014, 2016, 2017 and 2019.

**Note:** *SQL Server is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

## Download SQL Server Components

- SQL Server - database server,
- SQL Server JDBC Driver - database driver for Java platforms.

There are various JDBC drivers that can be used for various combinations of SQL Server and Java platform:

- JDBC Driver 6.2 – supports SQL Server 2008 - 2017 and Java 7 and 8

- JDBC Driver 6.4 – supports SQL Server 2008R2 - 2017 and Java 7, 8 and 9
- JDBC Driver 7.0 – supports SQL Server 2008R2 - 2017 and Java 8 and 10
- JDBC Driver 7.4 – supports SQL Server 2012 - 2019 and Java 8, 11 and 12

Please see Microsoft JDBC Driver for SQL Server support matrix for detailed information on selecting the appropriate driver.

## Install and Configure Database

You need to create an empty database for Collaborator. You can do this, for example, in SQL Server Management Studio.

When installing 79 Collaborator on a server, the installation wizard will prompt for the SQL Server host name, TCP/IP port number, the database name, user name, password and path to JDBC driver. The wizard will report any connectivity errors.

When you log in to the Collaborator Web Client for the first time after installation, it detects that you have a new database and will create all tables, indexes, and views for you automatically.

**Important notes:**

- When creating the database, be sure to select a case insensitive collation so that the case of column names (and user names and text searches) does not matter.
- It is recommended that you create a user account just for Collaborator and give this account at a minimum `db_owner` permissions to Collaborator database and no access to other databases.
- We recommend using a native SQL Server account authentication instead of Windows-based authentication, because the Collaborator service might not be running under a normal Windows-based login. SQL Server uses its native authentication by default, so no additional actions will be required. If you decide to use Windows-based authentication, you may need to change the Collaborator settings – see below 66.
- Collaborator uses the TCP/IP protocol for data exchange. By default, in SQL Server this protocol is disabled. To enable it, open the SQL Server Configuration Manager and navigate to **Network Configuration > Protocols**. Make sure TCP/IP is enabled.
- By default, SQL Server is configured to use dynamic ports, which means that the port used is changed each time the service is restarted. To use a static port instead, open the TCP/IP settings and change the IPAll setting value. Empty the **TCP Dynamic Ports** box and specify the desired port number in **TCP Port**.

## Use Windows Authentication

We recommend using a native SQL Server account authentication instead of Windows-based authentication, because the Collaborator service might not be running under a normal Windows-based login. Besides, Windows authentication is not available on *nix platforms. However, if you want to use Windows account, you may need to change the Collaborator connection settings:

1. After downloading the driver package from one of the links above, extract the files.

2. Copy the platform-specific sqljdbc_auth.dll to the tomcat subfolder of your <Collaborator installation folder>.

3. Open the following file in any text or XML editor --

```
<Collaborator installation folder>/tomcat/conf/Catalina/localhost/
ROOT.xml
```

4. Insert the `IntegratedSecurity=true` value into the url attribute of the `<Resource driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver" ... >` element that describes the JDBC configuration.

After the change, the element will look something like this --

```
<Resource driverClassName="com.microsoft.sqlserver.jdbc.
SQLServerDriver" ...
url="jdbc:sqlserver://SERVER:1433;databaseName=DATABASE-NAME;
IntegratedSecurity=true" ...
username="DOMAIN\USERNAME" validationQuery="select 1 ... " />
```

5. Restart the Collaborator service.

## Troubleshooting

- If you get the error "The TCP/IP connection to the host has failed. `java.net.UnknownHostException`", you probably need to enable TCP/IP for your database. See Setting Up the Database 65 above.
- By default, when Collaborator is installed, the service runs under the Local System account. If you are using SQL Server authentication rather Windows-based authentication, the Local System account must have the permissions necessary to communicate with the database. An error like "Error: Login failed for user 'domain\user$'". indicates that the Local System account does not have these permissions. To handle this situation, you can either grant your Local System account permissions to your database server, or edit the account the Collaborator service is using. To edit the account:
  - On your Collaborator server computer, from the **Control Panel > System and Security > Administrative Tools**, open **Services**.

- Find **ccollab-server** in the service list and stop the service.
- Right-click on ccollab-server, and select **Properties** from the context menu. This will open the Properties dialog
- Go to the **Log On** tab.
- Select **This account** and then specify the name and password for the account you want the Collaborator service to operate.

  **Note:** The account that you specify must have sufficient privileges on the machine, where the service is running (for instance, file system privileges for folder creation to allow the service create temporary subfolders in the Collaborator server installation folder).
- Click **Apply**.
- Start the ccollab-server service.

- If you are using SQL Server with named instances, it is important to know your port configuration. By default, named instances use dynamic ports. This means, every time SQL Server is restarted, it will search for available ports and assign one to your database. In this situation, you might have trouble finding the port upon which your database is running and could see errors when trying to connect to your database with Collaborator.

  Possible solutions:
  - If you are using named instances with dynamic ports, do not specify a port when providing Collaborator your SQL Server connection information. By leaving the port unspecified, a request will be sent to your SQL Server instance on port 1434 that will search for the port your database is running on and will then send that information back to Collaborator.
  - Alternatively, try using a static port. See Setting Up the Database 65 above.

#### 3.1.2.4 Oracle

## Supported Oracle Versions

Collaborator supports Oracle 11gR2, 12c, 18c and 19c.

***Note:*** *Oracle database is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

Oracle's extended support for the Oracle 11g database will end on 31st of December 2020 and Collaborator will no longer be able to support it.

## Download Oracle Components

- Oracle Database - database server,
- Oracle JDBC drivers - database driver for Java platforms.

## Install and Configure Database

Create a database for Collaborator. It is recommended that you also create a username/password pair just for Collaborator and give this account full access to the database and no access to other databases.

Install the CTX_DDL PL/SQL package on your DB server and grant the EXECUTE privileges on CTX_DDL to your Oracle user of Collaborator Server. This package allows use of the Oracle Text component instead of regular expression query for full-text searches 458 from the Web Client. By default, Oracle databases search based on regular expressions. If the default settings do not suit your requirements, you can change them on the Oracle side or through Collaborator admin UI. Once done, restart your Collaborator server to complete its upgrade.

When installing 79 Collaborator on a server, the installation wizard will prompt for the Oracle server host name, TCP/IP port (default is 1521), database service name (not the SID!), user name, password and path to JDBC driver. The installer will *not* report any connectivity errors.

> **Warning:** The database service name is not the same thing as the SID! This change was made by Oracle in version 9iR2.
>
> The database service name is fully-qualified, corresponding to GLOBAL_DBNAME in an .ora file. An example would be mysid.mydomain.com. The database service name is also sometimes referred to as "TNS alias" or "connect descriptor".
>
> The SID is the shorter name, corresponding to SID_NAME in an .ora file.

When you first visit the web page for Collaborator, it will detect that you have a new database and will create all tables, indexes, and views for you automatically, or give you an appropriate error message if there is a connectivity problem. Connectivity problems should be resolved by re-running the installer.

## Oracle Limitations

- **Oracle 11g server with non-english locale:** An error may occur when initializing Collaborator database on Oracle 11g servers with non-english locale. To workaround the problem:

  1. Append the following Java VM options 1232 to the `<collab server install dir>/ccollab-server.vmoptions` file:

```
-Duser.country=US
-Duser.language=en
```

  2. Restart Collaborator server to apply the changes.

3. Open Collaborator Web Client and proceed with database initialization.

- Double quote characters (") in custom field names may break Oracle reporting views. Custom field names become column headers in the views, and Oracle does not allow double quotes in column names. Because of this, Collaborator removes double quote characters from the names of custom fields when it creates reporting views for Oracle databases.
  If some custom field names differ only by double quotes this would result in an "ORA-00957: duplicate column name" error in server logs. To resolve the issue, you may either remove one of duplicate column names from the reporting view, or rename the custom fields to avoid coincidence.

- Loading large dump files to Oracle may take up to a full day to complete. This appears to be an issue with Oracle driver and is under investigation.

- On Oracle databases, Collaborator does not search the contents of custom fields by default (since this significantly reduces search performance). Instead, the search results page display additional fields that define in what areas to perform new search. In this panel you can enable searching in custom fields.

## Troubleshooting

We use the Oracle JDBC driver to connect to your Oracle database. The driver has a few undocumented behaviors that may come as a surprise. There are threads on Oracle's tech support forums about this.

Most of the problems arise in the `GLOBAL_DBNAME` field in your `SID_DESC` entry from your listener.ora file. A typical entry might look like this:

```
(SID_DESC =
(GLOBAL_DBNAME = mysid.mydomain.com)
(ORACLE_HOME = /appl/oracle1/product/10.2.0.1)
(SID_NAME = mysid)
)
```

Most other Oracle-based programs use the `SID_NAME` field to identify the database, but the JDBC driver uses `GLOBAL_DBNAME`. This would cause a connection error in the example above.

Also note that the database service name is not the same thing as the `SID`. This change was made by Oracle in version 9iR2. The installer asks for the database service name, not the SID. The database service name is also sometimes referred to as "TNS alias" or "connect descriptor".

Typically, this means you should use the `GLOBAL_DBNAME` in the installer (that is, mysid.mydomain.com) and not just the `SID` (that is, mysid).

### 3.1.3 Server Installation

Collaborator uses a stand-alone web server for a cross-platform, no-client, firewall-friendly user interface. Please review the System Requirements 57 section before continuing.

First, you need to download the correct installer for your platform from our website:

http://support.smartbear.com/downloads/collaborator/

To learn how to upgrade an existing Collaborator server, please see Server Upgrades 83.

In this section:



## The Graphical Installer

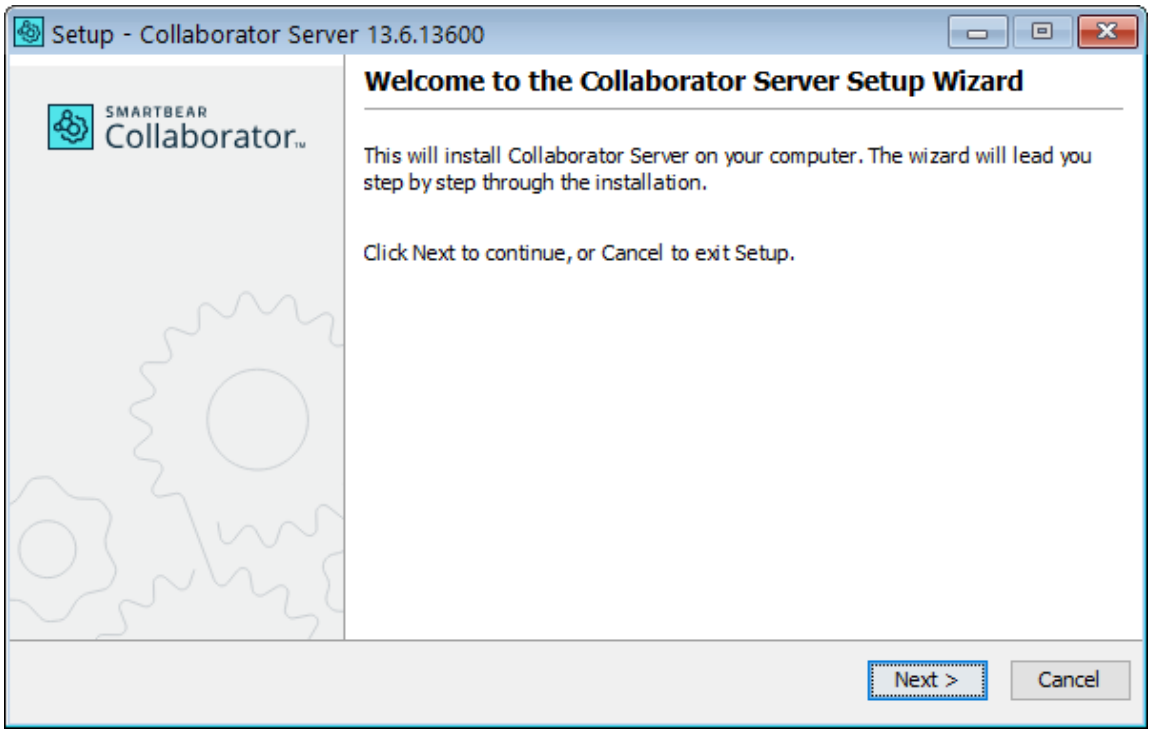
Installers can be run in a graphical, interactive mode or an automatic mode. For first installation, you will need to run the graphical version so you can select options.

On *nix platforms, you should grant execute permissions to the installer script and then launch it in shell:

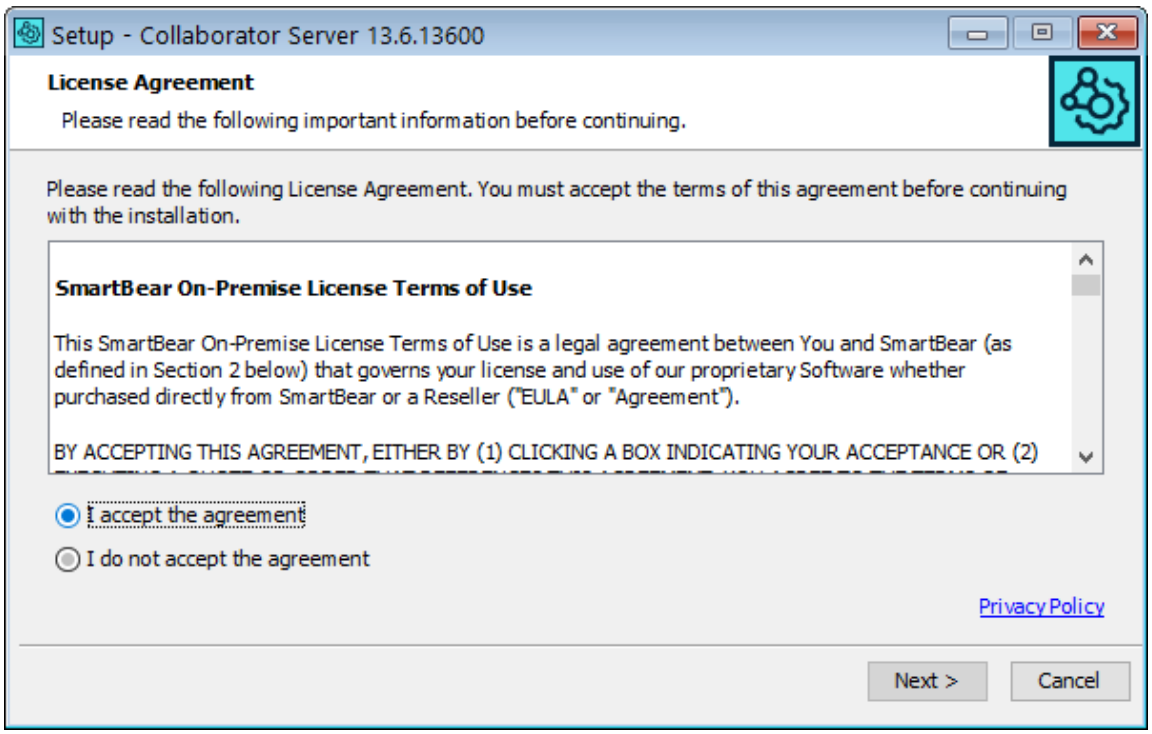
```
chmod +x ccollab_server_13_6_13601_unix.sh
sh ccollab_server_13_6_13601_unix.sh
```

## Normal Installation

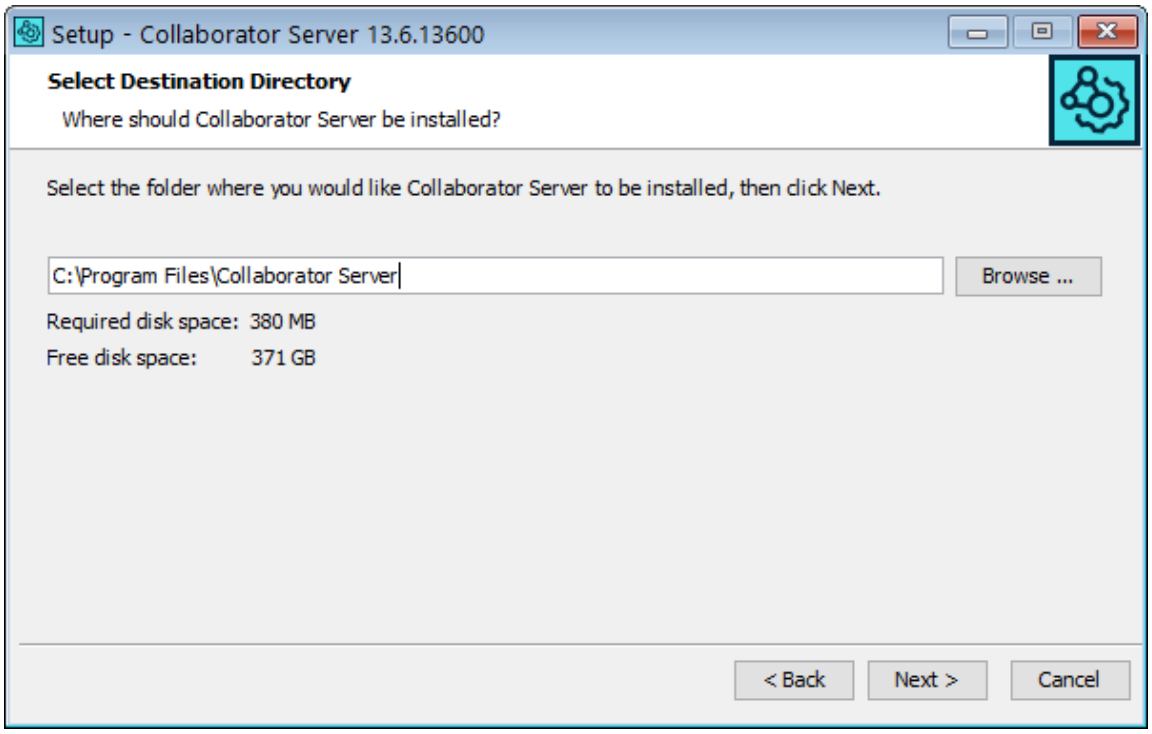
The first screen of the wizard lets you know it is working:




The second screen is the End User License Agreement (EULA):



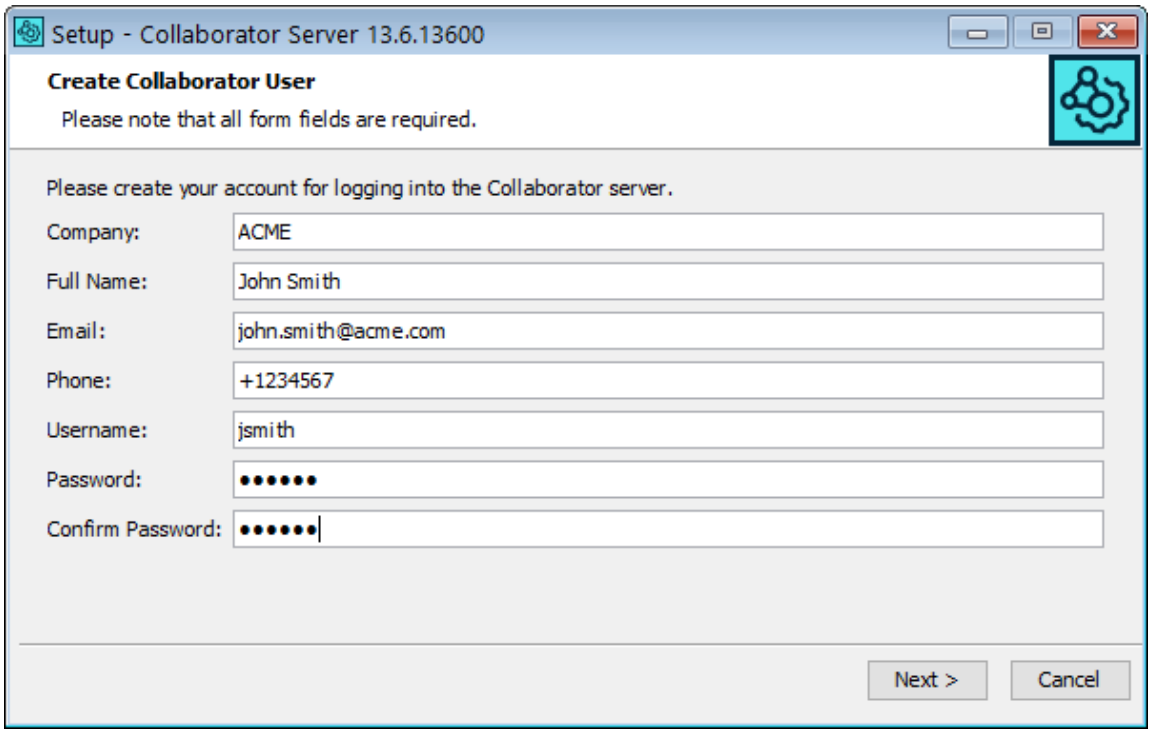

In the third screen, you can select disk location for the installation files.




This directory should be writable by the server because several dynamic files are created here including web server logs, temporary storage, and long-term storage for file content uploaded by users. All of these locations can be changed if it is critical that the installation location be read-only, but this requires significant work on the part of the administrator and makes upgrades more difficult.

Make sure at least 5 gigabytes of space is available in the named directory. Your users will need the space for file uploads. You can always move the file upload directory.

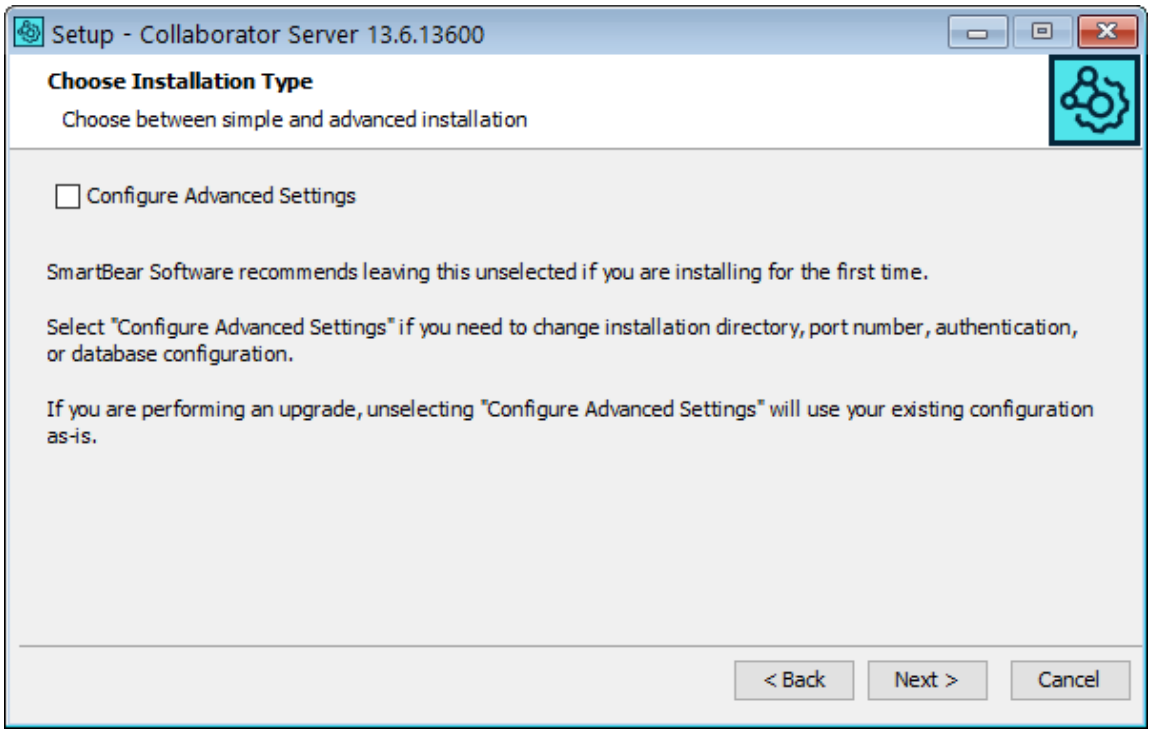
In the forth screen, the installer prompts to create the first user of your Collaborator server.



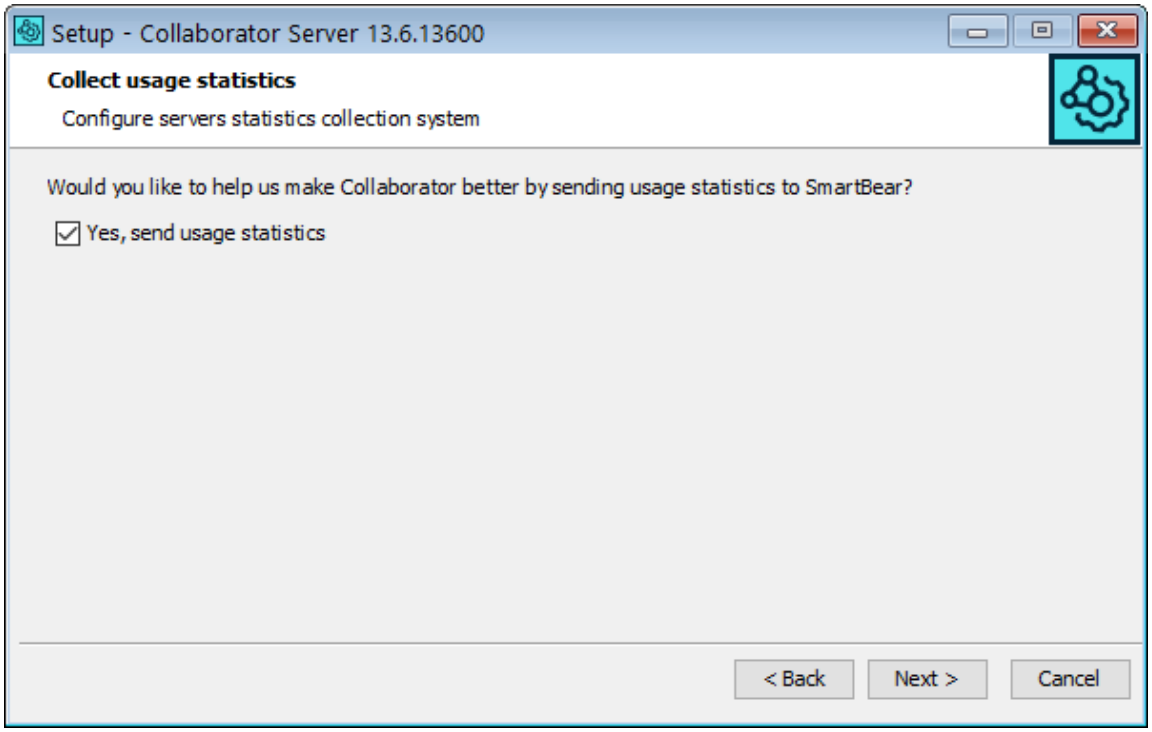

**Important:** The first user will have Administrator privileges on this Collaborator server.

Fill in company name, full name, email address, phone number, username and password. The contact information is primarily there for your users to know who to contact if they are having issues with Collaborator. SmartBear only uses this information to contact you when there is a critical issue with your Collaborator installation; we will not sell or share this information.

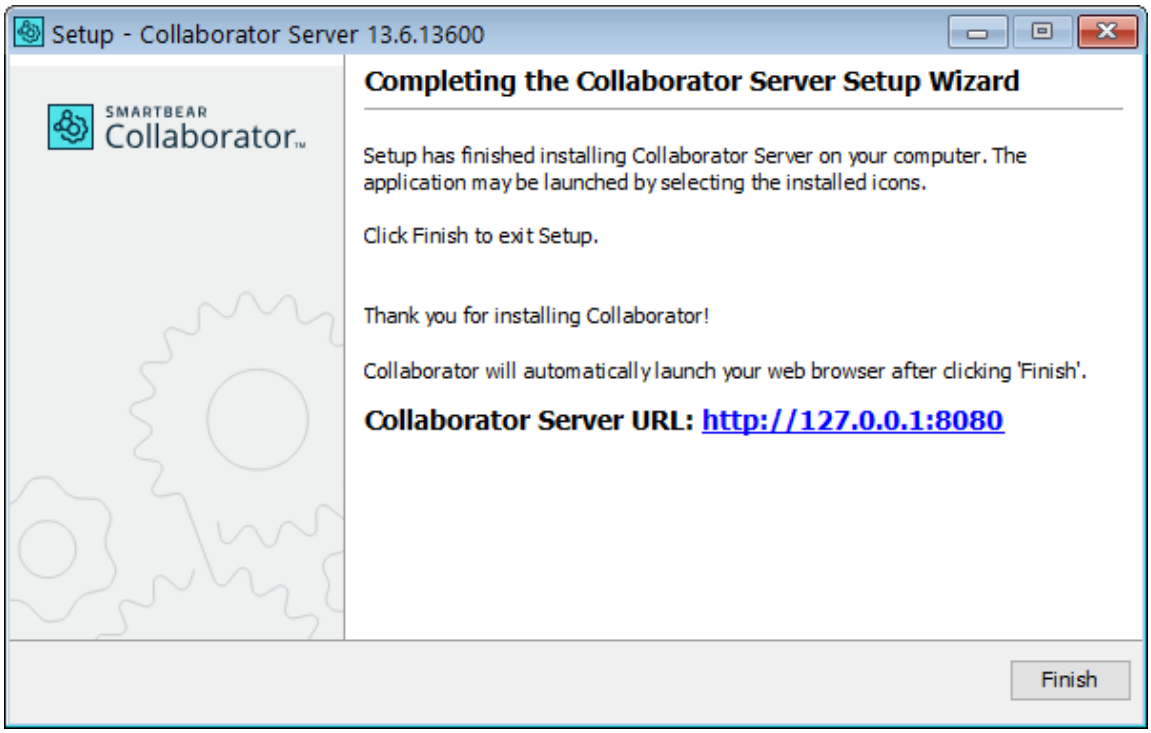
The next screen allows you to select whether to configure advanced options 75. By default, this checkbox is unselected and the default options are recommended for first-time and evaluation installations.




The next screen asks your permission to collect and send server usage statistics to SmartBear. To learn about our privacy policy, visit https://smartbear.com/privacy/.



After this, the installer copies the server files, stopping any existing server if necessary. Once the new files are installed, the server is started automatically, and the installer tells you it is finished:



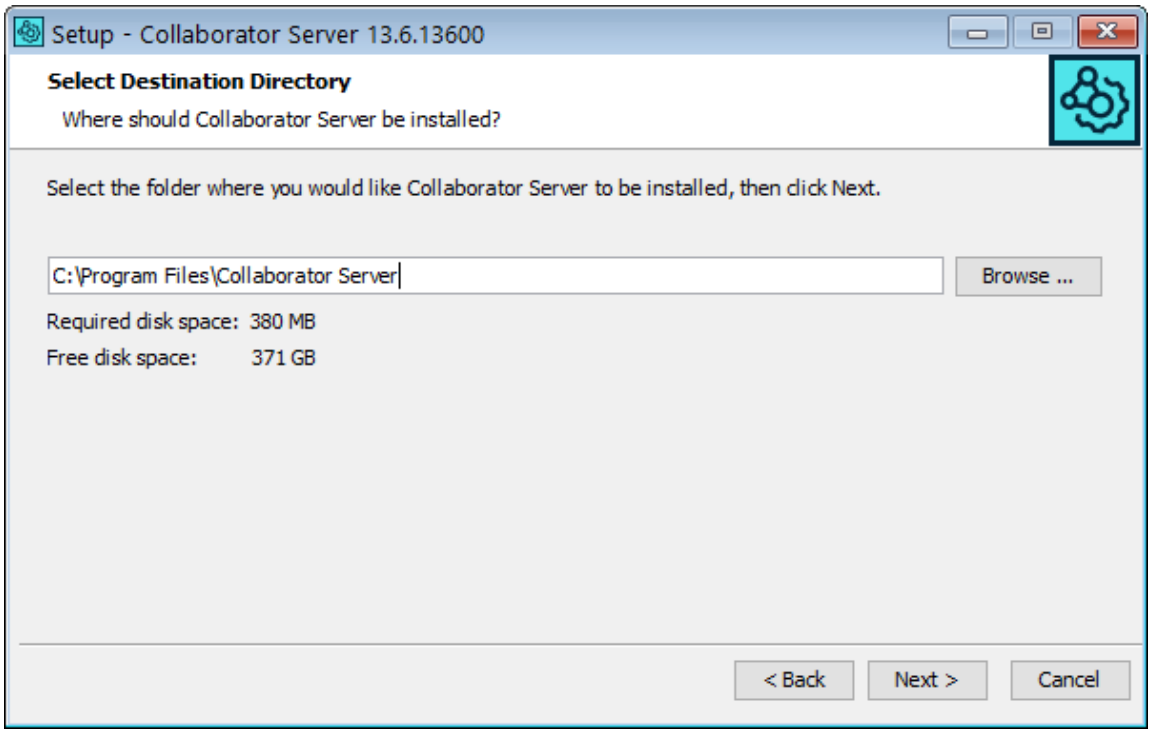

After you click Finish, a web browser will open, pointed at your installed server.

> **Warning:** If the server has not quite had enough time to fully start up, it might take a while for the web page to load and it might even fail to open. Just "refresh" the browser.

## Advanced Installation

Selecting the 'Configure Advanced Settings" checkbox, allows you to override the default settings and specify your own configuration.

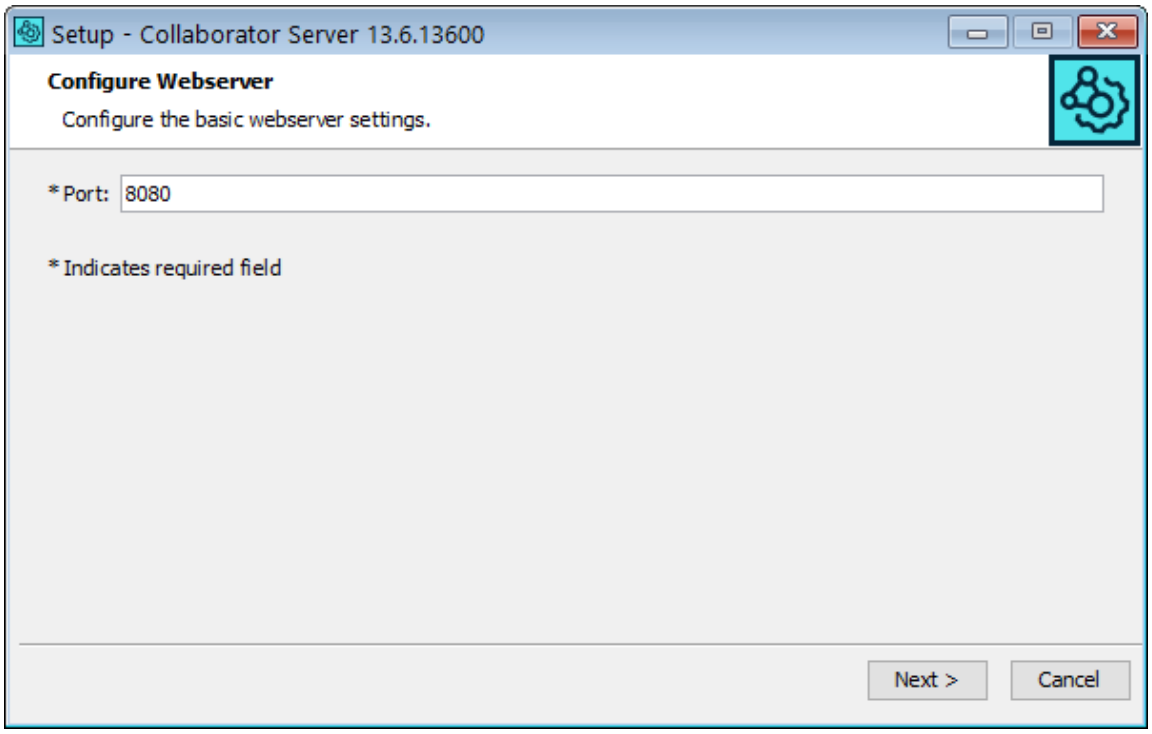
On the first step of advanced configuration, you can re-select disk location for the installation files.




This directory should be writable by the server because several dynamic files are created here including web server logs, temporary storage, and long-term storage for file content uploaded by users. All of these locations can be changed if it is critical that the installation location be read-only, but this requires significant work on the part of the administrator and makes upgrades more difficult.

Make sure at least 5 gigabytes of space is available in the named directory. Your users will need the space for file uploads. You can always move the file upload directory.

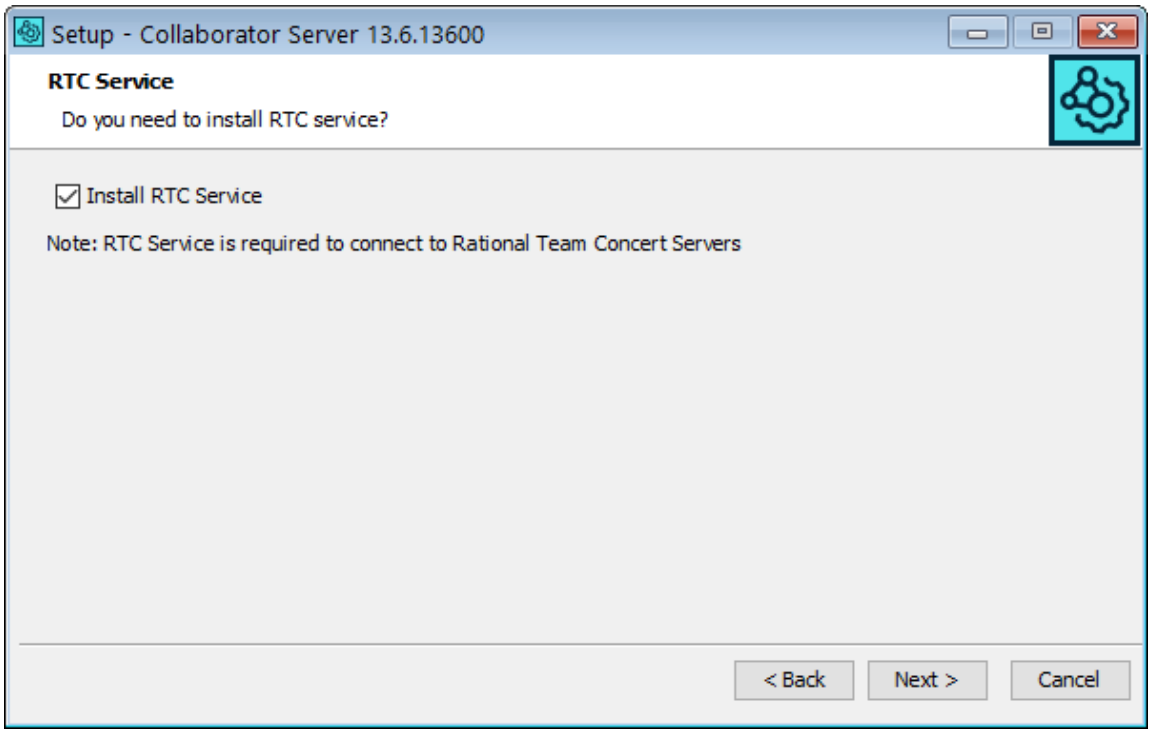
On the next screen, you can specify port number for your server:



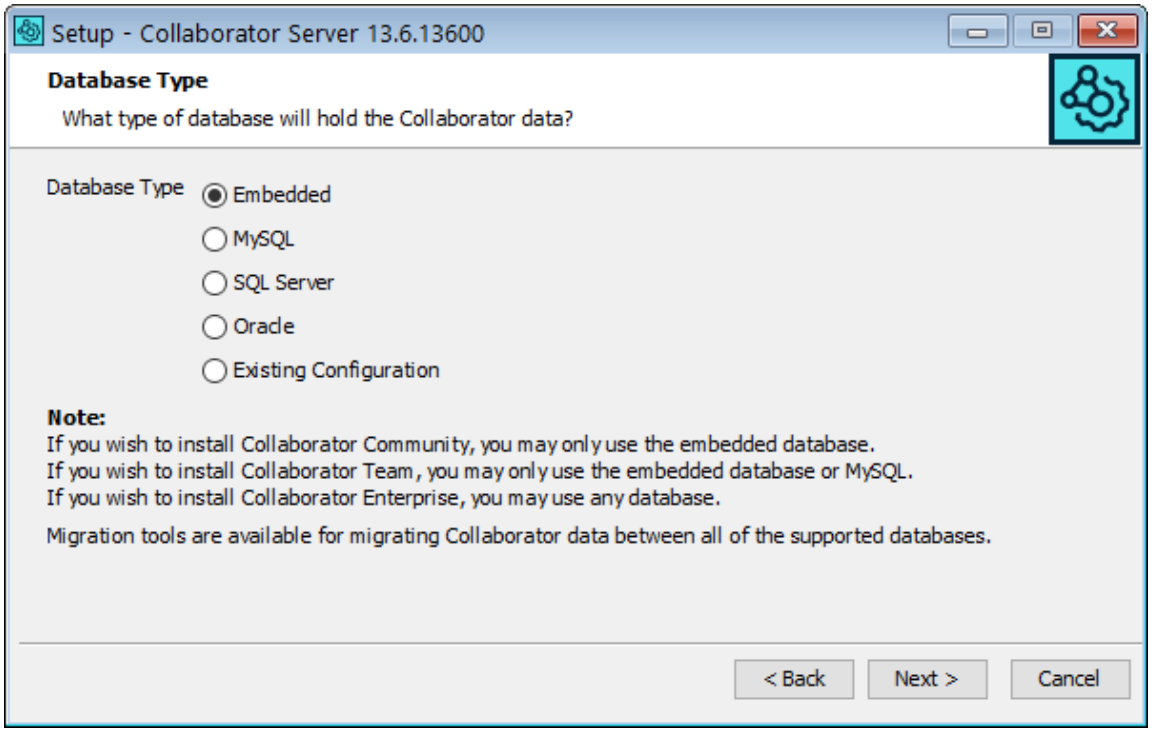

The port number should be selected as to not conflict with any existing services. The installer will attempt to connect to this port when you click Next; if the port is already taken by another process, you will get a warning message. You can elect to try a different port or continue. Continue at your own risk -- probably the server will not be able to start up and you will have to make the port available and restart the server manually.

Additionally, your firewall settings may prevent remote access on the default port or on your selected port. In order to allow access, you will have to modify your firewall settings accordingly. For more information see the troubleshooting section 565.

On the next screen, you can select whether to install helper RTC Service to your server. This service is required if you plan to use RTC integration 704.

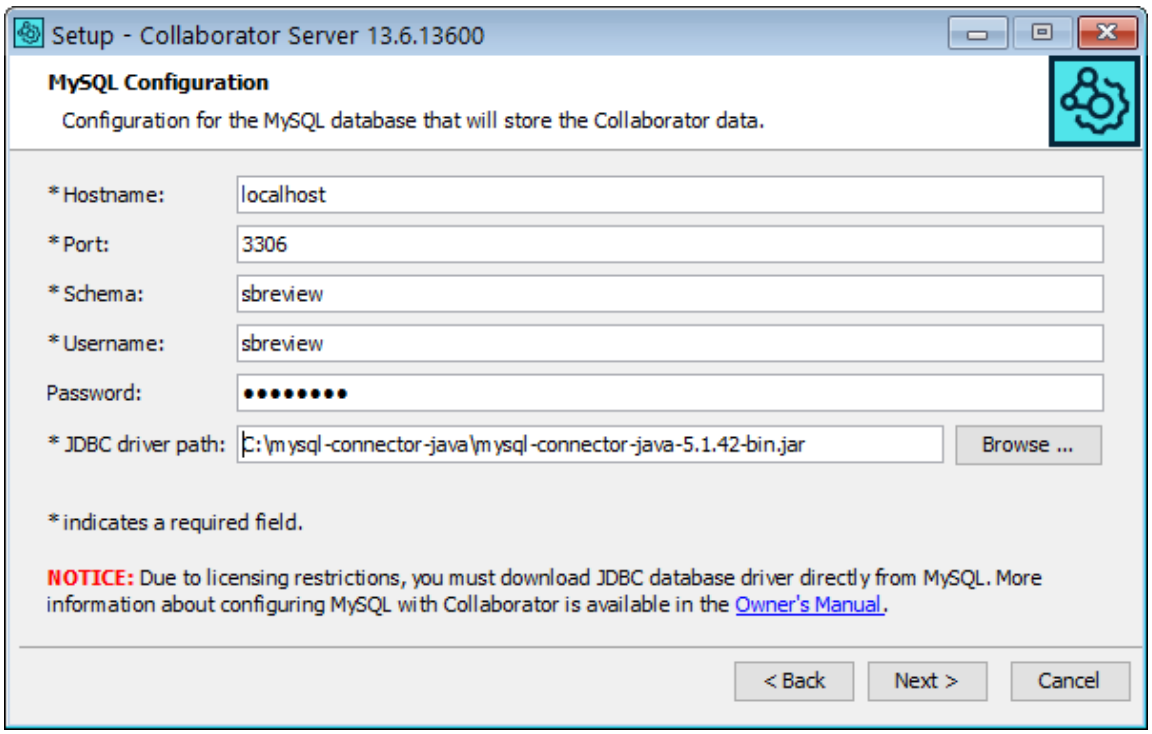



Next, you configure the database connection. The database should already be installed and ready 59.



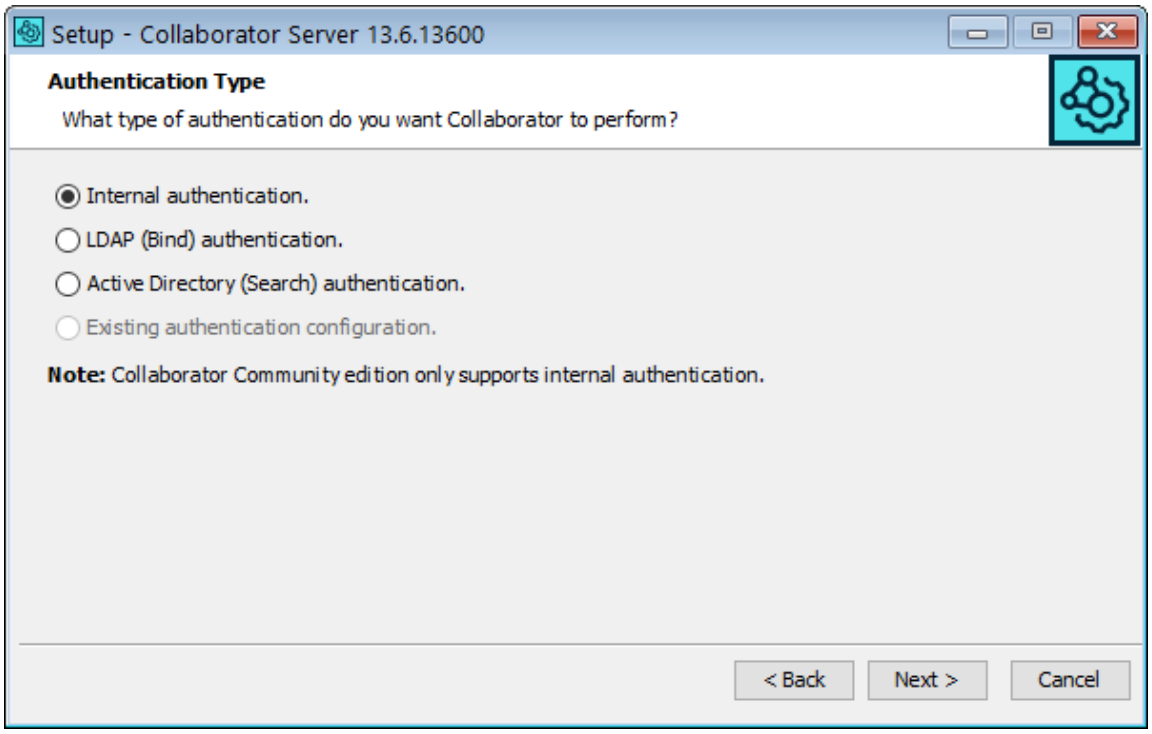
For trial installations, you will probably want to select the default "Embedded" database[60]. You should migrate[100] to a full-featured database before you use the server in production.

If you picked anything but the "Embedded" database, the next screen lets you configure the connection to the database server:




The exact format of this screen depends on the database you chose. The `Username` and `Password` fields here refer to the *database* username and password. Typically, you will want to create a special database username and password for the Collaborator application so you can control exactly which data it has access to. This usually means full access to the database created for Collaborator and no access to any other database.

The next screen lets you decide how users will be authenticated in Collaborator.

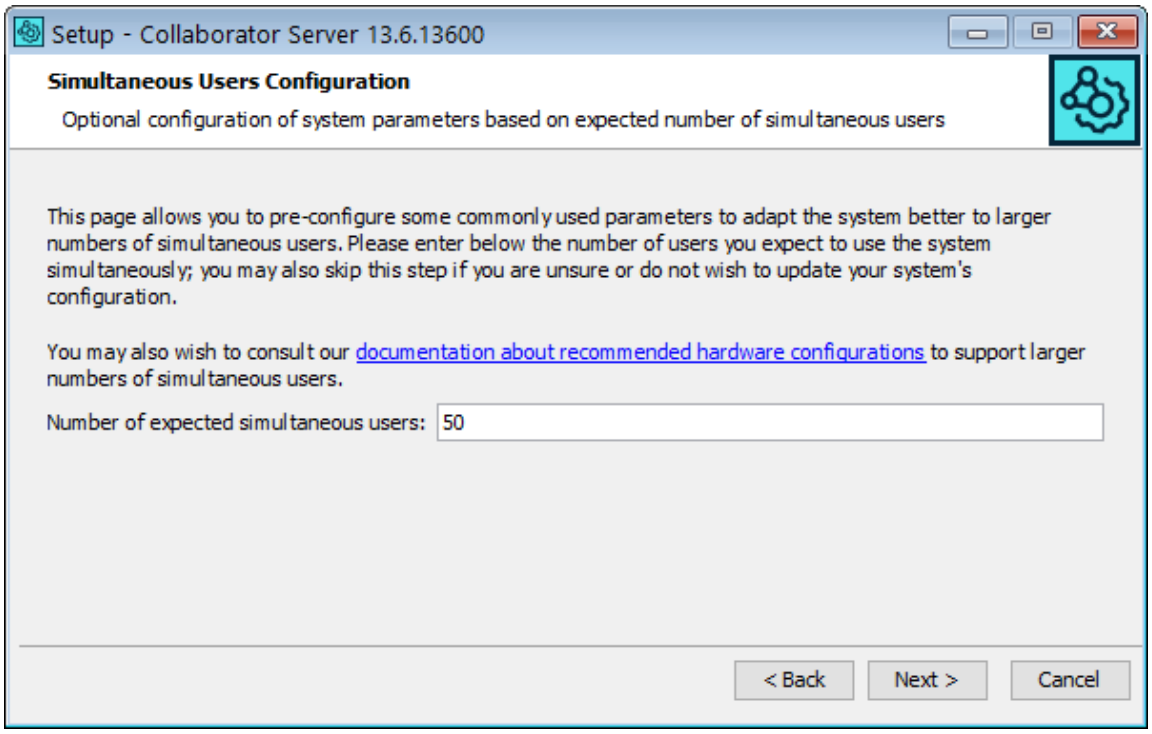



The simplest setting is "Internal," which means Collaborator should maintain usernames/ passwords in its own database.

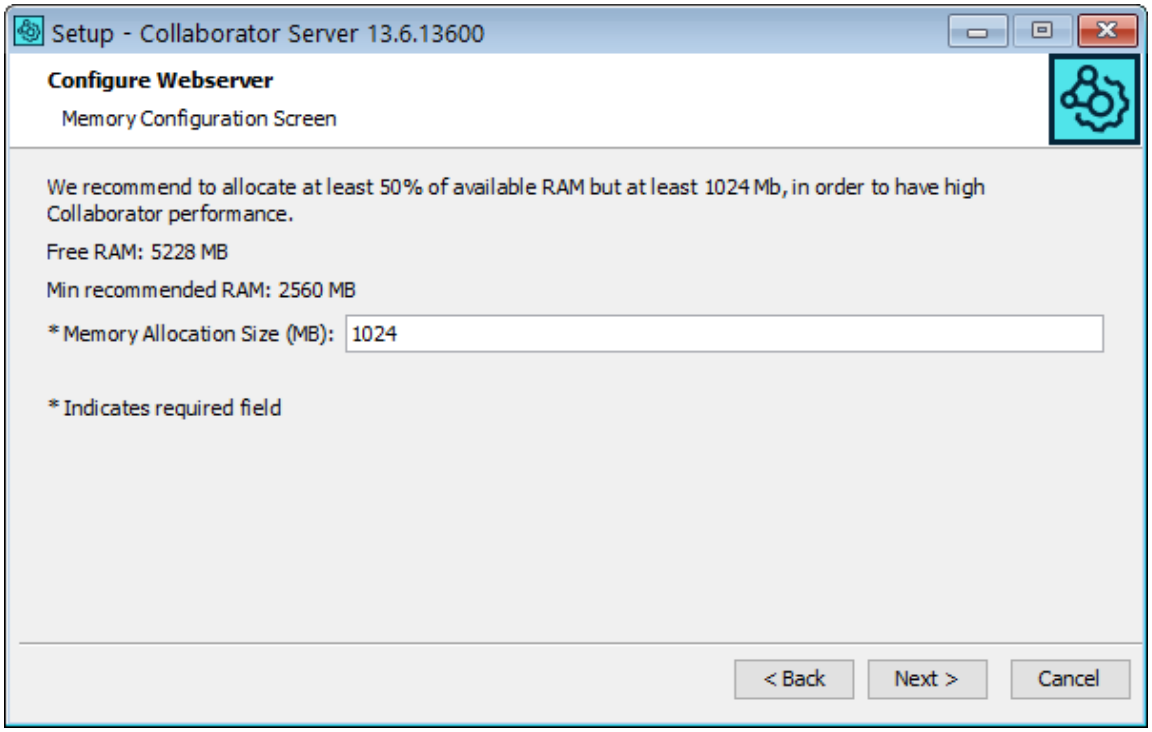
If your company uses LDAP or ActiveDirectory, you can use that method to authenticate users in Collaborator, which means you do not have to maintain the Collaborator user list at all. If you select the LDAP method you will get an additional wizard screen that lets you supply the settings for your LDAP server. See the LDAP section 119 for details.

You will also have the option to choose "Existing LDAP authentication configuration" if you are upgrading 83 an existing installation.

The next screen allows you to specify the approximate number simultaneous users and automatically configure your server to support this number of users.

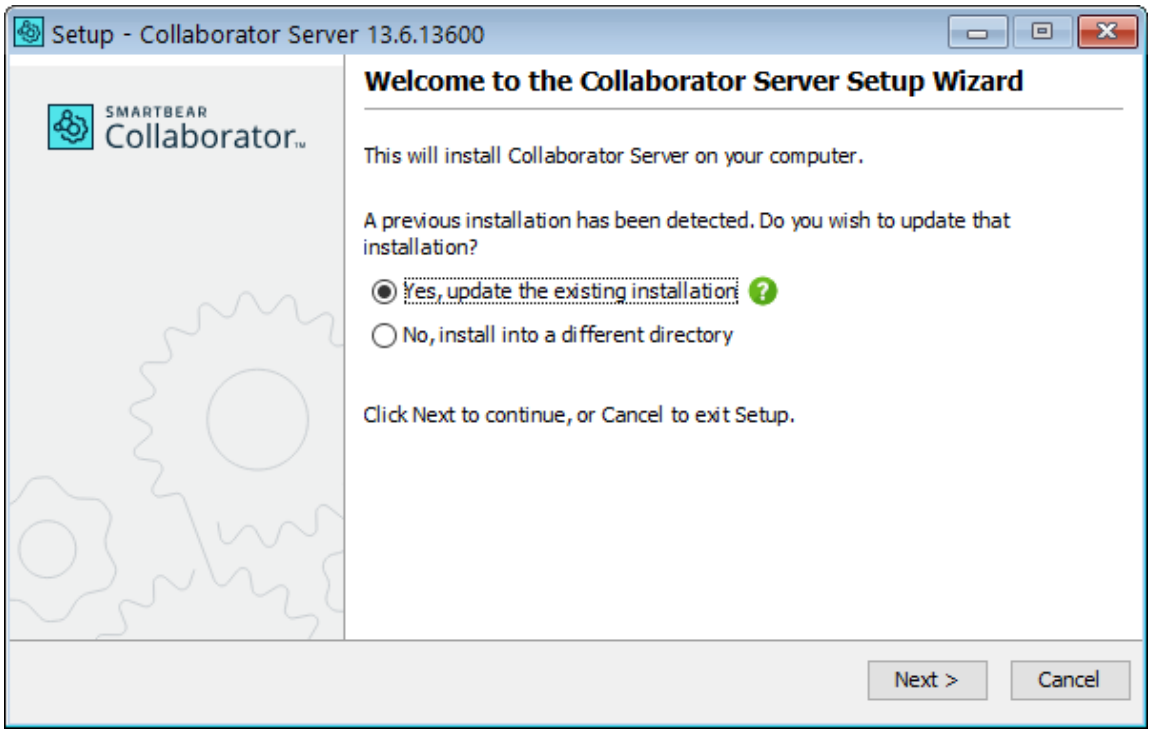



The next screen allows you to specify lower memory bound for Collaborator server memory allocation.



## Upgrade Installation

If the installer detects an existing Collaborator server, either to upgrade the existing server or to install into another directory.




When upgrading, the installer retains the current configuration of a server, that is, it assigns the same installation folder and all setting values as in the current configuration - though they can be edited, if desired.

## The Non-Graphical Installer (Windows only)

If you are upgrading the server instead of installing from scratch, you have the option of installing without any graphical user interface.

We recommend that you always use the graphical interface when possible because it gives you the chance to review settings, especially new settings that we might have added since your last installation.

To run the installer without a GUI, run the installer from a command-line using the `-q` switch. To set the installation directory from the command-line, use the `-dir [directory]` switch. The `-q` switch gives you a silent install and will not prompt you for any installation instructions. If you would like to be prompted for installation instructions without using the graphical installer, you can run the installer from the console using the `-c` command. Please note that in this case you should answer to prompt messages in console input, that follows the prompt message (that is, **not** in the prompt message itself).

```
C:\install>ccollab_server_13_6_13601_windows_x64 -c
C:\install>This will install Collaborator Server on your computer.
OK [o, Enter], Cancel [c]
C:\install>o
```

## Install Tips

### Installing more than one server instance on a single machine

All server instances need to be running on unique and available ports. Each server instance will also need a separate database and license code.

On Windows, install the different instances in different directories, on different ports, with separate database instances. This will get all of the software installed, but only the last one will be properly installed as a service. To install the others as a service, run the following command from the command line:

```
<install-dir>\ccollab-server /install service-name
```

This will install that instance as a Windows service with the specified service name. To uninstall a service that was misnamed or no longer used:

```
<install-dir>\ccollab-server /uninstall service-name
```

On *nix platforms, you should install the server in multiple directories and then edit your `/etc/init.d` scripts accordingly.

### Installing on a system with multiple JRE installations

On systems with multiple JREs installed, it may be necessary to specify to the installer which JRE should be used for Collaborator. On Windows platforms, running the installer with the -manual argument will suppress the JRE search and cause the installer to prompt for the JRE location (specifically, java.exe). On *nix platforms, you can specify the JRE location by setting the INSTALL4J_JAVA_HOME_OVERRIDE environment variable to the JAVA_HOME value.

## 3.1.4 Server Upgrades

This topic explains how to upgrade from a previous version of Collaborator server and retain all

your data and settings.

For major releases, we recommend that you install the new version on a test machine using a copy of your current database. Test a typical workflow on the system to make sure it works and that there are no surprises for your users. See Testing Newer Versions 86 for details.

## Back Up Before Upgrading

During the upgrade process, changes may be made to Collaborator's database schema. Thus the upgraded server's database may become incompatible with previous versions of Collaborator. Therefore, it is important to perform a complete backup 99 before attempting an upgrade.

Creating a backup is especially important if you are upgrading from an outdated version of Collaborator, CodeCollaborator or PeerReview Complete. Collaborator merges and replaces both CodeCollaborator and PeerReview Complete.

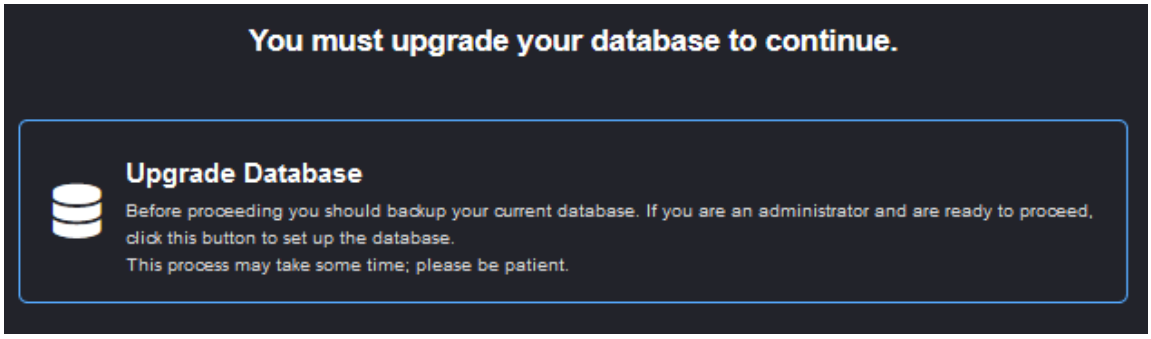
## How to Upgrade

Upgrading the server component works exactly like installing 70 it. To keep your data and configuration, **do not uninstall previous version** of Collaborator server. Just install a new version over the existing version.

Download the server installer(s) and proceed through the installation wizard. If the installer detects an existing Collaborator server, it automatically assigns the same setting values as in the current configuration - though they can be edited, if desired.

> During server upgrade, the installation wizard verifies Java Database Connectivity (JDBC) drivers your server uses to connect to MySQL, SQL Server or Oracle databases. If these are legacy JDBC drivers, the wizard removes them. In this case you may need to download and install new drivers as described in Database Installation 59 sections.

Once the installation is finished, an update of database schema could be required. In this case, on attempt to open a Collaborator Web Client, it will display the database update form:




Ensure that you have made a backup and press Upgrade Database button to proceed with the database migration. The rest of schema upgrade process is automatic. This may take some time depending on the scale of the changes required, so please be patient.

## Compatibility with Older Clients

Collaborator server is backward compatible with a certain number of earlier versions of Collaborator clients.

Currently, Collaborator 11.x and earlier clients are incompatible with Collaborator server 12.0 and later. To work with it, you will need to upgrade your clients to version 12.0 or later.

You can find the number of the oldest stable client application, as well as specify your own minimum allowable build number in the Administrator Settings 195. Client applications whose build number is lower than Minimum Client Build 195 are incompatible with the current version of Collaborator server.

In order to use new features of Collaborator, we recommend that you upgrade Collaborator clients, as well.

## Upgrade Notes

- **Windows-Only:** The system will be taken off-line while the upgrade takes place and will start automatically after the upgrade completes. When upgrading using a 64 bit installer on a server installed using the 32 bit installer, be sure to verify the install directory is the same as you currently have. The 64 bit installer will NOT automatically detect the existing install directory of a Collaborator install that was done using the 32 bit installer. In this scenario, simply edit the install location to be the fully qualified path of the existing install directory.

- **Unix-Only:** The installer will *not* automatically stop or start the server, so if the server is currently running you will need to manually stop it, perform the upgrade, and then restart it.

- After the upgrade validate the configuration of your Java Database Connectivity (JDBC) drivers. Upgrading an outdated server may remove legacy JDBC drivers.

#### 3.1.4.1 Testing Newer Versions

If you are currently using Collaborator and want to try a newer version without risking your existing installation, there are a couple ways to install a newer version in an existing Collaborator environment while still maintaining the integrity of your existing version.

## Fresh Installation

This option allows a quick and easy way to try Collaborator from scratch. However, you will not have access to all your existing review and user data.

To test a new Collaborator server with a blank database, download and install 70 the newer version of the Collaborator server software. If you are installing onto the same machine as your existing Collaborator server software, *be sure to specify a different directory, port number, and database instance.* Installing it in the same directory will upgrade your current installation, and you will not be able to "undo".

## Parallel Operation

This option allows you to continue running the old version of Collaborator in parallel with the newer version you are testing.

1. Do a complete backup 99 of your existing Collaborator server

2. After the backup finishes, install the exact same build of the Collaborator server software that you are currently using. For example, if you are currently running version 8.0.8001, then install version 8.5.8500 (older versions of the installer are available here).

   After the installer finishes it will display a web page in your default browser. That web page will have a button for creating the Collaborator database tables - do not push that button! You do not want those tables to be created because you are about to restore from your existing database.

   Again, if you are installing onto the same machine as your existing Collaborator server software, *be sure to specify a different directory, port number, and database instance.* Installing it in the same directory will upgrade your current installation, and you will not be able to "undo".

3. Restore the backup 99 that you just created to the new installation of Collaborator.

4. Then run the installer 70 for the newer version and when prompted, select the directory that contains the Collaborator server that you just installed. This will cause the installer to upgrade that installation.

## Windows Notes

If you choose either of the above options, you will end up with two instances of the Collaborator server running: one for your older version and one that is running the newer version. If you are running both instances on the same machine and if the operating system on that machine is Windows, then there is one additional step.

The Collaborator server installer for Windows always uses the same Windows service name: ccollab-server. This means that after you install a second instance of the Collaborator server on a Windows system, the Windows service named ccollab-server points to the new installation. The original installation no longer has a Windows service entry and is therefore no longer running.

This problem is easy to fix. In a command window set the working directory to the original installation of the Collaborator server software. Then enter this command:

```
ccollab-server /install <service-name>
```

You can specify anything you want for <service-name> except for ccollab-server. This will create a new Windows service with its own name for your existing Collaborator server installation.

### 3.1.5 Platform-Specific Notes

This is additional server-related information that is specific to certain operating systems.

## Windows

The web server is installed as a Windows Service. This means the server starts up automatically when the machine boots up, and no user needs to be logged in. The service can be started and stopped manually and even remotely.

The default installation is for the service to start automatically upon system startup using the default service user. All of these settings can be changed by the administrator after installation using Microsoft's standard service configuration control panel.

**Warning:** If you change the user under which the service runs, make sure the installation directory is still both readable and writable by the new user.

**Warning:** If Collaborator is located on the same machine as the database, and if the Collaborator service starts up before the database service, Collaborator might fail to start up. The work-around is to restart the Collaborator service, but the fix is to use service dependencies to tell Windows that the Collaborator service is dependant on the database service. This is a standard Windows service feature.

### Linux / Solaris / BSD

The web service is not automatically installed such that it will run automatically when the system starts up. However this is easy to set up.

The installation directory contains a file `ccollab-server` that can accept the usual start and stop commands. Create a symbolic link to this file from your standard installation directory (for example, `/etc/rc3.d` or `/etc/rc.local`) to cause the server to start automatically upon system startup.

**Warning:** If Collaborator is located on the same machine as the database, make sure it starts up *later* in the start-up process than the database server. Otherwise, Collaborator will fail to start up.

## 3.2 Server Administration

This chapter covers different aspects of server management and administration.

### In This Section

- Security Considerations 103
  Covers various options which affect the overall security of the system.
- LDAP and Active Directory Authentication 119
  Describes how to establish LDAP or Active Directory authentication.
- Configuring HTTPS 111
  Describes how to configure HTTPS connections.
- Singe Sign-On 130
  Describes how to establish single sign-on authentication.
- JMX Monitoring 155
  Describes how to use Java's standard, built-in network monitoring system.
- Technical Server Specifications 157
  Describes the technical aspects of the Collaborator server.
- High Availability Best Practices 159
  Describes how to establish a test server.
- Variable Substitution 161
  Describes the built-in system of variable substitution.
- Archiving Reviews 165
  Describes how to pack completed reviews to a ZIP archive.
- Branding Your Server 168
  Describes how to customize the web interface with your company logo.
- Logging 169
  Describes the logging system of the Collaborator server.
- Troubleshooting 173
  Describes administrator actions in case of troubles.

### 3.2.1 Network Configuration

Certain network topologies and configurations require specific configuration in the server component.

## Server Proxies

Collaborator optionally will connect to a licensing server hosted by SmartBear via standard HTTP protocol in order to validate your license code. If you are installing Collaborator in an environment where outbound HTTP requests are required to use a proxy, you will need to configure Collaborator to use the proxy.

Proxy settings are configured in the *installation-directory*/ccollab-server.vmoptions file. To enable an HTTP proxy, you will need to add the following lines to that file:

```
-Dhttp.proxyHost=proxy_hostname
-Dhttp.proxyPort=proxy_port
-Dhttp.proxySet=true
```

Many proxies are configured to reject connections to internal URL's via the proxy, as these connections are supposed to be made directly. For this reason, it is also a good idea to configure internal hosts as non-proxied hosts. Specifically, it is important to configure `localhost` (including its resolvable name), the bug tracking system, and the version control server as non-proxied hosts. These URL's are configured in Collaborator for integration purposes and a connection is made to validate the URL's entered in the configuration screens. If the proxy rejects the connections, you may not be able to properly edit those fields. To configure non-proxied hosts, add the following line to the `collab-server.vmoptions` file:

```
-Dhttp.nonProxyHosts="localhost|collabserver|*.mydomain.com"
```

The format for the value is a list of hostnames delimited by "|" and using "*" for a wildcard.

Some newer firewalls have been known to cause additional problems with HTTP proxying. For instance, some firewalls periodically redirect HTTP requests to a firewall-generated web page requiring the user to log in. These firewalls are specifically designed to limit web access to users; preventing services (such as Collaborator) from accessing web services. Organizations with such firewalls will need to work with their network administrators to exempt Collaborator from this policy or otherwise allow access.

### 3.2.2 Licensing

SmartBear offers both fixed-seat and floating-seat licenses for Collaborator. Additionally it offers separate fixed-seat licenses for Simulink integration.

## Licensing Types

**Fixed-seat** licenses specify the maximum number of users, who can work with Collaborator. A user is a human being (not a machine) who was active in the past 30 days. If a user did not work with Collaborator during this period, Collaborator does not count this user as a license consumer. The fixed-seat licenses are most appropriate when most of users in your organization work with the product on a daily basis.

**Floating-seat** licenses specify the maximum allowed number of active ***concurrent*** users. A user is considered active if they are currently consuming a license. Floating-seat licenses are most appropriate when you have many users that will work with Collaborator occasionally.

In both cases, a user is a human being, not a machine or client software. If someone uses the Eclipse or Visual Studio client, stand-alone client and command-line client, all the usages are counted as one user (one seat).

**Notes:**

- Only the Collaborator server is licensed. The Collaborator client software does not need licensing.
- Collaborator does not support both fixed-seat and floating-seat licenses on the same server.
- For information on when Collaborator increases the license counter, see below 91.
  When you disable 223 a user, Collaborator returns the license to the pool immediately. It logs off that user and does not allow them to log in.
- Each license key is linked to the node ID of your Collaborator server, and gets carried over during server migration. Therefore, a new license key is not needed when migrating the server.
- User authentication, sometimes referred to as a log in, is not necessarily tied to license consumption. As an example, the user can authenticate through the GUI client but not be consuming a license.

## License Consumption

The below describes how **floating-seat** license are consumed.

**Collaborator WebUI**

- When a floating-seat user logs in to Collaborator via a web browser, a license is assigned to that individual.
- The license will remain assigned to the user for as long as there is a Collaborator tab open in the browser.
- The license will be returned to the pool once all Collaborator tabs are closed, and one hour has passed.
- The license will be returned to the pool immediately if the user clicks “Logout”.

**Collaborator Command-Line Client and GUI Client**

- The command-line client only consumes a license when using the `ccollab admin wget` command.

- The GUI client never consumes a license.
- The tray notifier never consumes a license.

**Collaborator Eclipse Plug-in and Visual Studio Add-in**

- A license is consumed when the Collaborator review summary or diff viewer is open.
- Once a user leaves the Collaborator review summary or diff viewer windows, the license will be returned to the pool after one hour – provided they do not return to one of those screens.
- When Eclipse or Visual Studio is closed, the license is returned to the pool after one hour.

## Exceeding the License Limit

How does Collaborator behave when you reached the license limit? If a user attempts to connect to Collaborator, the server refuses the login and displays an error message explaining that there are no available licenses left in the pool and asks the user to contact their Collaborator administrator. Additionally, Collaborator sends a notification message to the administrator stating that a user login was denied because the license limit has been reached.

**Note:** A user login can include situations when a user returns back to Collaborator after some period of inactivity. In this case, Collaborator may not ask the user to enter their login credentials. However, the user will see an error message about reaching the license limit.

One way to decrease the license consumption is to disable 223 unneeded user accounts. Another way is to ask users who are not working with Collaborator currently to log off. The administrator can also log out these users manually.

The system administrator 224 is always allowed to log in, even if the license limit has been reached. This allows you to fix licensing issues either by disabling users or by activating a new license key.

## Monitoring License Usage

There are several ways to keep track of licenses used:

- You can view the current license usage number on the Users Administration 219 screen. You can also manually disable or log out users, if you need to reclaim the licenses quickly.
- You can also see the license usage number, the number of denied user logins, and optionally, the list of active floating-seat users on the Licensing 90 screen.
- You can see historical activity patterns on the System Status 212 screen.
- You can use JMX 155 to monitor usage and licensing information.

## How Many Licenses Do I Need?

It is hard to give guidance because it varies quite a bit.

Generally the best advice is just try using Collaborator. There is a report on the User Administration 219 page that tells you exactly how many "fixed" and "floating" seats you are using at the moment. Use this number to estimate the number of seats you will need and determine the license type.

Typically, a trial does not involve everyone, and typically the usage pattern is not exactly the same as when it will be deployed, so you will have to estimate. Still, you will be doing so with some real numbers.

Here are some additional pointers:

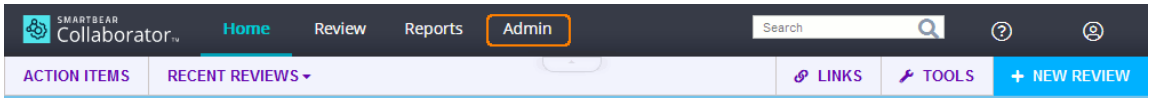
- If you have developers in many time zones, floating-seat licenses are usually considered to be a better option.
- If each user works with Collaborator daily, a fixed-seat license is typically the best option.
- If you expect large spikes in usage — when perhaps everyone is online at once during a review crunch — you most likely will want fixed-seat licenses. Otherwise, you have to get enough floating seats to handle that peak usage.

## Activating Your License

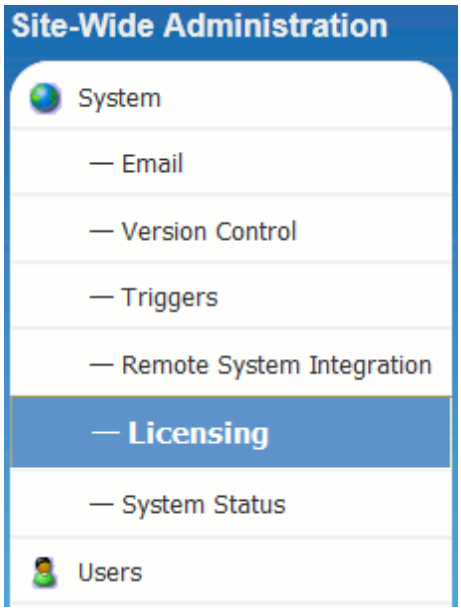
### 1. Getting Node ID

To get the license code, send the Node ID attribute of your Collaborator server to your sales account manager at SmartBear:

1. Log in to Collaborator as a user with administrator permissions.
2. Select **Admin** from the top menu:



3. Select **Licensing** from the list on the left:




4. Switch to the **Collaborator License** tab or **Simulink Integration** tab depending on what licence you want to activate.

5. In the **Current License** section on the right, find the Node ID value. Copy it and send to your sales account manager at SmartBear:


## 2. Entering Your License Code

After you receive the license code from SmartBear, you need to activate the license on the Collaborator server (client Collaborator software does not require licensing):

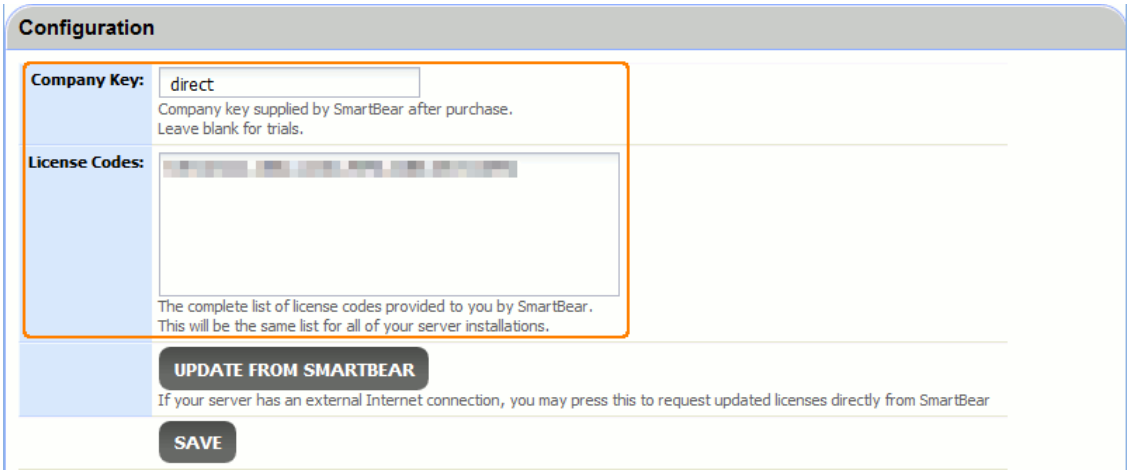
1. Log in to Collaborator as a user with administrator permissions.

2. Select **Admin** from the top menu.

3. Select **Licensing** from the menu on the left.

4. Switch to the **Collaborator License** tab or **Simulink Integration** tab depending on what licence you want to activate.

5. In the **Configuration** section, enter your company key in to the **Company Key** field and license code into the **License Codes** field:




6. Click **Save** to apply the changes.

If license was activated successfully, Collaborator will display your licensing information (the number of fixed and floating seats). In case of an error, Collaborator will displays the error message at the top of the page.

## 3. Assign Simulink integration licenses to Collaborator users

In order to review Simulink models[404], you should have Collaborator Enterprise edition license, obtain Simulink integration licenses and assign them to particular users.

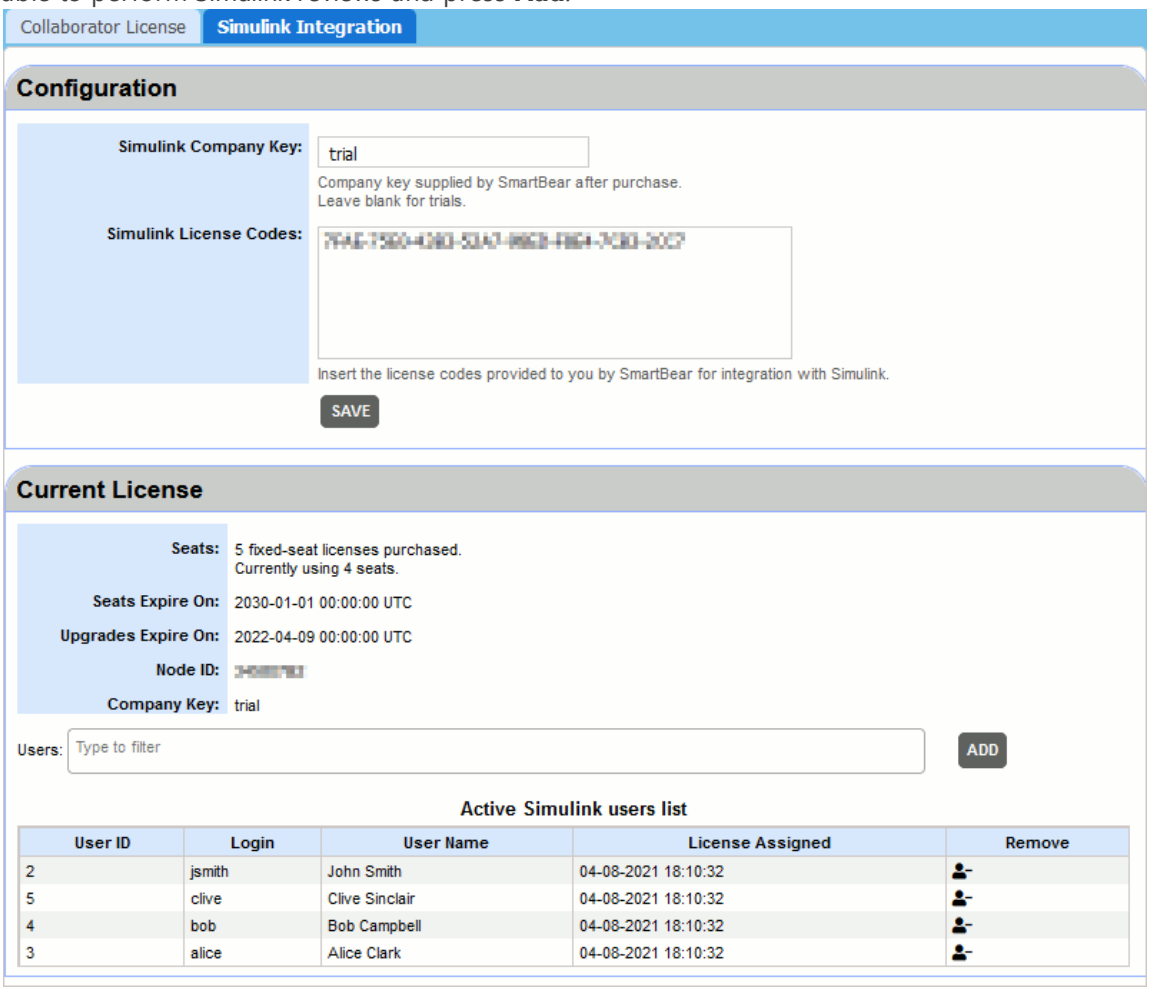
1. Log in to Collaborator as a user with administrator permissions.

2. Select **Admin** from the top menu.

3. Select **Licensing** from the menu on the left.

4. Switch to the **Simulink Integration** tab.

5. In **Current License** section, in the **Users** field, specify the login names of users who should be able to perform Simulink reviews and press **Add**.



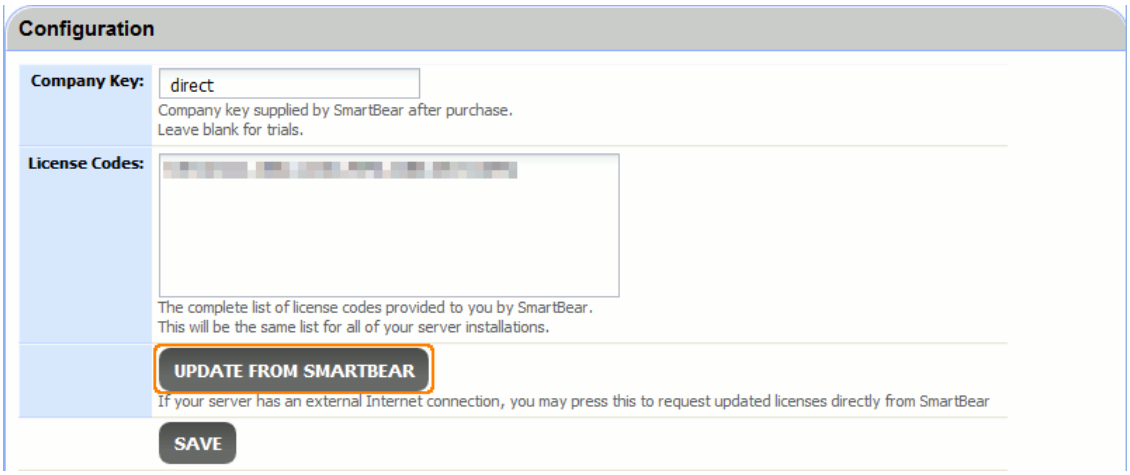

The **Active Simulink users list** displays users who currently have Simulink integration licenses assigned. To revoke a license from a user, click icon in the user list.

Simulink integration licenses can be re-assigned to another user only once in 24 hours.

## Updating License Code

You can receive updated licenses from the SmartBear server after your initial license setup. To do this, simply click **UPDATE FROM SMARTBEAR** in the Configuration box (see the image below).

In order for this feature to work, your computer must have a functioning Internet connection, and proxies and firewalls in your network should allow connection to the Collaborator license server (URL `http://licensing.codecollaborator.com`, port 80).




### 3.2.3 Content Storage

Most review data is stored in the database. The one exception to this is the contents of the files under review. These are stored in a folder described as the **content storage** or **content cache**.

## Content Storage Location

The default location for the content storage is: `<Collaborator Server>/tomcat/collaborator-content-cache`

The internal structure of the content storage is subject to change between versions of Collaborator, so we do not recommend altering the contents of the content storage directly without specific instructions from SmartBear technical support.

## Change Content Storage Location

In some environments, the default content storage location could not be an acceptable solution. In this case you can change it.

Below are some things to consider when choosing a content storage location:

- File permissions may not allow writing in the installation folder.
- Maintenance of network storage may be easier if backups are already in place and disk usage is monitored automatically.
- Network storage allows for warm standby Collaborator servers to be available in the event of failure of the primary server.

To change content storage location:

1. Stop Collaborator server if it is running.

2. Open the `<Collaborator Server>/tomcat/conf/Catalina/localhost/ROOT.xml` file.

3. Find the `Parameter` element with the `content-cache` name attribute.

4. Specify the new path to the content storage in the `value` attribute of this element. Relative paths in this attribute are interpreted relative to the server installation folder.

5. Copy the contents of the <Collaborator Server>/tomcat/collaborator-content-cache folder to the new content storage folder. Otherwise, review participants would encounter this error message when trying to open a file in a review: "Content for is not available – it was probably archived by your administrator."

6. Start Collaborator server.

## Archive Content Storage

Over time, content storage will grow to be quite large, and will periodically need to be purged. When deleting a review 346, Collaborator deletes files that were uploaded for this review from the content storage as well (unless they are used in some other reviews). To check the current status of the content cache and to archive (or delete) files that have not been in use for a long time, see the Archive section 299 of the administrator interface.

## Share Content Storage

Sharing the content cache between servers is *not supported*. The only environment where Collaborator servers should be configured to use the same content cache is if one is configured as a warm backup for the primary server. A warm backup is a system that configured, *but not running*. It can be started in the event of primary server failure to reduce down time.

## Upgrade Content Storage Format

Starting from version 6, Collaborator has a new file content storage format that works better with many file systems by having a deeper folder structure, reducing the number of files in each folder. By default all new installations of Collaborator server use the new file content storage format.

To upgrade the content storage format of existing installations of legacy (CodeCollaborator 5 and earlier) servers:

1. Stop the server if it is running.

2. Open the `<Collaborator Server>/tomcat/collaborator-content-cache/cache.properties` file.

3. Edit it as follows:

```
version=2
lazy-upgrade-from-version=1
```

4. Start the server.

The cache.properties file is a Java properties file having two configuration keys, `version` and `lazy-upgrade-from-version`. The `version` key determines whether to use the new format (2), or the old format (1) for storing new data. The `lazy-upgrade-from-version` key determines whether to search for and upgrade data stored in the old format (1), or not (blank).

### 3.2.4 Backup / Migration

Collaborator can be backed up while it is running. Also, there is a system for migrating data from one database to another.

Collaborator stores almost all data in the database initially set up for it. It also uses file content storage 97 - a local directory that keeps copies of uploaded files. These are the two systems that need to participate in the backup/restore process.

## Back Up Database

We recommend that you automate database backup process and perform it regularly.

The mechanism for backing up the database depends on the database. See the documentation for your database and use backup software designed to integrate with that.

Collaborator servers that use embedded database automatically create a daily database backup at 3 AM server time. This daily backups are stored in the file content storage (*`installation-directory`*`/tomcat/collaborator-content-cache/db_backups`). That is, on embedded databases you can just backup the content storage as described below.

There is an alternate method for backing up the database which is to perform the first half of a database migration 100 and save the migration data file.

## Back Up File Content Storage

The default location of file content storage 97 - a local directory that keeps copies of uploaded files - is listed below (however, the default location can be changed by the system administrator):

```
installation-directory/tomcat/collaborator-content-cache
```

Typically, a backup mechanism will either copy this directory elsewhere or will keep a zip or other compressed archive file updated with the contents. You can do this while Collaborator is running, although most backup mechanisms will run at off-peak hours.

Files in this cache are stored in such a way that a file is written *only once* and thereafter is never changed. This means incremental backups of the directory are particularly easy -- only new files must be copied. Most file-copy utilities have a mode that means "only copy new files".

## Restore a Back Up

To restore a Collaborator installation, first restore the database as directed in your database documentation. Then, install the Collaborator server software. Finally, restore the contents of the collaborator-content-cache directory in the new installation directory. You can do this last step while Collaborator is running.

If you have used the database migration technique to back up the database, refer to steps 3-11 of Restoring the Data Dump File 101 for instructions on migrating the database data.

## Migrating Data Between Databases

Collaborator has a generic mechanism for migrating data between databases -- even if the databases are completely different types.

Applications of this migration technique include:

- migrating between the embedded database and one of the other databases when a trial moves to a production environment
- backing up a database in a database-independent and easily-inspected manner
- when we add support for a new database and you want to switch over to it
- we need to debug your database and you need to send a "dump" of your data to SmartBear Technical Support 173

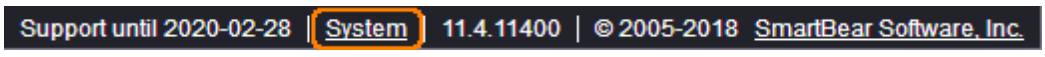
**NOTE:** Each license key is linked to the node ID, and it gets carried over during server migration. Therefore, a new license key is not needed when migrating the server.

There are two halves to the migration process: Creating the migration data dump file, and "restoring" the data dump file into a Collaborator installation.

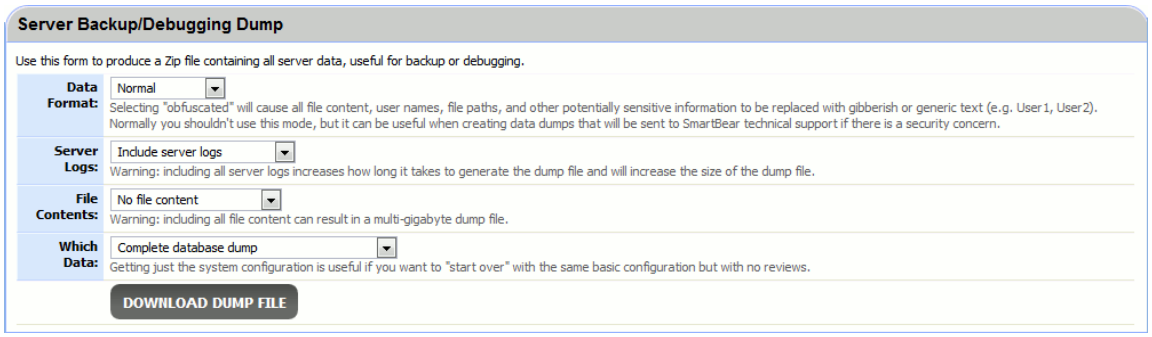
### 1. Creating the Migration Dump File

**Note:** If your Collaborator server runs on a 32-bit Java virtual machine, set any non-blank value to the `com.smartbear.ccollab.datamodel.manager.chunk.for.32.jdk` setting in the `ccollab-server.vmoptions` 1232 file and restart the server, before creating a dump file.

To create the migration data dump file, log into the Collaborator web server, and click the System link at the bottom of the screen:




Near the top of the screen is a form and a button which will allow you to download a whole system dump:




It contains the following settings:

**Data Format:** *Normal* - Leaves the data as is.

*Obfuscated* - Changes the data to conceal sensitive information.

**Server Logs:** Specifies whether to include server logs in dump file.

Note: If you are creating a dump file to send to technical support, it is important to include the server logs.

**File Contents:** *No file content* - Do not include file content.

*Include all file content* - Include all file content. This may generate huge dump file and cause a longer download.

**Which Data:** *Complete database dump* - This must be selected for a whole system dump.

*System configuration only - no review data* - Selecting this will only save the system settings.

Once you have filled out the form, click the "**Download Dump File**" button to download the database dump file. This is a ZIP file containing all your database data in a platform- and database-independent XML format, plus additional files that describes your server environment.

## 2. Restoring the data dump file

Loading this data into another Collaborator installation requires some effort. This process is intentionally complex to prevent accidental destruction of real data.

Below are the steps to restoring a database migration data dump file:

1. Verify your Collaborator Server install.

Make sure the Collaborator server version and build number are identical between the server that created the data dump file and the server that is loading it. If the versions do not match exactly, the restore might not succeed. SmartBear maintains installers for previous versions of the server if you need one. You can always upgrade to another version after the migration.

Be sure when running the installer, you give the correct connection parameters to point the server to the database to which you plan to migrate. Note: Even if the Collaborator Server install location is not changing, you will still need to rerun the installer to point Collaborator to the correct database.

Once the install is complete, a browser window should open. Do NOT initialize the database as prompted in the browser.

2. Move the dump file into a known location. In this example we will assume the location is:

```
c:\temp\ccollab-dump.zip
```

3. On a 32-bit Java virtual machine, set maximum Java heap size to 1.5Gb or above (`-Xmx1500M`) in the `ccollab-server.vmoptions` 1241 file.

4. Make sure the Collaborator server where you will load the dump is not running. This will not work on a running server.

5. Open the Tomcat session configuration file located here:

```
installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml
```

6. Find the parameter called database-migration-data-path, or create one if it does not already exist. It should look something like this:

```
<Parameter name="database-migration-data-path" value="c:
\temp\ccollab-dump.zip" override="false" />
```

7. Make sure the value string matches the location of the dump file, as in the example above. Use an absolute file path.

8. Save the configuration file.

9. Use a database administration tool to make sure the database configured for use with Collaborator has no tables in it. If there are any tables in the database the restore will not work. This prevents accidental restoring over an existing database.

10. Start the Collaborator server.

11. The server will automatically load the data from the database migration data dump file. If there were any problems with migration, you will see a helpful error message in the server log 177. The server will also log progress reports as data is loaded up, so if you have a large database and you wish to monitor migration progress you can "tail" the log file to see what's happening. Loading migration data can take a long time, so be patient!

12. Upon starting after restoring a migrated database, email notifications will be disabled. This is to prevent users from receiving spurious duplicate email notifications when administrators restore into test configurations. You may re-enable email notifications from the email administration page 199.

### 3.2.5 Security Considerations

Collaborator administrators need to be aware of several security issues and options which affect the overall security of the system. This section covers those issues.



#### Built-in Administrator Account

Each Collaborator server has a built-in administrator account "admin". By default its password is "admin". To improve security, you will need to specify your own password for the built-in administrator account in the Users 219 category of Collaborator settings, or using the following command-line:

```
ccollab login http://your_collabserver admin admin
ccollab admin user edit admin --password newpassword
```

#### File System Security

Collaborator relies on the underlying operating system as a foundation for overall system security. Several potentially sensitive items are stored in the local file system, including database credentials, LDAP credentials (if used), and file contents. Care should be taken to maintain system security of the server's operating system, so this information is not compromised. SmartBear does not have any specific security expertise, so we recommend you follow the guidance of your operating system's vendor.

## HTTP Transport Security

By default, the Collaborator server operates over regular HTTP. This means that all communications between clients and servers are unencrypted on the wire. Therefore, it is possible for someone with access to the network to use network sniffing tools to gather information from that traffic. Some things that are available over the wire are file contents, user conversations, and even authentication credentials (usernames and passwords). If wire-level security is a concern, administrators should configure the server to use secure http (HTTPS)[111]. Enabling HTTPS, also adds the "Secure" attribute to the browsers session cookies, that is, they can only be transmitted over an HTTPS connection.

## User Session Information Storage / Cookies

There are two cookies that store user login and session information (`CodeCollaboratorLogin` and `CodeCollaboratorTicketId` respectively). The expiration date of the session cookies is so far in the future that they will never expire, but there is a server setting that overrides that expiration date and controls the length of time that the session is valid. This setting is called "Login Ticket Time-To-Live[196]" and can be configured by the Collaborator admin. This setting is not an idle timeout; it is an absolute time the ticket will remain valid after it is created.

The application server-managed session cookie, JSESSIONID, is only valid during the session and it is used to identify the session, but it does not contain any user information. We also store some WebUI preferences locally, but that data also does not contain any user information.

All session cookies have the "HTTPOnly" attribute set, which means that these cookies can only be used by web browsers, and cannot be accessible via scripts or by other means.

## Obfuscating Database Passwords

Some environments dictate that sensitive passwords stored in configuration files be obfuscated. In the case of Collaborator, this most commonly occurs in conjunction with the database connection information stored in `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml`.

Starting in Collaborator 8.4.8403, obfuscating the database password has *preliminary* support as a post-install operation. Three forms of obfuscation are supported: base64 encoding, base64-encoded AES 128 bit and base64-encoded AES 256 bit. AES obfuscation uses ECB mode with a fixed key and PKCS#5 padding.

Of the three forms, base64-encoding is the recommended process, if sufficient, as it is simpler.

**Note:** To use AES-256 bit obfuscation additional files are required.

Due to the import restrictions of some countries, Java SE have built-in restrictions on available cryptographic strength. Cryptographic strength can be configured via jurisdiction policy files that can be downloaded separately. In order to use AES-256 bit obfuscation, you will need to download and install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle web site: Java Cryptography Extension for JRE 7 and Java Cryptography Extension For JRE 8.

**Base64 obfuscation:**

1. Stop your Collaborator server instance. See Platform-Specific Notes 87 for instructions for Windows, Linux, Solaris and BSD platforms.

2. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

3. Locate the `<Resource ... />` XML tag. This tag declares the database configuration of your Collaborator server.

4. Append the "factory" attribute to the Resource tag:

```
<Resource ... factory="com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory"/>
```

5. Using the tool of your choice, generate a base64-encoded version of your database password (as found in the "password" attribute of the Resource tag). Many base64-capable tools and online encoder/decoder interfaces exist, a web search for "base64 encode" should prove illuminative.

6. Prepend the encoded string with "$1$". This indicates to Collaborator that plain base64 encoding is being used. For example, the existing password "testpass" would become "$1$dGVzdHBhc3M=".

7. Replace the value of the existing "password" attribute with the string composed in the prior step. The Resource tag will look similar to:

```
<Resource ... password="$1$dGVzdHBhc3M=" factory="com.smartbear.
ccollab.CollaboratorBasicDataSourceFactory"/>
```

8. Save the `ROOT.xml` file.

9. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

**AES-128 obfuscation:**

*Note: base64 is recommended due to simplicity unless your environment absolutely requires this level of obfuscation.*

1. Stop your Collaborator server instance. See Platform-Specific Notes[87] for instructions for Windows, Linux, Solaris and BSD platforms.

2. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

3. Locate the `<Resource ... />` XML tag. This tag declares the database configuration of your Collaborator server.

4. Append the "factory" attribute to the Resource tag:

```
<Resource ... factory="com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory"/>
```

5. Using the command line interface, in the `<server install path>/tomcat/` directory of your Collaborator instance, execute the following command line:

```
java -cp webapps/ROOT/WEB-INF/classes;webapps/ROOT/WEB-INF/lib/*;lib/
tomcat-dbcp.jar;bin/tomcat-juli.jar com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory
```

   (On windows platforms, replace ":" in the command above with ";".)

6. Enter your password when prompted and specify the encoding format as "aes128". The program will then output the encoded string, which has a prefix of "$2$" to indicate that AES-128 obfuscation is being used. For example, the pre-obfuscation password "testpass" would become "$2$Nobujw9X9ZJsSOYapNZh+w==".

7. Replace the value of the existing "password" attribute with the string composed in the prior step. The Resource tag will look similar to:

```
<Resource ... password="$2$Nobujw9X9ZJsSOYapNZh+w==" factory="com.
smartbear.ccollab.CollaboratorBasicDataSourceFactory"/>
```

8. Save the `ROOT.xml` file.

9. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

**AES-256 obfuscation:**

*Note: base64 is recommended due to simplicity unless your environment absolutely requires this level of obfuscation.*

1. To enable AES-256 encryption you need to install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. They can be downloaded from the Oracle's website: Java Cryptography Extension for JRE 7 and Java Cryptography Extension For JRE 8.
   Download and install the JCE package as described in the README.TXT file shipped within the downloaded archive.

2. Stop your Collaborator server instance. See Platform-Specific Notes 87 for instructions for Windows, Linux, Solaris and BSD platforms.

3. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

4. Locate the `<Resource ... />` XML tag. This tag declares the database configuration of your Collaborator server.

5. Append the "factory" attribute to the Resource tag:

```
<Resource ... factory="com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory"/>
```

6. Using the command line interface, in the `<server install path>/tomcat/` directory of your Collaborator instance, execute the following command line:

```
java -cp webapps/ROOT/WEB-INF/classes;webapps/ROOT/WEB-INF/lib/*;lib/
tomcat-dbcp.jar;bin/tomcat-juli.jar com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory
```

   (On windows platforms, replace ":" in the command above with ";".)

7. Enter your password when prompted and specify the encoding format as "aes256". The program will then output the encoded string, which has a prefix of "$3$" to indicate that AES-256 obfuscation is being used. For example, the pre-obfuscation password "testpass" would become "$3$w8MhPu6t7vobGeNvTx8RoA==".

8. Replace the value of the existing "password" attribute with the string composed in the prior step. The Resource tag will look similar to:

```
<Resource ... password="$3$w8MhPu6t7vobGeNvTx8RoA==" factory="com.
smartbear.ccollab.CollaboratorBasicDataSourceFactory"/>
```

9. Save the `ROOT.xml` file.

10. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

## Obfuscating LDAP Passwords

Starting in Collaborator 8.5.8501, LDAP passwords may be obfuscated in a similar fashion to the database password above. Three forms of obfuscation are supported: base64 encoding, base64-encoded AES 128 bit and base64-encoded AES 256 bit. AES obfuscation uses ECB mode with a fixed key and PKCS#5 padding.

Of the three forms, base64-encoding is the recommended process, if sufficient, as it is simpler.

**Note:** To use AES-256 bit obfuscation additional files are required.

Due to the import restrictions of some countries, Java SE have built-in restrictions on available cryptographic strength. Cryptographic strength can be configured via jurisdiction policy files that can be downloaded separately. In order to use AES-256 bit obfuscation, you will need to download and install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle web site: Java Cryptography Extension for JRE 7 and Java Cryptography Extension For JRE 8.

**Base64 obfuscation:**

1. Stop your Collaborator server instance. See Platform-Specific Notes 87 for instructions for Windows, Linux, Solaris and BSD platforms.

2. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

3. Locate the `<Realm ... />` XML tag. This tag declares the LDAP (JNDI) authentication resource that Collaborator uses.

4. Change the "className" attribute of the Realm tag to `"com.smartbear.ccollab.auth.CollaboratorJNDIRealm"`:

```
<Realm ... className="com.smartbear.ccollab.auth.
CollaboratorJNDIRealm"/>
```

5. Using the tool of your choice, generate a base64-encoded version of your database password (as found in the "connectionPassword" attribute of the Realm tag). Many base64-capable tools and online encoder/decoder interfaces exist, a web search for "base64 encode" should prove illuminative.

6. Prepend the encoded string with "$1$". This indicates to Collaborator that plain base64 encoding is being used. For example, the existing password "testpass" would become "$1$dGVzdHBhc3M=".

7. Replace the value of the existing "connectionPassword" attribute with the string composed in the prior step. The Realm tag will look similar to:

```
<Realm ... className="com.smartbear.ccollab.auth.
CollaboratorJNDIRealm" connectionPassword="$1$dGVzdHBhc3M="/>
```

8. Save the `ROOT.xml` file.

9. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

**AES-128 obfuscation:**

*Note: base64 is recommended due to simplicity unless your environment absolutely requires this level of obfuscation.*

1. Stop your Collaborator server instance. See Platform-Specific Notes 87 for instructions for Windows, Linux, Solaris and BSD platforms.

2. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

3. Locate the `<Realm ... />` XML tag. This tag declares the LDAP (JNDI) authentication resource that Collaborator uses.

4. Change the "className" attribute of the Realm tag to `"com.smartbear.ccollab.auth.CollaboratorJNDIRealm"`:

```
<Realm ... className="com.smartbear.ccollab.auth.
CollaboratorJNDIRealm"/>
```

5. Using the command line interface, in the `<server install path>/tomcat/` directory of your Collaborator instance, execute the following command line:

```
java -cp webapps/ROOT/WEB-INF/classes;webapps/ROOT/WEB-INF/lib/*;lib/
tomcat-dbcp.jar;bin/tomcat-juli.jar com.smartbear.ccollab.
CollaboratorBasicDataSourceFactory
```

   (On Windows platforms, replace ":" in the command above with ";")

6. Enter your password when prompted and specify the encoding format as "aes128". The program will then output the encoded string, which has a prefix of "$2$" to indicate that AES-128 obfuscation is being used. For example, the pre-obfuscation password "testpass" would become "$2$Nobujw9X9ZJsSOYapNZh+w==".

7. Replace the value of the existing "connectionPassword" attribute with the string composed in the prior step. The Realm tag will look similar to:

```
<Realm ... className="com.smartbear.ccollab.auth.
CollaboratorJNDIRealm" connectionPassword="$2$Nobujw9X9ZJsSOYapNZh+w=="/>
```

8. Save the `ROOT.xml` file.

9. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

**AES-256 obfuscation:**

*Note: base64 is recommended due to simplicity unless your environment absolutely requires this level of obfuscation.*

1. To enable AES-256 encryption you need to install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. They can be downloaded from the Oracle's website: Java Cryptography Extension for JRE 7 and Java Cryptography Extension For JRE 8.
   Download and install the JCE package as described in the README.TXT file shipped within the downloaded archive.

2. Stop your Collaborator server instance. See Platform-Specific Notes 87 for instructions for Windows, Linux, Solaris and BSD platforms.

3. Open the `<server install path>/tomcat/conf/Catalina/localhost/ROOT.xml` file in your chosen text editor.

4. Locate the `<Realm ... />` XML tag. This tag declares the LDAP (JNDI) authentication resource that Collaborator uses.

5. Change the "className" attribute of the Realm tag to `"com.smartbear.ccollab.auth.CollaboratorJNDIRealm"`:

   ```
   <Realm ... className="com.smartbear.ccollab.auth.
   CollaboratorJNDIRealm"/>
   ```

6. Using the command line interface, in the `<server install path>/tomcat/` directory of your Collaborator instance, execute the following command line:

   ```
   java -cp webapps/ROOT/WEB-INF/classes;webapps/ROOT/WEB-INF/lib/*;lib/
   tomcat-dbcp.jar;bin/tomcat-juli.jar com.smartbear.ccollab.
   CollaboratorBasicDataSourceFactory
   ```

   (On Windows platforms, replace ":" in the command above with ";".)

7. Enter your password when prompted and specify the encoding format as "aes256". The program will then output the encoded string, which has a prefix of "$3$" to indicate that AES-256 obfuscation is being used. For example, the pre-obfuscation password "testpass" would become "$3$w8MhPu6t7vobGeNvTx8RoA==".

8. Replace the value of the existing "connectionPassword" attribute with the string composed in the prior step. The Realm tag will look similar to:

   ```
   <Realm ... className="com.smartbear.ccollab.auth.
   CollaboratorJNDIRealm" connectionPassword="$3$w8MhPu6t7vobGeNvTx8RoA=="/>
   ```

9. Save the `ROOT.xml` file.

10. Restart your Collaborator server instance. The server should come up without issue with the obfuscated password.

## 3.2.6 Configuring HTTPS

The basic Collaborator installation configures the server to handle requests over standard HTTP. In many environments this is sufficient as the network is trusted. However some organizations require that all network applications are secured with Transport Layer Security (TLS) or Secure Sockets Layer (SSL). Collaborator supports HTTP over TLS (or HTTPS), but this requires additional manual server configuration:

# What you need

### Certificates

In order to authenticate to clients, the Collaborator server must have a "Certificate" that serves as proof of identity. Certificates come in two forms: Certificate Authority (CA) signed certificates and "self-signed" certificates. CA signed certificates provide an additional level of security because they can be automatically verified and do not rely on human verification. By providing you a certificate, the Certificate Authority is vouching for your identity. Software systems such as web browsers and the Java Runtime Environment (JRE) include the public keys of the trusted Certificate Authorities that are used to verify server certificates were vouched for by a trusted CA. To acquire a CA signed certificate, contact the appropriate person in your IT department.

Self-signed certificates have the advantages of being free and easy to generate. The disadvantage of self-signed certificates is that they cannot be automatically verified. Instead, their security relies on users verifying the certificate signature against a signature obtained through another, ideally secure, channel.

### Keystores

A keystore is Java's mechanism for storing cryptographic keys and certificates. The persistent form of a keystore is a password protected file on disk. A keystore may contain "key entries" that allow applications with access to the keystore to authenticate themselves to users or other services. A keystore is also used to store "trusted certificate entries" that contain the public keys of services that are trusted and certificate authorities that are trusted to vouch for services.

To enable HTTPS in Collaborator, you will need to create a keystore with the server's certificate (either CA- or self-signed) and if CA-signed, the certificate authority's "chain certificate". Follow the instructions below to create a keystore with the type of certificate you will be using to secure Collaborator.

# Creating a keystore for Collaborator

## Creating a private key and certificate in your keystore

To generate a keystore file, use the Java `keytool` utility. You can find the keytool utility in the `$JAVA_HOME/bin` directory.

Running the keytool with the following arguments will create a new private key, install a self-signed certificate and generate a new `collab.ks` keystore file at `<Collaborator Server>/tomcat/conf/` directory:

```
$JAVA_HOME/bin/keytool -genkeypair -alias tomcat -keyalg RSA -
validity 360 -keysize 2048 -keystore <Collaborator Server>/tomcat/
conf/collab.ks
```

**Tip:** We recommend keeping the keystore file along with Tomcat configuration (for example at `<Collaborator Server>/tomcat/conf/collab.ks`), yet that is not strictly necessary. You can specify any other name or location. In further instructions we will use the `<collab-keystore-path>` placeholder instead of the actual keystore path.

You will be prompted to specify keystore password and answer a series of questions. When you are prompted to enter your first and last name, enter the host and domain name you intend to use to address the Collaborator server instead. For example:

```
Enter keystore password: <collab-keystore-password>
Re-enter new password: <collab-keystore-password>
What is your first and last name?
[Unknown]: collab.acme.com
```

Once you complete the rest of the prompts, you have created a private key and a self-signed certificate.

Make sure to remember the keystore location and password you set, as you must include them in your server configuration.

For many organizations, this is sufficient to guarantee security. However, web browsers and the Collaborator client will request confirmation from end users, unless the certificate is not signed by an existing certificate authority.

If you wish to have a certificate authority sign your server certificate, please follow the steps described in Create Certificate Signing Request 113.

### Create certificate-signing request

After the steps described above, you can create a certificate-signing request with the following command:

```
$JAVA_HOME/bin/keytool -certreq -alias tomcat -keystore <collab-keystore-path> -file tomcat.csr
```

You will be prompted to enter the keystore password. This should create a file named "tomcat.csr" which you will need to provide to your signing authority. This may be Verisign, Thawte, another certificate vendor, or a group internal to your organization. Once they have verified the information provided, the signing authority will send you a certificate file.

### Importing CA-signed Certificates

If you are using a CA-signed certificate, you will need the server certificate and the chain certificate of the certificate authority (or root certificate). The CA chain certificates are publicly available, but the instructions for acquiring them vary by certificate authority. To get the CA chain certificate, ask the IT person in your organization responsible for procuring SSL certificates for services. Once you have the certificates, import them into a new keystore using the Java `keytool` utility. You can find the `keytool` utility in the `$JAVA_HOME/bin` directory.

To import the CA chain certificate:

```
$JAVA_HOME/bin/keytool -importcert -alias root -keystore <collab-keystore-path> -trustcacerts -file <path to chain certificate file>
```

To import the server certificate:

```
$JAVA_HOME/bin/keytool -importcert -alias tomcat -keystore <collab-keystore-path> -trustcacerts -file <path to server certificate file>
```

Remember the keystore password you select, as you will need this when configuring the Collaborator server.

### Alternate method to import server and chain certificate simultaneously

Some popular signature authorities now provide both a primary and intermediate certificate. The steps below may work more effectively in those cases than the steps above.

First, assemble a single file containing all of the certificates (primary, intermediate, and server) provided to you by your signature authority.

- On UNIX systems:

```
cat primary.cert intermediate.cert tomcat.cert > combined.txt
```

- On Windows: In Notepad, copy and paste all the provided certificates into a single file, then save it as *combined.txt*.

- Once you have that, run the following command:

```
keytool -importcert -alias tomcat -keystore <collab server install
dir>/tomcat/conf/collab.ks -trustcacerts -file combined.txt
```

After that, run the following command:

```
$JAVA_HOME/bin/keytool -importcert -alias tomcat -keystore <collab-
keystore-path> -trustcacerts -file combined.txt
```

This will import all three (or more) certificates at the same time, establishing the correct chain of trust.

## Configuring the Collaborator Server

The final step in configuring the Collaborator server is to instruct the server to use SSL and tell it what certificates to use. This is done by editing the `server.xml` file located in `<collab server install dir>/tomcat/conf`. In server.xml, locate the "Connector" element and ***replace it*** with the following:

```
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443" enableLookups="true" disableUploadTimeout="true"
acceptCount="100" maxThreads="100"
scheme="https" secure="true"
SSLEnabled="true" sslEnabledProtocols="+TLSv1.3+TLSv1.2"
keystoreFile="conf/collab.ks" keystorePass="<collab-keystore-
password>"
clientAuth="false" sslProtocol="TLS"/>
```

In this XML snippet, you have to replace the keystore password, and keystore path if necessary.

The snippet above, enables the `TLSv1.3` and `TLSv1.2` protocols. To use the TLS1.3 protocol your Collaborator server should be running on OpenJDK 11 or higher, since support for this protocol was added in OpenJDK 11.

**Tip:** If you need to specify a full path to the keystore file in the server.xml then use the file protocol notation:

```
keystoreFile="file:///c:/program%20files/collaborator%20server/
tomcat/conf/collab.ks"
```

Restart the server and it will be operating over SSL on port 8443.

Be sure to update the External URL[180] in the server settings to reflect the correct https URL and enable the Secure authentication cookies[196] setting to send login cookies over HTTPS.

## Impact on Clients

After you have generated the keys for the server, you may wish to distribute a keystore and its signature to users so they can validate the certificate when asked. To do this, first export the certificate to a file:

```
$JAVA_HOME/bin/keytool -exportcert -keystore <collab-keystore-path> -
alias tomcat > <export path>collab.cert
```

Then print the certificate information that you will need to distribute. Of particular interest are the fingerprints. To do this, run keytool as below:

```
$JAVA_HOME/bin/keytool -printcert -file collab.cert
```

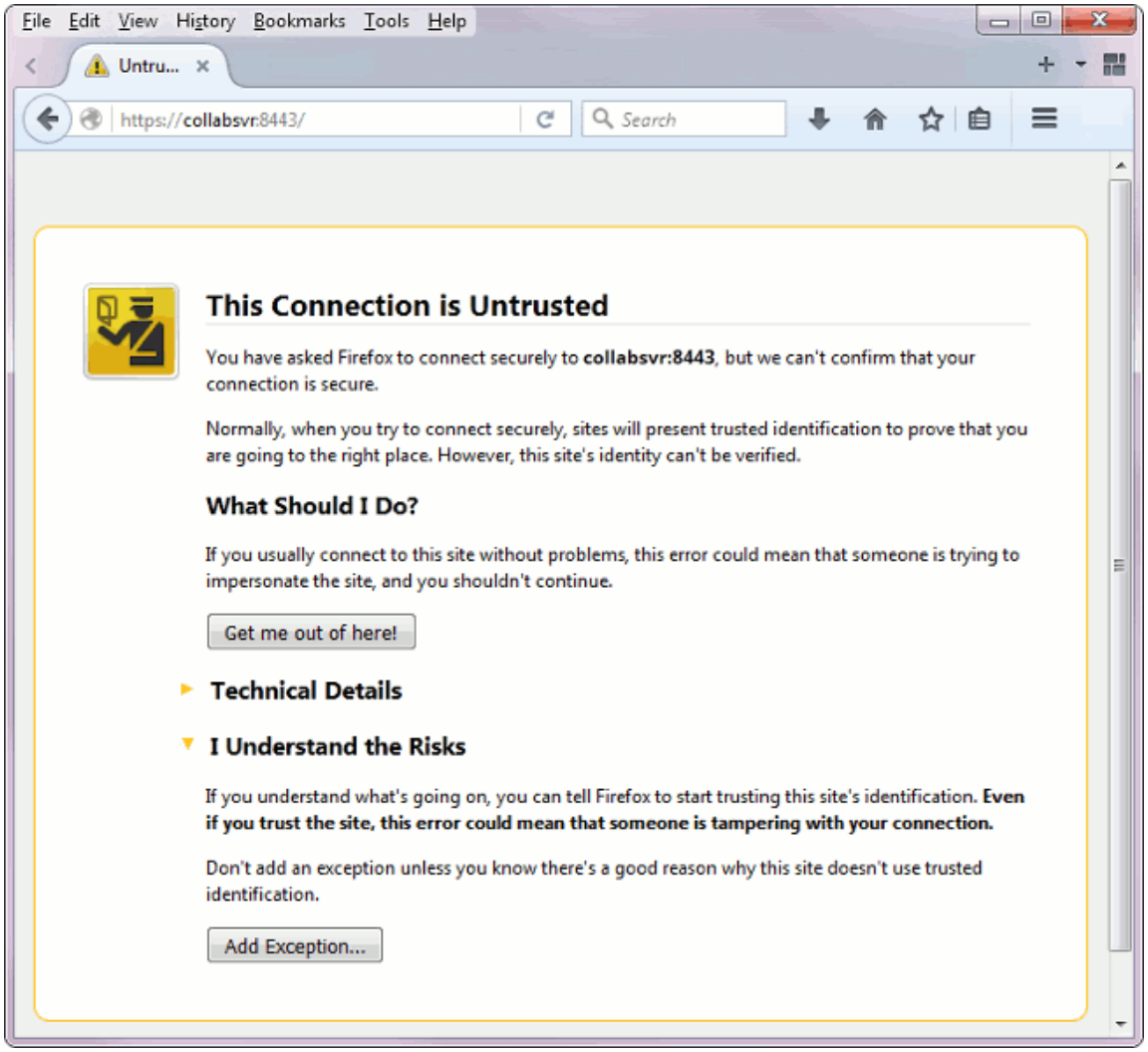
This will print something like the following:

```
Owner: CN=collab.aus.smartbear.com, OU=SmartBear, O=SmartBear,
L=Austin, ST=TX, C=US
Issuer: CN=collab.aus.smartbear.com, OU=SmartBear, O=SmartBear,
L=Austin, ST=TX, C=US
Serial number: 463251eb
Valid from: Fri Apr 27 14:41:31 CDT 2007 until: Thu Jul 26 14:41:31
CDT 2007
Certificate fingerprints:
 MD5: 67:D7:74:5E:72:9D:B2:82:88:3F:33:AA:A0:41:01:F0
 SHA1: E2:4A:1F:9B:9A:38:0F:6B:7B:33:12:73:1B:50:76:30:AC:A6:B2:EA
```

If the certificate used to secure Collaborator is a CA signed certificate from a trusted CA, there will be very little impact on users. The only thing that they must do is configure their clients[494] with the correct HTTPS url.

If you are using a self-signed certificate or the certificate cannot be automatically verified, users will be asked to verify the certificate by comparing its signature to the published signatures for the service. The exact nature of this confirmation varies by client, but the process is the same: look at the certificate the server presents and confirm that its signature matches the published signature.

In the web browser, the certificate confirmation dialog looks something like this:




The above screenshot is from Firefox, but other browser users will see a similar dialog. The users will need verify that the fingerprint (or signature) of the certificate by comparing it to the signature of that you distributed for the server.

Certificates that are accepted permanently are stored in a local keystore in $HOME/.smartbear/trustedcertificates. This file can safely be deleted to remove all trusted certificates. If this is done, the user will have to reauthorize any certificates that are self-signed (or signed by a CA that is not in the Java trusted keystore).

If you want each client to automatically recognize and authorize your internal CA without being prompted to do so, one solution is to accept the key with one client, and then distribute the trustedcertificates file that is generated.

To reiterate, the default location for Unix clients is: `~/.smartbear/trustedcertificates` and for Windows clients: `%USERPROFILE%\.smartbear\trustedcertificates`

If you are capable of distributing files to all user's machines, distributing that file to those locations will prevent remaining users from seeing the prompt.

Unlike other clients, Collaborator Visual Studio Extension cannot read certificates from Java keystore. Therefore, to work with HTTPS servers from Collaborator Visual Studio Extension, users should manually install the server's certificate into Windows keystore.

**Note on self-signed certificates:** You have to use certificate signed with Certificate Authority (CA). It can be any CA - even yourself. And you must install that CA certificate in trusted authorities. You cannot use self-signed certificate directly.

See detailed instructions in the Configuring HTTPS Connection 574 section of Visual Studio Extension Configuration.

## Redirect HTTP requests to HTTPS

To enable the redirection from HTTP to HTTPS, you will need to create both HTTP and HTTPS Connectors in `server.xml` and add the appropriate attribute to the `web.xml` file. For example:

```
<Connector acceptCount="100" compression="on"
connectionTimeout="20000" disableUploadTimeout="true" maxThreads="95"
minSpareThreads="25" port="8080" redirectPort="8443"/>
<Connector port="8443" minSpareThreads="5" maxSpareThreads="75"
enableLookups="true" disableUploadTimeout="true" acceptCount="100"
maxThreads="100" scheme="https" secure="true" SSLEnabled="true"
sslEnabledProtocols="+TLSv1.3+TLSv1.2" keystoreFile="conf/collab.ks"
keystorePass="<collab-keystore-password>" clientAuth="false"
sslProtocol="TLS"/>
```

The attribute below should be added after all servlets inside of <webapp> element in the `tomcat\webapps\ROOT\WEB-INF\web.xml` file:

```
<security-constraint>
            <web-resource-collection>
                <web-resource-name>HTTPSonly</web-resource-name>
                <url-pattern>/*</url-pattern>
            </web-resource-collection>
            <user-data-constraint>
                <transport-guarantee>CONFIDENTIAL</transport-
guarantee>
            </user-data-constraint>
</security-constraint>
<security-constraint>
            <web-resource-collection>
                <web-resource-name>HTTPSorHTTP</web-resource-name>
                <url-pattern></url-pattern>
            </web-resource-collection>
            <user-data-constraint>
                <transport-guarantee>NONE</transport-guarantee>
            </user-data-constraint>
</security-constraint>
```

This means that the entire application is meant to be HTTPS only, and the container should intercept HTTP requests for it and redirect them to the equivalent "https://" URL. The exception is certain requests having the URL patterns that match the “HTTPSorHTTP” web resource collection. This will allow your end users to use both HTTP or HTTPS for the specified URL patterns.

**Notes:**

- Do not forget to restart your Collaborator server after the changes have been made.
- This redirection will not work correctly for the Collaborator desktop clients. Desktop clients should be configured to use the "https" protocol to avoid the redirection.

## Additional Resources

- Tomcat SSL Configuration HOW-TO
- Tomcat HTTP Connector Configuration
- Keytool documentation

### 3.2.7 LDAP and Active Directory Authentication

By default the Collaborator server authenticates users against the users in its database. For large organizations with hundreds or thousands of users in multiple product groups, it is simply impractical to add each would-be Collaborator user to the database. For this reason, Collaborator can integrate with an existing LDAP directory or Active Directory to perform user authentication.

***Note:*** LDAP/AD authentication *is supported in Collaborator Team and Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

When LDAP authentication is configured, Collaborator authenticates users attempting to login against their entry in the directory. When a user logs in 315 for the first time, a user account is created for them automatically in Collaborator to store their user preferences. Users must be authenticated by logging into the Collaborator server through the web client before they will be able to work via GUI client, command-line interface and other clients.

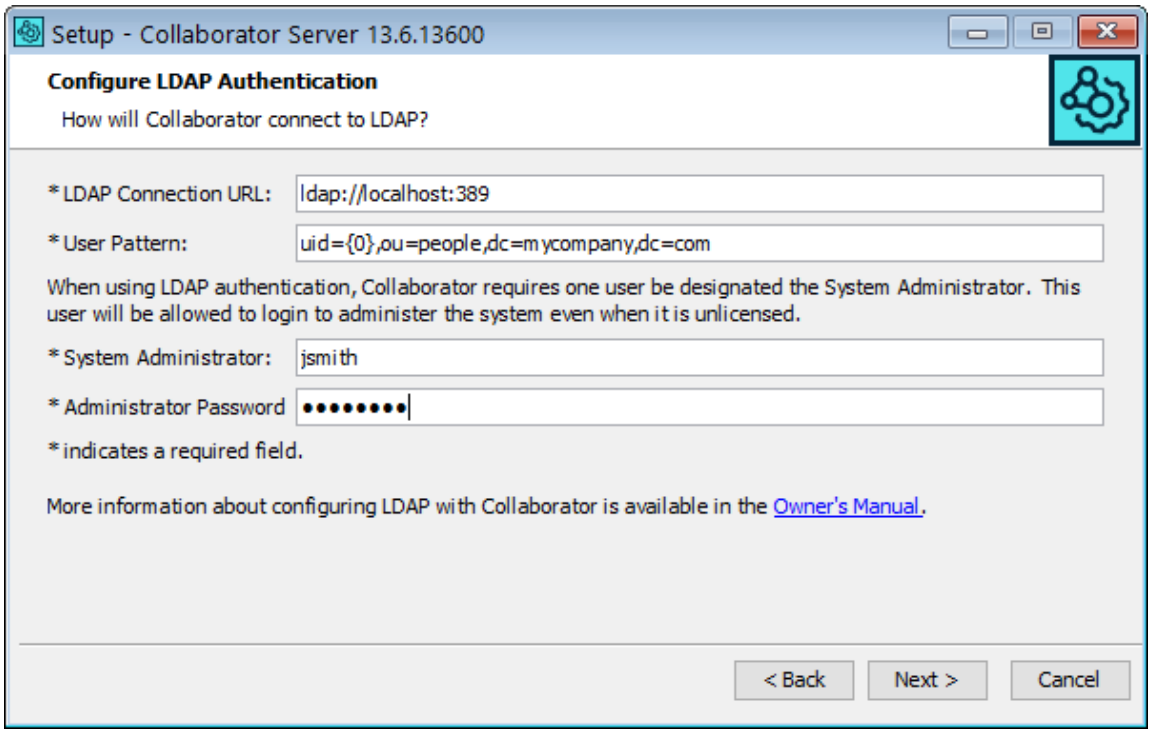
> **Warning:** Only users with accounts can be assigned to roles within reviews, so it is not possible to add a user to a review who has not previously logged in to Collaborator.

Internally, Collaborator uses the Tomcat Servlet Container's JNDI Realm for LDAP authentication. For a detailed description of how it works and for complete configuration information, consult the JNDI Realm Documentation.

## LDAP Authentication

The installation wizard 70 provides a screen to perform basic configuration of LDAP authentication. This wizard minimally configures Collaborator to use LDAP authentication. For advanced configuration, see Advanced Configuration 122.

When LDAP authentication is selected, you are prompted for the following items:




- **LDAP Connection URL** - This is a URL where Collaborator can connect to the LDAP server. The format of the URL is as follows: `ldap://servername:port`
  The standard default port for LDAP is 389.

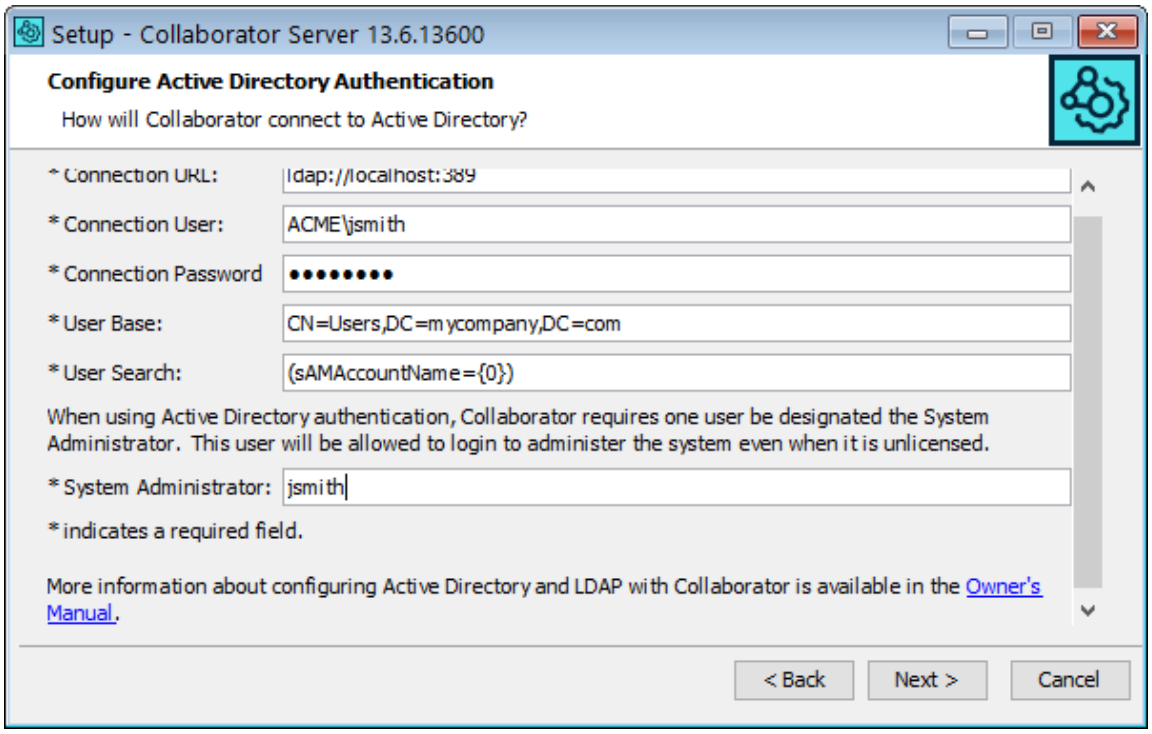
- **User Pattern** - This string instructs &product-name; how to locate a user within LDAP. The format of the string is the same as the LDAP-distinguished name of a user with the username replaced with the {0} string. For example, `uid={0},ou=people,dc=mycompany,dc=com`

- **System Administrator** - The LDAP account of a user who will be Collaborator server administrator. For example, if your LDAP distinguished name is `uid=jsmith,ou=people, dc=mycompany,dc=com`, then your LDAP account is `jsmith`
  The Collaborator server administrator will always be able to log in and administer license codes and user accounts. This account takes the place of the admin account when using internal authentication.

## Active Directory Authentication

Microsoft Active Directory is an LDAP compliant directory and can be used to authenticate users to Collaborator. The installation wizard 70 provides a screen to perform basic configuration of Active Directory authentication. This wizard minimally configures Collaborator to use AD authentication. For advanced configuration, see Advanced Configuration 122.

When AD authentication is selected, you are prompted for the following items:




- **Connection URL** - This is a URL where Collaborator can connect to the LDAP server. The format of the URL is as follows: `ldap://servername:port`
  The standard default port for LDAP is 389.

- **Connection User** and **Connection Password** - The user name (in DOMAIN\username format) and password which Collaborator will use to connect to Active Directory to find the user records. If anonymous connections are allowed to your directory (not typical), then these attributes are not required.

- **User Base** - Specifies the starting point in the Active Directory for the subtree containing users.

- **User Search** - Specifies a search pattern of how to locate a user within the Active Directory, with {0} marking where the username should be substituted. For example, `(sAMAccountName={0})`. The filter may combine multiple conditions using Boolean operators: AND (needs to be SGML encoded to "`&`" because the configuration file is an XML document); OR (|) and NOT (!).

- **System Administrator** - The LDAP account of a user who will be Collaborator server administrator. For example, if your LDAP distinguished name is `cn=jsmith,ou=people, dc=mycompany,dc=com`, then your LDAP account is `jsmith`
  The Collaborator server administrator will always be able to log in and administer license codes and user accounts. This account takes the place of the admin account when using internal authentication.

## Advanced Configuration

The basic configuration provided by the installer sets up the simplest possible LDAP or AD configuration. Many more sophisticated configurations are possible, but they may require manually editing the context configuration file (`<collab server install dir>/tomcat/conf/Catalina/localhost/ROOT.xml`), namely, the `Realm` configuration element which defines authentication information.

The JNDI Realm Documentation describes the different operational modes and explains each of the configuration attributes.

Performing the basic configuration in the installation wizard will result in the following values of the Realm element of `ROOT.xml` file.

LDAP:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionURL="ldap://mycompany.com:389"
       userPattern="uid={0},ou=people,dc=mycompany,dc=com"
       allRolesMode="strictAuthOnly"
/>
```

Active Directory:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionName="your_ldap_username"
       connectionPassword="xxxx"
       connectionURL="ldap://mycompany.com:389"
       referrals="follow"
       userBase="CN=Users,DC=mycompany,DC=com"
       userSearch="(sAMAccountName={0})"
       userSubtree="true"
       allRolesMode="strictAuthOnly"
/>
```

When using LDAP or Active Directory, please check that the "collaborator-authentication" parameter is "false":

```
<Parameter description="Is the Code Collaborator database used for
authentication?" name="collaborator-authentication" override="false"
value="false"/>
```

**Warning:** Modifying the ROOT.xml file will cause Tomcat to dynamically reload the Collaborator application, terminating any active sessions. Changes to ROOT.xml should be done in the context of stopping and restarting the Collaborator service (that is, in a production environment coordinating the restart with user activity), regardless of whether the service itself is actually stopped and restarted, or just reloaded by Tomcat.

### Obfuscate LDAP Passwords

As a more secure alternative to storing LDAP passwords as plain text, you can obfuscate them. Collaborator supports two forms of obfuscation: base64 encoding, and base64-encoded AES. For instructions on enabling LDAP passwords obfuscation, see Security Considerations 107.

### Restrict Access

If you need to restrict access to Collaborator, we recommend that you configure your Realm definition to use `userSearch` for searching for user accounts rather than `userPattern` for direct bind (see Active Directory Configuration 120 for examples). With `userSearch`, you can expand your search criteria to include only members of the specified group or groups. For example, the following `userSearch` would restrict access to only members of the `ccusers` security group:

```
userSearch="(&amp;(sAMAccountName={0})(memberOf=CN=ccusers,
OU=Security Groups,OU=Accounts and Groups,DC=xxxx,DC=xxxx,DC=com))"
```

Please note that the ampersand (&), needs to be SGML encoded to "`&`" because the configuration file is an XML document.

If you need to broaden the search criteria, you can use the OR operator, "|". For example, to allow users who are in group "foo" or group "bar" (or both), you might use:

```
userSearch="(&amp;(sAMAccountName={0})(|(memberOf=CN=foo,OU=groups,
DC=xxxx,DC=com)(memberOf=CN=bar,OU=groups,DC=xxxx,DC=com)))"
```

Above, the inner OR requires that the user be a member of group "foo" or group "bar". The outer AND ensures that that user's login also matches the one provided to the Collaborator login form.

### Administrator Rights

When users first log in to Collaborator, their user account is created automatically, as a standard user account. It is possible to have users in certain roles (see Restricting Access above) to automatically receive administrator privileges. This configuration is done in the *<collab server install dir>*/tomcat/conf/Catalina/localhost/ROOT.xml file.

Near the bottom of the ROOT.xml file, there is a section of server parameters (Parameter XML elements). These are name-value pairs of configuration options. To configure certain LDAP groups to be assigned administrator rights, create an admin-roles parameter and for its value specify a pipe-separated ("|") list of LDAP group names. You must also add 'userRoleName="memberOf"' to the Realm configuration.

```
<Parameter description="Automated Collaborator admin rights"
name="admin-roles" override="false" value="CN=foo,OU=groups,DC=test,
DC=com|CN=bar,OU=groups,DC=test,DC=com|CN=baz,OU=groups,DC=test,
DC=com"/>
```

The admin-roles parameter requires an exact match, including case. New users that are members of the specified LDAP groups will be given administrator privileges when their accounts are created. Users that already have accounts in Collaborator will not automatically be promoted to administrator. Those privileges will need to be assigned manually from the Admin screen.

## Secure LDAP (LDAPS)

Configuring Collaborator to communicate securely with an LDAP server using LDAPS (LDAP over SSL) can be done, but requires manual configuration outside of the installer wizard. To configure LDAPS, first install Collaborator configured for normal LDAP access. The service will start automatically upon completion of the install, so you will need to shut it down to continue configuration. Open the context configuration file (*<collab server install dir>*/tomcat/conf/Catalina/localhost/ROOT.xml) in a text editor and find the Realm configuration element. It will look something like the following:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
      connectionURL="ldap://localhost:389"
      userPattern="uid={0},ou=people,dc=mycompany,dc=com"
      allRolesMode="strictAuthOnly" />
```

If you are configuring Collaborator for use with Microsoft Active Directory using LDAPS, follow the Active Directory instructions 120 to make a best effort to configure the realm for your Active Directory server. Do not worry if you do not get it exactly right or cannot test the connection because the server refuses insecure connections. That issue can be resolved once connectivity is established.

To the realm configuration above (or your Active Directory realm configuration), you will need to add an attribute 'protocol' with the value 'ssl' and you will probably need to change the 'connectionURL' attribute to an LDAPS url. The updated configuration below highlights the changes:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionURL="ldaps://localhost:636"
       userPattern="uid={0},ou=people,dc=mycompany,dc=com"
       allRolesMode="strictAuthOnly"
       protocol="ssl" />
```

Depending on the LDAP server's SSL certificates, this configuration *may* be enough to establish the connection. However, often companies generate their own SSL certificates signed by their own Certificate Authority (CA) certificate. Unless additional measures are taken, these certificates may not be trusted so Collaborator will still not connect to the LDAP server.

To establish trust, you need to import the public key of either the Certificate Authority or the public key of the LDAP server as a trusted certificate to Collaborator keystore file:

1. Get the certificate file from your LDAP or network administrator.

2. Locate the keystore file which you have generated while configuring Collaborator HTTPS connection[112].

   Default location is <Collaborator Server>/tomcat/conf/collab.ks, yet that could be changed while generating keystore.

3. Use Java's `keytool` utility to import the server's certificate to Collaborator keystore file. You can find the keytool utility in the `$JAVA_HOME/bin` directory:

```
$JAVA_HOME/bin/keytool -importcert -alias ldapserver -keystore
<collab-keystore-path> -trustcacerts -file <path-to-chain-
certificate-file>
```

For more information on command-line arguments of the keytool utility, see keytool documentation.

4. Most likely you will be prompted to confirm the validity of the certificate. It is imperative for the security of the overall system that you verify the key matches the trusted material. Before accepting the certificate, you should contact the administrator that sent you the certificates and verify that the certificate fingerprints that you see match the certificate fingerprints that they intended to send you.

5. The final step is to configure Collaborator to use the keystore. Open the `<Collaborator Server>/ccollab-server.vmoptions` file in a text editor, and add the following lines to it:

```
-Djavax.net.ssl.trustStore=<collab-keystore-path>
-Djavax.net.ssl.trustStorePassword=<collab-keystore-password>
```

6. Restart the Collaborator server.

Upon restart, the Collaborator service should be connecting to the LDAP server via SSL. If you are still getting errors, check that the other LDAP configuration options have been configured correctly. If you are using Active Directory, it is now worth revisiting the Active Directory configuration above.

Finally, a note on troubleshooting SSL connections: adding the following line to the `ccollab-server.vmoptions` file will enable Java's network debug logging.

```
-Djavax.net.debug=all
```

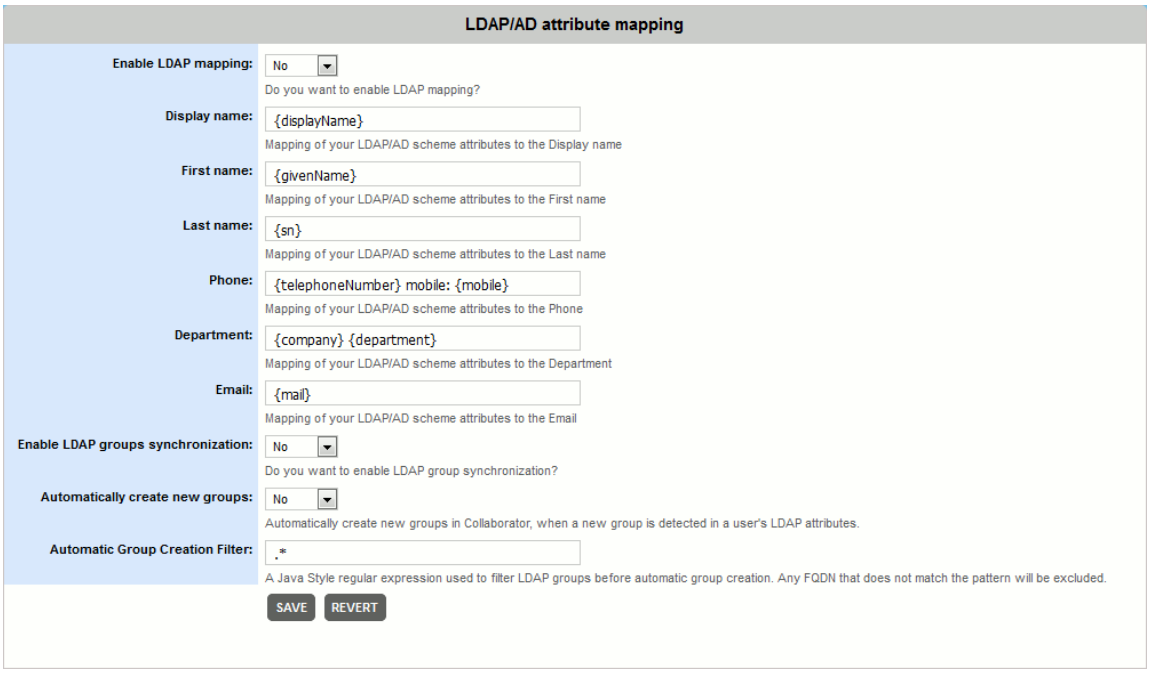
Upon restarting Collaborator, this information will be written to <collab server install dir>/output.log. If you need assistance interpreting this log, contact SmartBear Customer Support 32.

Note: Do not run in a production environment with network debug logging enabled. This will severely impact the performance of the system and will also consume vast quantities of disk space.

## Synchronize Users and Groups

You may configure user and group synchronization between Collaborator and the LDAP directory or Active Directory. In this case, Collaborator will retrieve user properties (name, phone, email, and so forth) and their membership in groups from the LDAP directory or Active Directory when the users login. Additionally, you can select whether to create new groups automatically and specify regular expression for automatic group creation.

To enable synchronization on Active Directory systems, you will need to open the LDAP Settings 198 tab of General settings, enable the respective properties and possibly adjust the attribute mapping configuration.




In order to perform synchronization on LDAP systems, you will need to configure the above-mentioned LDAP Settings[198] and also need to modify the ROOT.xml. Namely, you will need to add the following fields: the `connectionName` and `connectionPassword` fields which define a user account the Collaborator will use to connect to LDAP to find the group membership user records, and the `roleBase` and `roleSearch` fields which define the base entry for the role search and the search filter for selecting role entries.

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionName="cn=read-only-admin,dc=example,dc=com"
       connectionPassword="xxxx"
       connectionURL="ldap://xxxx.com:389"
       userPattern="uid={0},dc=example,dc=com"
       roleBase="dc=example,dc=com"
       roleSearch="(&amp;(objectClass=groupOfUniqueNames)(uniqueMember={0}))"
       allRolesMode="strictAuthOnly"
/>
```

**Technical details**

For mapping user membership in groups Collaborator uses the group's fully qualified domain name (FQDN) retrieved from the LDAP or AD. It checks if some of existing user groups[226] has matching FQDN and adds the user to this group on success. Otherwise, it can create new group (if automatic group creation is enabled and group name matches filter) and adds the user to the new group.

To name the new group Collaborator uses the first entry of group's fully qualified domain name (FQDN). For example, a group having the "ou=`ccusers`,dc=example,dc=com" FQDN will have the `ccusers` title. If some other group already has the same title, Collaborator will append the ordinal number to the group title: `ccusers1, ccusers2` and so on.

If users have multiple emails specified on the LDAP or Active Directory side, Collaborator will retrieve them and use the first of the emails for notifications. Email addresses should be separated by comma.

Collaborator checks existing groups created via LDAP or Active Directory synchronization and actualizes user membership in those groups on every login. Such algorithm allows to keep a consistency between Collaborator and LDAP or Active Directory.

## Troubleshooting LDAP:

The directory administrator that manages the directory will be a valuable resource in resolving the issue, either directly, or in conjunction with SmartBear's Customer Support team.

If the directory administrator is unavailable, SmartBear's Customer Support team can help you resolve configuration issues, but often do not have enough information about the directory schema, permissions, and so on, to efficiently resolve issues. We will walk you through several basic configurations that we have seen work with other directories. Having an LDAP browser tool available (there are many good free and commercial browsers available) when you call will help answer some of the questions required to properly configure your server. In some cases, we will still need to discuss details with your directory administrator.

There is a process that we go through to help people debug their LDAP issues. You can do it on your own (of course, if you get stuck you can always contact technical support). We recommend using the JXplorer LDAP browser for this task because it is a Java tool and as such it uses the same underlying LDAP library that Collaborator will use. Here is the process:

1. Download and install JXplorer, following the recommended installation guidelines.

2. Start up JXplorer.

3. From the File menu, choose Connect.

4. In the connect dialog ("Open LDAP/DSML Connection"), specify the following:

   **Host**: the hostname portion of the connectionURL attribute from the Realm declaration in ROOT.xml.

   **Port**: the port portion of the connectionURL attribute from the Realm declaration in ROOT.xml. Usually this is 389, which is the default if unspecified. Some Active Directory configurations require connecting to the "Global Catalog" which is port 3268 (you may see errors that say "DomainDnsZones.foo.bar.com" which means you need to use this port).

**Base DN**: The value of the userBase attribute from from the Realm declaration in ROOT.xml.

**Security Level**: User + Password

**Security User**: The value of the connectionName attribute from the Realm declaration in ROOT.xml. This should be a name that looks like an email address (jason@...) or something that looks like an LDAP distinguished name (uid=jason,ou=people, dc=mycompany,dc=com)

**Security Password**: The value of the connectionPassword attribute from the Realm declaration in ROOT.xml.

Click Ok to connect to establish the connection.

If the connection establishes normally, you should see the Explore tree populate with some nodes that represent entities in your directory. Note: You may see a Error that "Search partially failed!" This seems to be normal and is a consequence of the way Active Directory does its searching.

If the connection fails to establish normally, check the error message that explains the failure.

Is it a network issue? This could mean that the hostname or port number is wrong or that a firewall (local or on the network somewhere) is preventing the connection to the LDAP server. It could also mean that the LDAP server is simply offline. Double check your connection information and firewalls and if you still cannot connect, contact your LDAP administrator. (This error looks like: "Error opening connection: 192.168.10.441:389")

Is it a login issue? The specific error messages that come back will depend on the LDAP server in question, but if it is a "security" related message it probably means the user DN or password is wrong. Double check those and try to reestablish the connection. If that fails, perhaps the LDAP account does not exist or does not have query permissions on the directory. Consult your LDAP administrator for help getting the appropriate access. An example of an error message from an Active Directory server is as follows:

```
Error opening connection:
 [LDAP: error code 49 - 80090308: LdapErr: DSID-0C090334, comment:
AcceptSecurityContext error, data 52e, vece
```

If it is neither of those issues, send the error message verbatim to SmartBear technical support (the JXplorer error dialog accepts Ctrl-C to copy the message to the clipboard), or consult Google or your LDAP administrator.

5. Once the connection is established, it is time to verify the search parameters. From the Search menu, select "Search Dialog".

6. In the Search dialog, confirm that the "Start Searching From" is set to the value of the userBase attribute from the Realm declaration in ROOT.xml.

7. Search Level should be set to "Search Full Subtree".

8. Select the "Text Filter" tab at the bottom of the search dialog.

9. Copy and paste the userSearch attribute from the Realm declaration in ROOT.xml into the text filter box (for example, (sAMAccountName={0})).

10. Replace the {0} with the value of a user that you expect to be able to login. For example, if the user would use the login jason and the filter is (sAMAccountName={0}), change this value to (sAMAccountName=jason).

11. Press the Search button. Exactly one result should be returned. If zero results are returned, then the query is at fault. Some possible causes:

The user name as substituted into the filter is incorrect.

The user does not exist or does not exist within the subtree rooted at userBase.

The filter itself is too restrictive. Filters can be arbitrarily complex with AND and OR clauses. If the filter is complex, we recommend simplifying the query down to only the username portion (the place where the {0} occurs) and verifying that portion and then building up the query to the ultimate query, verifying that at each step the user account is still findable.

If the user is returned and you know that user's password (for example, if it is the Code Collaborator administrator account), verify that the account can login using JXplorer.

12. Select the user from the results tree.

13. Select the Table Editor tab.

14. Copy the `distinguishedName` attribute to the clipboard.

15. From the File menu, select Connect. Follow the login procedure as above, substituting the user's distinguished name (on the clipboard) for User DN and that user's password for Password.

If that login fails, it is possible that the password is incorrect or has been changed or that the account has been disabled or locked. Consult with your LDAP administrator to confirm that the account has "bind access" as necessary.

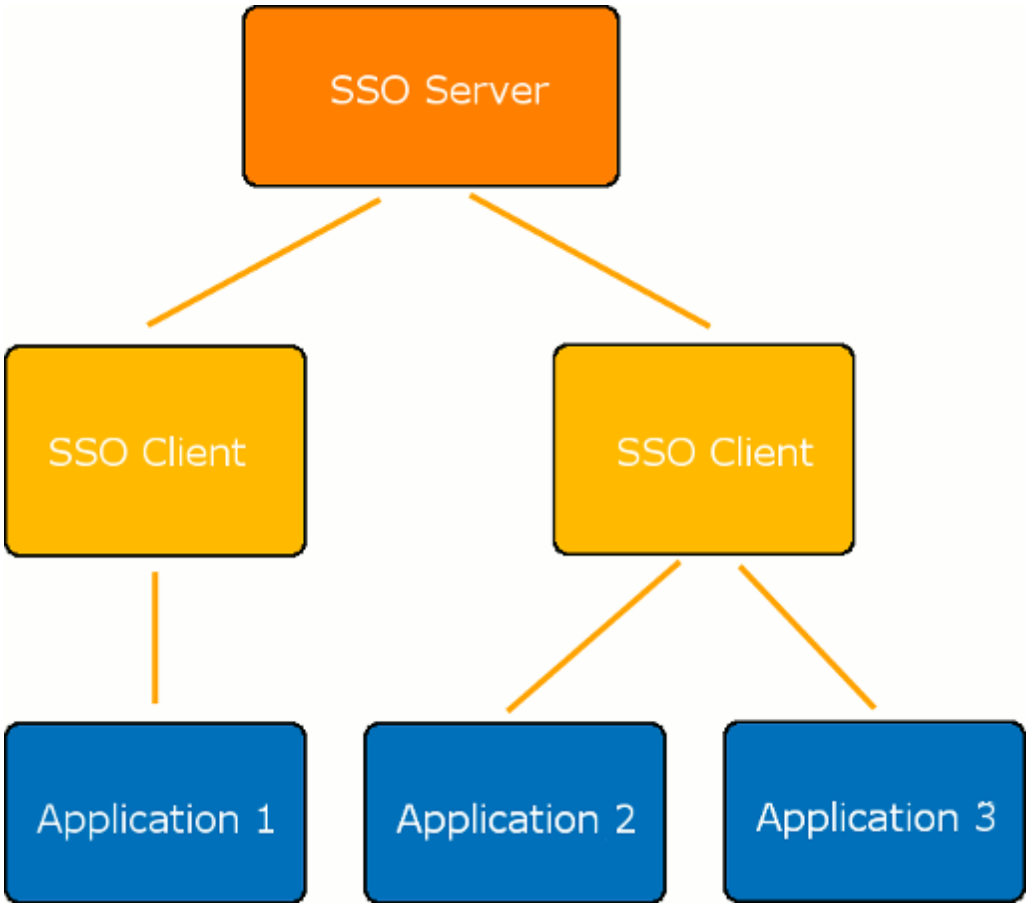
### 3.2.8 Single Sign-On

## General Concept

Single sign-on (SSO) is a user authentication process that permits a user to enter one name and password in order to access multiple applications. The process authenticates the user for all the applications they have been given rights to and eliminates further prompts when they switch applications during a particular session. That is, having entered a login and password once on a central SSO server a user will be automatically logged into other applications, like, issue tracker, source control system, code review system and so on.

***Note:*** *Single sign-on authentication is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page.*

Typically Single Sign-On solutions consist of several components - a SSO server and a number of SSO clients. A SSO server is a component that performs authentication, issues and validates tokens and so forth. SSO clients are intermediate components that can be integrated with various software platforms and applications in order to communicate with the SSO server via some authentication protocol. Most SSO solutions also provide Single Logout functionality - that is they allow users to log out from the application and from the SSO server simultaneously.




The authentication process will consist of the following steps:

1. A user tries to access a Collaborator server.
2. Collaborator recognizes that the user is not logged-in and redirects them to the SSO server.
3. The SSO server authenticates the user, adds some security assertion parameters and redirects back to the Collaborator server.
4. Collaborator detects the security assertion parameters and logs the user in.

5. If a user with the specified credentials is not found, Collaborator creates a new user.

The logout process will consist of the following steps:

1. A user tries to logout from the Collaborator server.

2. Collaborator sends logout request to the SSO server.

3. The SSO server logs the user out and sends the response back to the Collaborator server.

4. Collaborator logs the user out.

## Single Sign-On Implementations in Collaborator

Collaborator supports single sign-on authentication for web client[312]. To use desktop clients[484] (GUI Client, Command-Line Client, Office plug-ins, IDE plug-ins) when single sign-on authentication is enabled, users should generate login tickets[318] and specify them in client connection settings instead of password.

There are several ways to enable single sign-on authentication:

- **via SAML protocol**: If your SSO vendor supports Security Assertion Markup Language (SAML) standard, then you can configure the SSO server and the Collaborator server to use SAML protocol for authentication. Read Configuring SSO via SAML[132] for detailed instructions.

- **via Crowd OpenID protocol**: If you use Atlassian Crowd server, then you can configure it and the Collaborator server to use OpenID protocol for authentication. Read Configuring SSO via Crowd OpenID[145] for detailed instructions.

### 3.2.8.1 Configuring SSO via SAML

This topic describes how to establish single sign-on between an SSO server and Collaborator using Security Assertion Markup Language (SAML) protocol. To learn general principles of how single sign-on operates, see Single Sign-On[130].

In SAML terms, SSO server will act as Identity Provider (IdP) and Collaborator will act as Service Provider.

Below we describe how to establish single sign-on between Collaborator and OneLogin SSO server. Integration with Single Sign-On servers of other vendors is performed in a similar manner.

Enable HTTPS Connections[133]

Install and Configuring SSO Server[133]

## (Optional) Enable HTTPS Connections

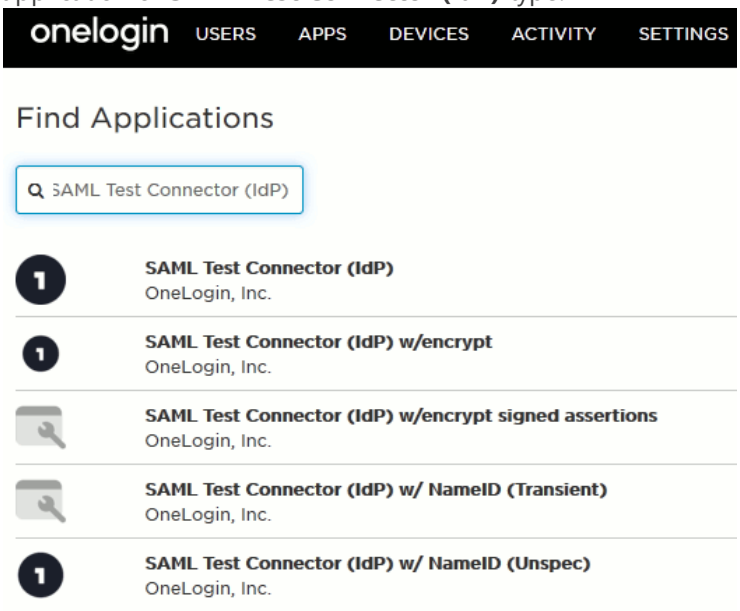
Single sign-on servers mostly use HTTPS connections, thus you will likely need to enable it for Collaborator server as well. For instructions, please see, Configuring HTTPS 111. Do not forget to restart the Collaborator server to apply changes.

## Install and Configure SSO Server

1. Install your preferred SSO server. In this instruction we will use OneLogin SSO server.

2. Log in to your SSO server as Account owner or Super user.

3. Add new application integration: Select **Apps | Add Apps** from the main menu and search for application of **SAML Test Connector (IdP)** type.



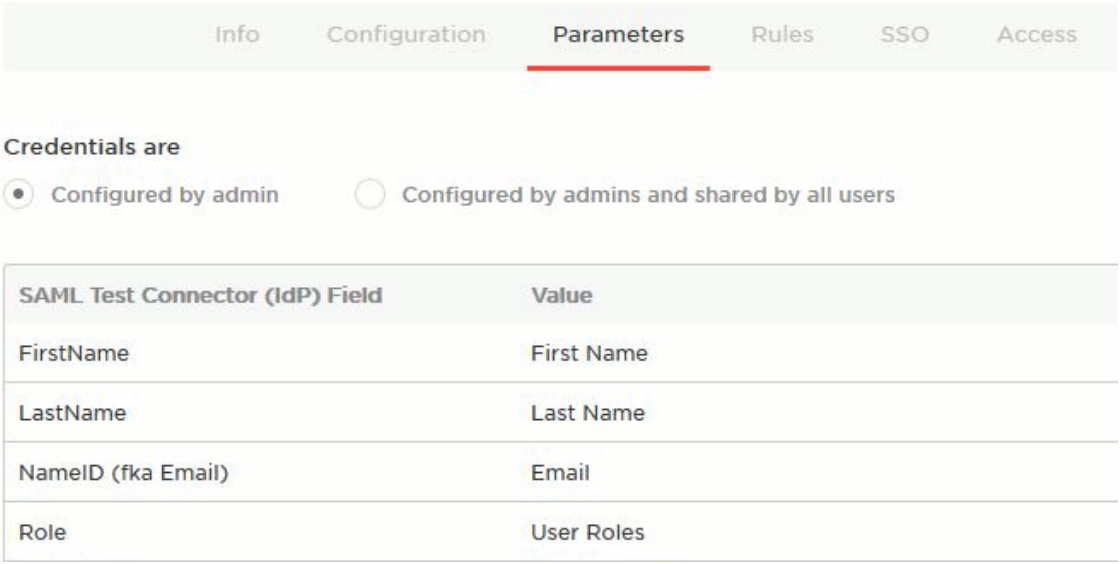

4. Specify application display name and optionally its icons and click **Save**.

5. Switch to the **Configuration** tab and specify the following data:

**Audience** — The URL of the entity that is expected to receive the SAML message. In our case, this will be the URL of your Collaborator server.
For instance: `https://yourcollabserver.com`

| | |
|---|---|
| | **Important.** Your Collaborator server must be accessible to SSO server. Configure a firewall or enable tunneled connections to do so. |
| **ACS (Consumer) URL** | The URL of Assertion Consumer Service (ACS) where the SAML response will be sent to.<br>Collaborator servers have this service at the following endpoint: `https://yourcollabserver.com/services/saml/acs` |
| **ACS (Consumer) URL Validator** | The regular expression to validate ACS URL.<br>For instance: `^https:\/\/yourcollabserver\.com\/services\/saml\/acs\/$` |
| **Single Logout URL** | The URL of endpoint where the logout request will be sent to.<br>Collaborator servers have the following logout endpoint: `https://yourcollabserver.com/services/saml/logout` |

6. Switch to the **Parameters** tab. Append assertion parameters with the following with names: **FirstName, LastName** and **Role** and enable the *Include in SAML assertion* option for them. These parameters will respectively contain the users first name, last name, and their role (regular user or administrator).

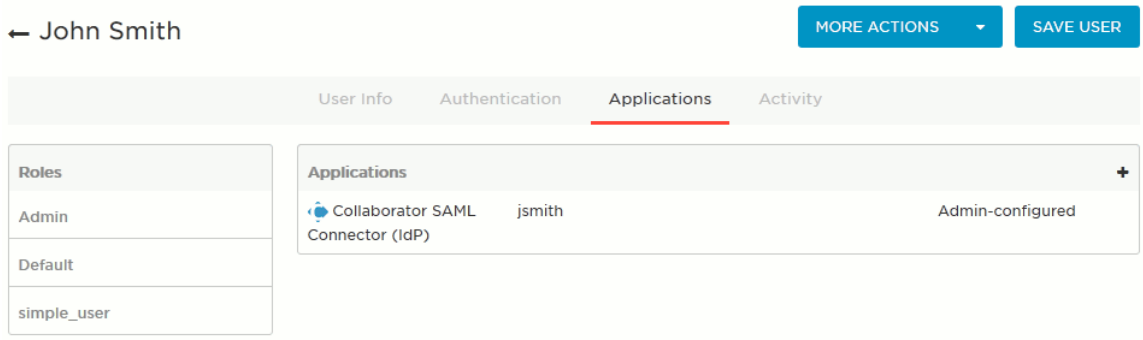



⚠ Please ensure, that parameters are named exactly as **FirstName, LastName**, and **Role** (including letter-case), since Collaborator would expect these parameter names in the assertion response. These parameters are not strictly required. If **FirstName** and **LastName** parameters are missing, user account would obtain blank values for first name, last name and display name. If **Role** parameter is missing, or its value is not "admin" (case-sensitive), user account would obtain regular user role (even if this account had administrator role previously).

Optionally, modify the **NameID** parameter to pass username rather than e-mail address as primary user identifier.
Optionally, add the **MemberOf** parameter (or other dedicated parameter) that will contain information about user membership in groups. See "Synching User Group Membership with SAML Single Sign-On Server 142" below.
Alternatively, you can use the **NameID** parameter or add one more parameter to store LDAP user name for synchronizing group membership via LDAP or Active Directory. See "Synching User Group Membership with LDAP/Active Directory 143" below.

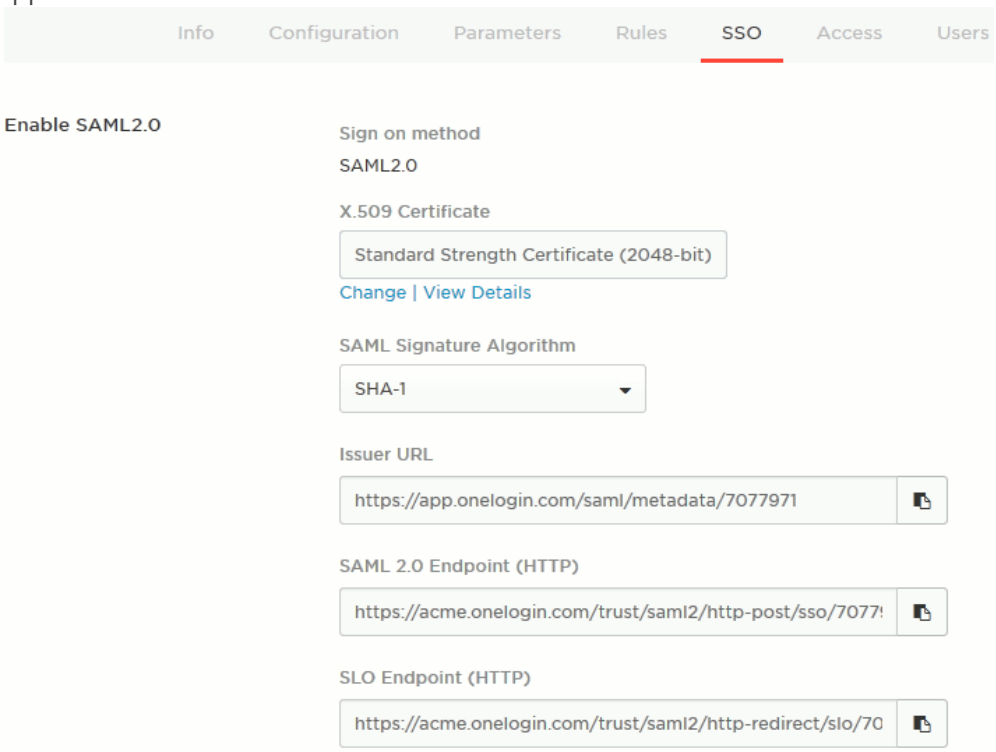
7. Click **Save** to confirm application configuration changes.

8. Assign users to the created Collaborator SAML Connector application: Select **Users | All Users** from the main menu, click the desired user, switch to the **Applications** tab and click the plus sign to assign an app to the user.




⚠ Make sure that you have created at least one SSO account having admin privileges (`Role = admin`). If the **User Role sync** option is enabled, Collaborator will synchronize user roles between SAML and Collaborator servers.Users with admin privileges on the SSO server will automatically become administrators on Collaborator server. SSO accounts having any other value of the `Role` parameter will become regular users on Collaborator server.
Collaborator is expecting SSO server response with only one attribute for the "Role" name. If server response contains multiple attributes with the "Role" name, Collaborator fetches the value of the last attribute.

9. Select **Apps | Company Apps** from the main menu, click Collaborator SAML Connector application and switch to the **SSO** tab.



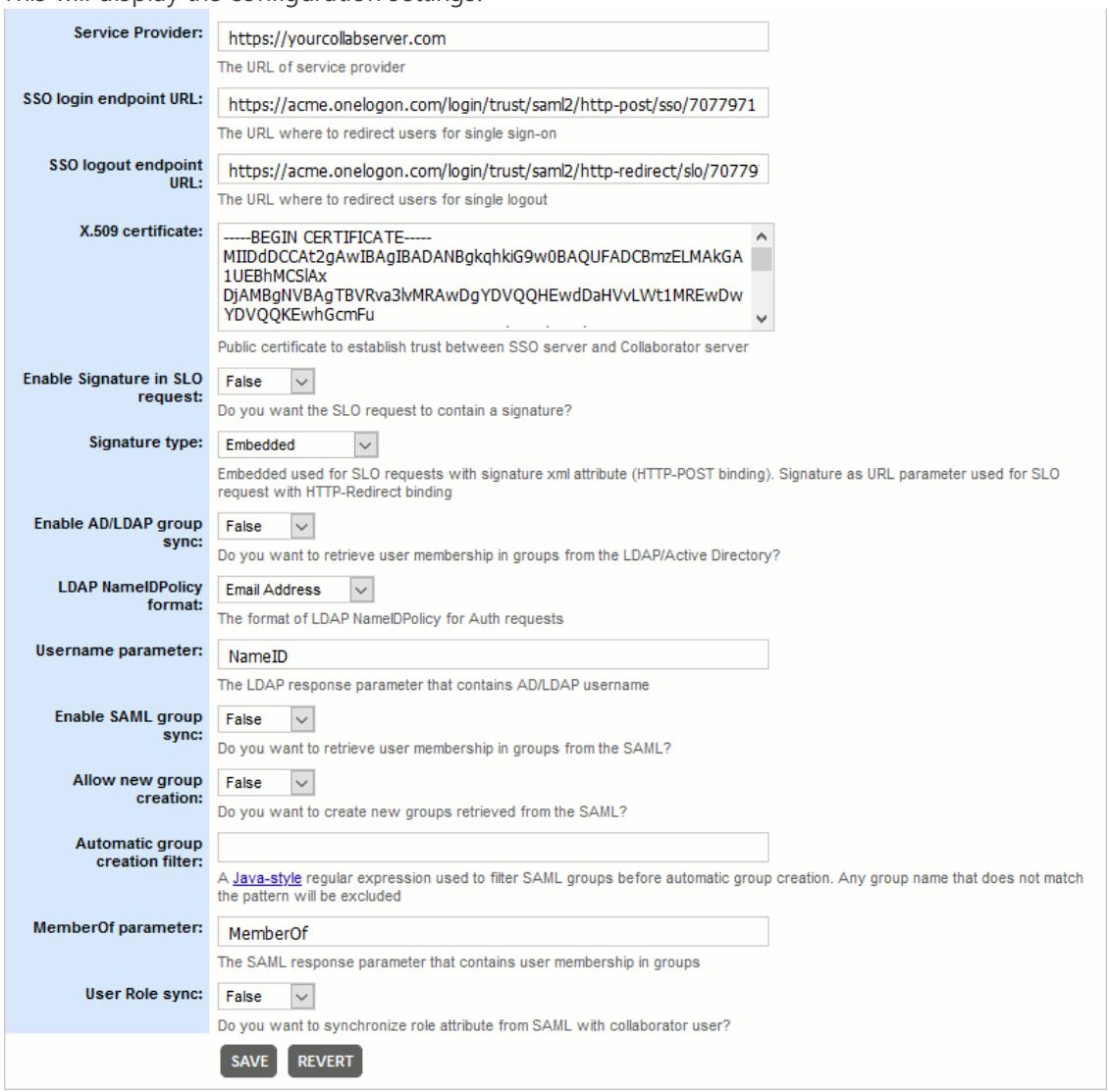

This tab holds identity provider SAML metadata that you must provide to your Collaborator server to complete the integration.

## Configure Collaborator Server

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.

2. In Collaborator, go to **Admin > Single Sign-On**

3. In the **New SSO Configuration** section select **SAML SSO** configuration type and click **Create**. This will display the configuration settings.




4. Specify the setting values:

| | |
|---|---|
| **Service Provider** | The URL of the entity that is expected to receive the SAML message. In our case, this will be the URL of your Collaborator server.<br>For instance: https://yourcollabserver.com<br><br>Important. Your Collaborator server must be accessible to SSO server. Configure a firewall or enable tunneled connections to do so. |
| **SSO login endpoint URL** | The URL where to redirect users for single sign-on procedure, if a session is not already established. |

This should be the value from the SAML 2.0 Endpoint (HTTP) setting of Collaborator SAML Connector application on SSO server.

**SSO logout endpoint URL**

The URL where to redirect users for single logout procedure.

This should be the value from the SLO Endpoint (HTTP) setting of Collaborator SAML Connector application on SSO server.

**X.509 Certificate**

The public certificate that establishes trust between SSO server and Collaborator server.

This should be the value from the X.509 Certificate setting of Collaborator SAML Connector application on SSO server.

To copy the X.509 certificate, click View Details and then click the Copy to Clipboard icon.

**Enable signature in SLO request**

Specifies whether to append digital signature to single logout requests.

**Signature type**

Specifies which type of signature to use. For HTTP-POST binding single logout requests use embedded xml attribute signature. For HTTP-Redirect binding single logout requests use signature as URL parameter.

**Enable AD/ LDAP group sync**

Specifies whether Collaborator should synchronize its native groups with group information from the LDAP/Active Directory. See "Synching User Group Membership with LDAP/Active Directory 143" below.

**SAML NameIDPolicy Format**

Specifies the format of the `NameIDPolicy` parameter in the authentication requests.

**Username parameter**

Defines the name of SAML response parameter that contains LDAP username. See "Synching User Group Membership with LDAP/Active Directory 143" below.

If the parameter name is not specified, or the parameter returns empty value the synchronization is not performed.

**Enable SAML group sync**

Specifies whether Collaborator should synchronize its native groups with group information from the SAML single sign-on server.

Once enabled, Collaborator will check user membership in groups in the SSO server and automatically add this user to the corresponding groups on the Collaborator server. If a group with the specified name does not exist, Collaborator will act according to the "Allow new group creation" and "Automatic group creation filter" settings below.

| | |
|---|---|
| | See Synching User Group Membership with SAML Single Sign-On Server 142 for details. |
| **Allow new group creation** | Specifies whether to create new groups on the Collaborator server when a group with the specified name does not exist. If enabled, Collaborator will create new group, if disabled Collaborator will only synchronize membership of existing groups. |
| **Automatic group creation filter** | A Java-style regular expression used to filter SSO groups before automatic group creation. Any group name that does not match the pattern will be excluded. |
| **MemberOf parameter** | Defines the name of SAML response parameter that contains information about user membership in groups. |
| **User Role sync** | Specifies whether to synchronize user roles between SAML and Collaborator servers, when synching user membership with SAML SSO server 142. |

5. After you specified the values, click **Save**. This will create a SAML SSO configuration.

6. If you have multiple configurations, press **Make Active** to mark current configuration as active.

7. Scroll the **Admin > Single Sign-On** screen up to the **Single Sign-On (SSO) Status** section and change the **Enable Single Sign-On** setting to **Yes**.

Now the integration between Collaborator and single sign-on server is configured and running. Further attempts to log into Collaborator server will be processed by the sign-on server.

## SAML Metadata

SAML metadata is an XML file describing SAML entities in a standardized way. Identity provider and service provider could share their metadata to establish a baseline of trust and interoperability.

When SAML SSO integration is enabled, Collaborator publishes its service provider metadata at the following endpoint: `https://yourcollabserver.com/services/saml/metadata`

Sample service provider metadata

```
<?xml version="1.0"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" validUntil="2020-03-
28T13:34:57Z" entityID="https://yourcollabserver.com"><md:
SPSSODescriptor AuthnRequestsSigned="false"
WantAssertionsSigned="false" protocolSupportEnumeration="urn:oasis:
names:tc:SAML:2.0:protocol"><md:KeyDescriptor use="signing"><ds:
KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:
X509Data><ds:X509Certificate>-----BEGIN CERTIFICATE-----
MIIDdDCCAt2gAwIBAgIBADANBgkqhkiG9w0BAQUFADCBmzELMAkGA1UEBhMCSlAxDjAMB
gNVBAgTBVRva3lvMRAwDgYDVQQHEwdDaHVvLWt1MREwDwYDVQQKEwhGcmFuazRERDEYMB
YGA1UECxMPV2ViQ2VydCBTdXBwb3J0MRgwFgYDVQQDEw9GcmFuazRERCBXZWIgQ0ExIzA
hBgkqhkiG9w0BCQEWFHN1cHBvcnRAZnJhbms0ZGQuY29tMCIYDzE3NjkwODE1MTU0NjQx
WhgPMTgyMTA1MDUxNjUzMjFaMIHbMQswCQYDVQQGEwJGUjEXMBUGA1UECBQOw45sZS1kZ
S1GcmFuY2UxDjAMBgNVBAcTBVBhcmlzMRwwGgYDVQQKExNIZXJlZGl0YXJ5IE1vbmFyY2
h5MRYwFAYDVQQLEw1IZWFkIG9mIFN0YXRlMSkwJwYJKoZIhvcNAQkBFhpuYXBwaUBncmV
hdGZyZW5jaGVtcGlyZS5mcjEbMBkGA1UEAxMSRW1wZXJvciBOYXBvbGVvbiBJMRIwEAYD
VQQEEwlCb25hcGFydGUxETAPBgNVBCoTCE5hcG9sZW9uMIIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIBCgKCAQEAvdcwMpsM3EgNmeO/
fvcuZsirnBZCh0yJlySGmev792ohOBvfC+v27ZKcXN+H7V+xSxEDDAFq9SCEk4MZ8h/
ggtw63T6SEYeNGtnfyLv8atienun/6ocDp0
+26xvj8NxmaKL4MQM9j9aYgt2EOxUTH5kBc7mc621q2RJi0q/y0/
SdX2Pp/3MKDirOs81vfc2icEaYAisd5IOF9vpMpLr3b3Qg9T66/4hQS6DgIOkfUurWqe3
3sA2RRv7ql1gcxL1ImBxBtYQGsujn8fCNRK5jtMtICkEi9tks/
tYzSaqgby3QGfbA18xl7FLLjnZDLVX3ZVchhveR78/f7U/
xh8C2WQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFmEiU2vn10fXvL+nJdRHJsf0P+f6v8H
+vbkomog4gVbagDuFACJfAdKJhnc/
gzkCF1fyeowOD68k4e0H1vyLuk23BUmjW41nOjdg8LrTAS8fMwkj5FVSKR2mHciHWgY/
BU4UypYJtcgajH1bsqwUI50wfbggW4VzLD842q5LhnW-----END CERTIFICATE-----
</ds:X509Certificate></ds:X509Data></ds:KeyInfo></md:
KeyDescriptor><md:KeyDescriptor use="encryption"><ds:KeyInfo xmlns:
ds="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data><ds:
X509Certificate>-----BEGIN CERTIFICATE-----
MIIDdDCCAt2gAwIBAgIBADANBgkqhkiG9w0BAQUFADCBmzELMAkGA1UEBhMCSlAxDjAMB
gNVBAgTBVRva3lvMRAwDgYDVQQHEwdDaHVvLWt1MREwDwYDVQQKEwhGcmFuazRERDEYMB
YGA1UECxMPV2ViQ2VydCBTdXBwb3J0MRgwFgYDVQQDEw9GcmFuazRERCBXZWIgQ0ExIzA
hBgkqhkiG9w0BCQEWFHN1cHBvcnRAZnJhbms0ZGQuY29tMCIYDzE3NjkwODE1MTU0NjQx
WhgPMTgyMTA1MDUxNjUzMjFaMIHbMQswCQYDVQQGEwJGUjEXMBUGA1UECBQOw45sZS1kZ
S1GcmFuY2UxDjAMBgNVBAcTBVBhcmlzMRwwGgYDVQQKExNIZXJlZGl0YXJ5IE1vbmFyY2
h5MRYwFAYDVQQLEw1IZWFkIG9mIFN0YXRlMSkwJwYJKoZIhvcNAQkBFhpuYXBwaUBncmV
hdGZyZW5jaGVtcGlyZS5mcjEbMBkGA1UEAxMSRW1wZXJvciBOYXBvbGVvbiBJMRIwEAYD
VQQEEwlCb25hcGFydGUxETAPBgNVBCoTCE5hcG9sZW9uMIIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIBCgKCAQEAvdcwMpsM3EgNmeO/
fvcuZsirnBZCh0yJlySGmev792ohOBvfC+v27ZKcXN+H7V+xSxEDDAFq9SCEk4MZ8h/
ggtw63T6SEYeNGtnfyLv8atienun/6ocDp0
+26xvj8NxmaKL4MQM9j9aYgt2EOxUTH5kBc7mc621q2RJi0q/y0/
SdX2Pp/3MKDirOs81vfc2icEaYAisd5IOF9vpMpLr3b3Qg9T66/4hQS6DgIOkfUurWqe3
```

```
3sA2RRv7ql1gcxL1ImBxBtYQGsujn8fCNRK5jtMtICkEi9tks/
tYzSaqgby3QGfbA18xl7FLLjnZDLVX3ZVchhveR78/f7U/
xh8C2WQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFmEiU2vn10fXvL+nJdRHJsf0P+f6v8H
+vbkomog4gVbagDuFACJfAdKJhnc/
gzkCF1fyeowOD68k4e0H1vyLuk23BUmjW41nOjdg8LrTAS8fMwkj5FVSKR2mHciHWgY/
BU4UypYJtcgajH1bsqwUI50wfbggW4VzLD842q5LhnW-----END CERTIFICATE-----
</ds:X509Certificate></ds:X509Data></ds:KeyInfo></md:
KeyDescriptor><md:SingleLogoutService Binding="urn:oasis:names:tc:
SAML:2.0:bindings:HTTP-Redirect" Location="https://acme.onelogon.com/
login/trust/saml2/http-redirect/slo/7077971"/><md:NameIDFormat>urn:
oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:
NameIDFormat><md:AssertionConsumerService Binding="urn:oasis:names:
tc:SAML:2.0:bindings:HTTP-POST" Location="https://acme.onelogon.com/
login/trust/saml2/http-post/sso/7077971" index="1"/></md:
SPSSODescriptor></md:EntityDescriptor>
```

## Synching User Group Membership with SAML Single Sign-On Server

You may configure user and group synchronization between Collaborator and your SAML single sign-on server. In this case, Collaborator will retrieve user properties (email, first and last name) and their membership in groups from the single sign-on server when the users login.

The login procedure will look like as follows:

1. When user logs in, sign-on server sends SAML response with some pre-configured parameter which contains information about user membership in groups. This could be any existing parameter, like `memberOf`, or some other dedicated parameter.

2. Collaborator receives the response and extracts the group names from the that parameter.

3. Collaborator checks if some of existing user groups has matching name. Once such group is found, it adds the user to this group. Otherwise, it checks if the "Allow new group creation" setting is enabled and if the group name matches the "Automatic group creation filter", creates new group on success and adds the user to the new group.

4. Collaborator server logs the user in.

### Technical details of SAML synchronization

Collaborator retrieves user membership from the SAML single sign-on server and checks if some of existing user groups 226 has matching name. Once such group is found, it adds the user to this group. Otherwise, it checks if the "Allow new group creation" setting is enabled and if the group name matches the "Automatic group creation filter", creates new group on success and adds the user to the new group.

Along with membership information, Collaborator is able to sync user permissions with SAML SSO server. If the **User Role sync** option is enabled, it will synchronize user roles between SAML and Collaborator servers. SSO accounts having admin privileges (Role = admin) will automatically become administrators on Collaborator server. SSO accounts having empty or any other value of the `Role` parameter will become regular users on Collaborator server.

Collaborator actualizes user properties, their membership in groups and permissions on every login. Such algorithm allows to keep a consistency between Collaborator and SSO server.

When SAML SSO integration is disabled or deleted, synchronized groups will become disabled.

## Synching User Group Membership with LDAP/Active Directory

Alternatively, you may configure user membership synchronization between Collaborator, SAML single sign-on server and LDAP directory or Active Directory. In this case, Collaborator will use single sign-on server to manage users, but will query user membership from the LDAP directory or Active Directory.

The login procedure will look like as follows:

1. When user logs in, sign-on server sends SAML response with some pre-configured parameters.
2. Collaborator receives the response, extracts the parameter which holds LDAP username, and sends this username to LDAP directory or Active Directory.
3. LDAP directory or Active Directory searches for groups this user belongs to and returns them to Collaborator.
4. Collaborator checks if the specified user and group(s) exist, creates them if needed, and adds the user to the group(s).
5. Collaborator server logs the user in.

### Configuration for LDAP/AD synchronization

In order to enable synchronization of group membership via LDAP or Active Directory, you need to perform the following actions:

1. Select/Create an SAML assertion parameter in user info section that will to store user's username. This could be any existing parameter, like NameID, FirstName or LastName, or some dedicated parameter.

2. Modify Collaborator server's Root.xml (`<collab server install dir>/tomcat/conf/Catalina/localhost/ROOT.xml)` to use the Combined Realm element:

```
<Realm className="org.apache.catalina.realm.CombinedRealm" >
```

```
 <Realm allRolesMode="strictAuthOnly" className="org.apache.catalina.
realm.DataSourceRealm" dataSourceName="/jdbc/collabserver"
localDataSource="true" roleNameCol="user_admin"
userCredCol="user_password" userNameCol="user_login"
userRoleTable="user" userTable="user">
       <CredentialHandler className="org.apache.catalina.realm.
MessageDigestCredentialHandler" algorithm="md5" />
 </Realm>

 <Realm allRolesMode="strictAuthOnly" className="org.apache.catalina.
realm.JNDIRealm"
       connectionURL="ldap://ldap.example.com:389"
       userPattern="uid={0},dc=example,dc=com"
       roleBase="dc=example,dc=com"
       roleSearch="(&amp;(objectClass=groupOfUniqueNames)
(uniqueMember={0}))"
       connectionName="cn=read-only-admin,dc=example,dc=com"
       connectionPassword="password"
 />
</Realm>
```

The `roleSearch` parameter above defines the pattern used to identify user membership in a group. It uses standard LDAP query format syntax.

3. Verify that the `collaborator-authentication` parameter remains `true` as for regular authentication:

```
<Parameter description="Is the Collaborator database used for
authentication?" name="collaborator-authentication" override="false"
value="true"/>
```

4. In **Admin > Single Sign-On** screen locate the desired SAML SSO configuration, set the **Enable AD/LDAP group sync** setting to "Yes" and in **Username parameter** field specify the name of SAML assertion parameter that contains the LDAP username.

### Technical details of LDAP/AD synchronization

For mapping user membership in groups Collaborator uses the group's fully qualified domain name (FQDN) retrieved from the LDAP or AD. It checks if some of existing user groups[226] has matching FQDN and adds the user to this group on success. Otherwise, it creates a new group and adds the user to the new group.

To name the new group Collaborator uses the first entry of group's fully qualified domain name (FQDN). For example, a group having the "ou=`ccusers`,dc=example,dc=com" FQDN will have the `ccusers` title. If some other group already has the same title, Collaborator will append the ordinal number to the group title: `ccusers1, ccusers2` and so on.

On every login Collaborator checks existing groups created via LDAP or Active Directory synchronization and actualizes user membership in those groups. Such algorithm allows to keep a consistency between Collaborator and LDAP or Active Directory.

## Troubleshooting

If you experience troubles with single sign-on authentication and cannot login to your Collaborator server (wrong redirect URLs, cannot login as admin and so on), you can disable SAML single sign-on authentication via the `-Dcom.smartbear.server.sso.disable=true` Java VM option[1232].

## Known Issues With OneLogin Server

When users log out from OneLogin server, they still remain logged in Collaborator server. For some reason OneLogin server does not trigger Collaborator's SLO endpoint when user logs out directly from OneLogin server. However, when users log out from Collaborator server, they will be logged out both from Collaborator server and from OneLogin server.

#### 3.2.8.2 Configuring SSO via Crowd OpenID

This topic describes how to establish single sign-on between an Atlassian Crowd OpenID server and Collaborator. To learn general principles of how single sign-on operates, see Single Sign-On[130].

## (Optional) Enable HTTPS Connections

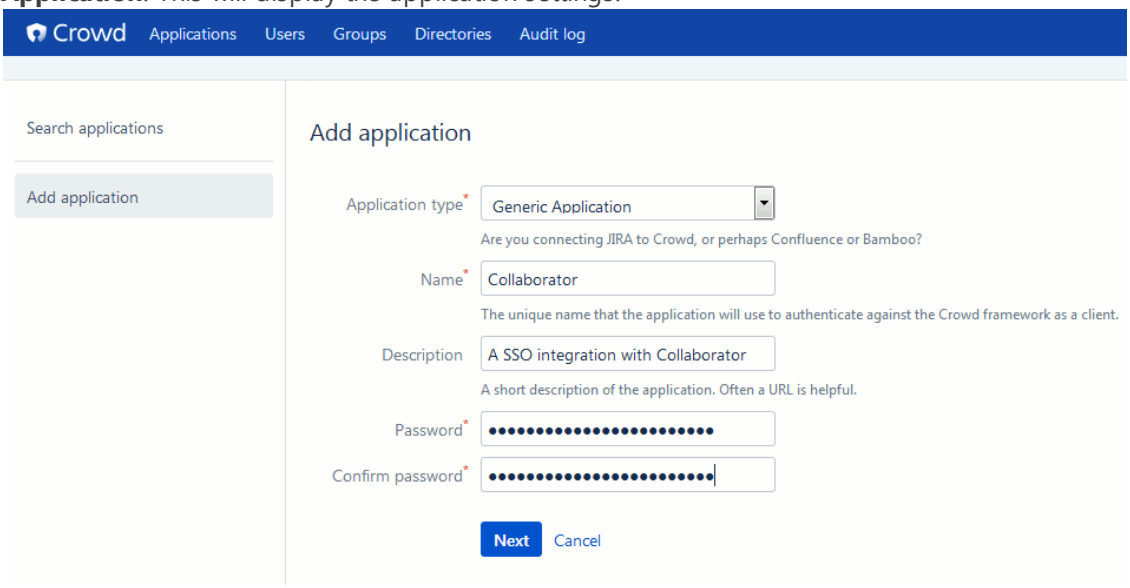
Single sign-on servers mostly use HTTPS connections, so you will likely need to enable it for Collaborator server as well. For instructions, please see, Configuring HTTPS 111. Do not forget to restart the Collaborator server to apply changes.

## Install and Configure Crowd Server

1. Install Atlassian Crowd server.

2. Log in to your Crowd server as an administrator.

3. Add new application integration: Select **Applications** from the main menu and then click **Add Application**. This will display the application settings.

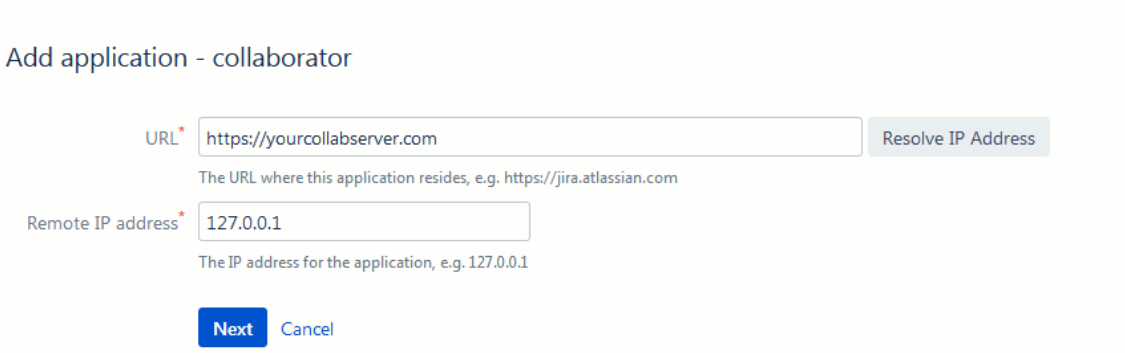



4. Specify the setting values:

| | |
|---|---|
| **Application type** | The type of application you are adding to Crowd. For Collaborator integration select the **Generic Application** option. |
| **Name** | The name of the client application. Must be unique across the server. |
| **Description** | A short description of the application. |
| **Password** and **Confirm password** | A password application will use when it authenticates against the Crowd server. Retype the password to confirm it. |

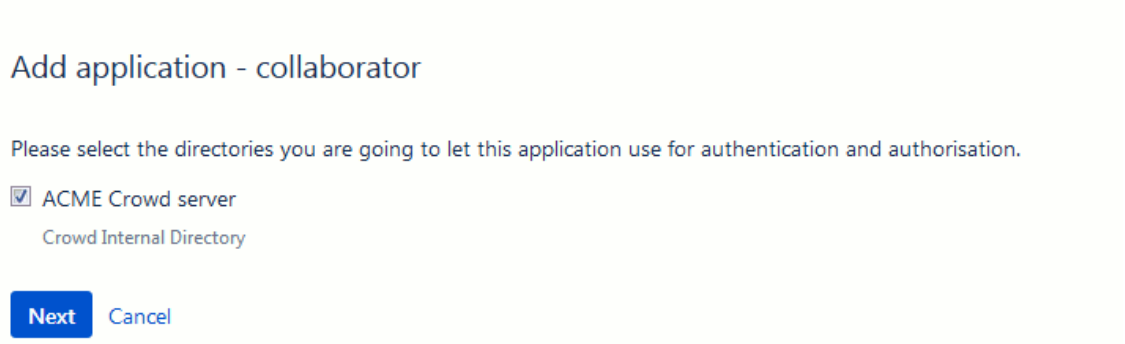
Click **Next**

5. Specify the URL and the remote IP address of your Collaborator server. Specify port number, if you are not using a proxy.
   You can click **Resolve IP Address**, which prompts Crowd to resolve the IP address for your application.
   **Important.** Your Collaborator server must be accessible to SSO server and vice versa. Configure a firewall or enable tunneled connections to do so.



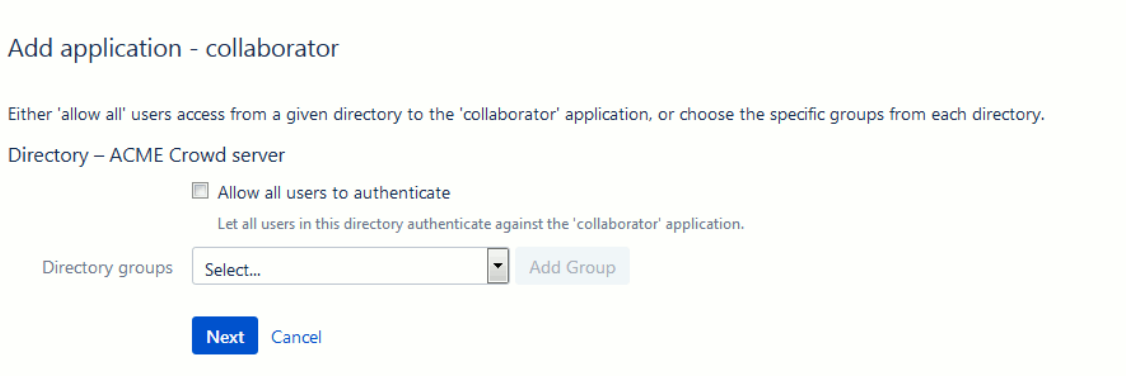

Click **Next**

6. Select one or more directories that this application can use for authentication and authorization:




Click **Next**

7. Specify the users who are authorized to access the application. Do one one of the following: either select **Allow all users to authenticate**, to grant application access to all users defined in the directory, or select one or more groups you wish to have access, and click **Add Group**.




Click **Next**

8. Confirm the details for your application and click **Add Application**.

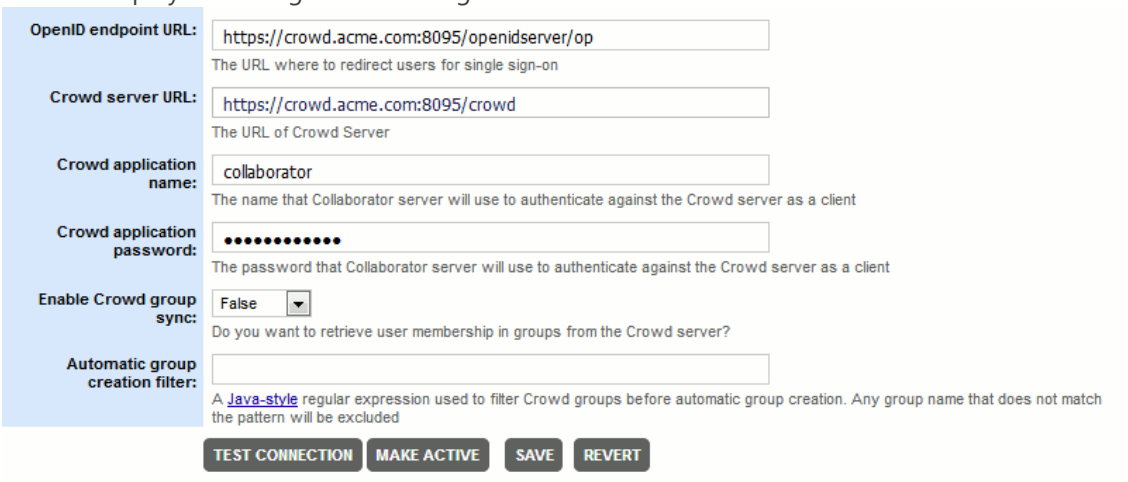
9. If you use proxy server, switch to the **Remote addresses** tab of application settings and add the proxy server's IP to the list.

## Configure Collaborator Server

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.

2. In Collaborator, go to **Admin > Single Sign-On**

3. In the **New SSO Configuration** section select **Crowd SSO** configuration type and click **Create**. This will display the configuration settings.




4. Specify the setting values:

| | |
|---|---|
| **OpenID endpoint URL** | The URL where to redirect users for single sign-on procedure, if a session is not already established.<br><br>The endpoint typically has the following format: *http(s)://yourcrowdserver:8095/openidserver/op* |
| **Crowd server URL** | The URL of your Crowd server (including port number).<br><br>The Crowd server URL typically has the following format: *http(s)://yourcrowdserver:8095/crowd* |
| **Crowd application name**and **Crowd application password** | The name and password of Crowd application for Collaborator integration.<br><br>This is the application that we have created on the Crowd server earlier 146. |
| **Enable Crowd group sync** | Specifies whether Collaborator should synchronize its native groups with group information from the Crowd server.<br><br>Once enabled, on every logging-in, Collaborator will check user membership in groups on the Crowd server, create new groups (if needed), and automatically add this user to the corresponding groups on the Collaborator server. This operation could be time-consuming, if there are multiple groups on your Crowd server. If you have any problems, check the server log. |
| **Automatic group creation filter** | A Java-style regular expression used to filter Crowd groups before automatic group creation. Any group name that does not match the pattern will be excluded. |

5. After you specified the values, click **Save**. This will create a Crowd OpenID configuration.
6. If you have multiple configurations, press **Make Active** to mark current configuration as active.
7. Scroll the **Admin > Single Sign-On** screen up to the **Single Sign-On (SSO) Status** section and change the **Enable Single Sign-On** setting to **Yes**.

Now the integration between Collaborator and Crowd OpenID server is configured and running. Further attempts to log into Collaborator server will be processed by the sign-on server.

## Troubleshooting

If you experience troubles with Crowd OpenID single sign-on authentication, please check the following:

- Ensure that Crowd home directory is specified in `{CROWD_INSTALL}/crowd-webapp/WEB-INF/classes/crowd-init.properties` file. Check that this directory is writable.
- Check that CrowdID add-on is installed and configured. By default, it is installed and configured during the installation of Crowd server using the Crowd Setup Wizard. If you have skipped those steps, then install and configure it as described in Atlassian Documentation.
- If all user attempts to log-in to Collaborator server are always redirected to `http(s)://yourcrowdserver:8095/openidserver/` (even when they are already logged-in on Crowd server), please verify that the **OpenID endpoint URL** 149 value is specified correctly. Namely, that the URL ends with `/openidserver/op`

If the problem persists, you can disable single sign-on authentication via the `-Dcom.smartbear.server.sso.disable=true` Java VM option 1232.

#### 3.2.8.3 Configuring SSO via Java Servlet

Single sign-on integration using Java servlet for Tomcat is deprecated and will be removed in future releases. Please consider implementing single sign-on using SAML 132 or Atlassian Crowd OpenID 145 instead.

This topic describes how to establish single sign-on between an SSO server and Collaborator using Java servlet for Tomcat. To learn general principles of how single sign-on operates, see Single Sign-On 130.

Below we describe how to establish single sign-on between Collaborator and Apereo Central Authentication Service (CAS). Integration with Single Sign-On servers of other vendors is performed in a similar manner.

## Enabling HTTPS Connections

Most Single Sign-On servers use HTTPS connections, thus you will likely need to enable it for Collaborator server as well. For instructions, please see, Configuring HTTPS 111. Do not forget to restart the Collaborator server to apply changes.

## Installing SSO Server and Client Components

In Apereo Central Authentication Service, SSO server component is implemented as a Java servlet, and SSO clients are provided for different platforms and technologies (Java, .NET, PHP, Python and so on). Since Collaborator is a Java application, we will use Java CAS client.

**Notes:**

To simplify the example, we will install both CAS server and CAS client on the same instance of Tomcat server.

Not all versions of CAS server and CAS client are compatible with each other. For example, CAS server 4.0 does not work with CAS client 3.3. In this integration, will use CAS server 3.5.2 and CAS client 3.1.12.

1. Download CAS server archive and unzip it to temporary folder.
2. Copy `modules\cas-server-webapp-3.5.2.war` file to `<collab server install dir>\tomcat\webapps` folder.
3. Download CAS client archive and unzip it to another temporary folder.
4. Copy all files from `cas-client-3.1.12\modules\` folder to `<collab server install dir>\tomcat\webapps\ROOT\WEB-INF\lib` folder.

## Adding Filters to Collaborator's Tomcat Server

In this step we will add several filters to Tomcat server. These filters would detect unauthenticated users, redirect them to SSO server, validate users and perform single sign-out. To learn more about filter configuration, see "The Essentials of Filters" on Oracle's website.

Open `<collab server install dir>\tomcat\webapps\ROOT\WEB-INF\web.xml` file and append the following lines to it:

```
<filter>
 <filter-name>CAS Authentication Filter</filter-name>
 <filter-class>org.jasig.cas.client.authentication.
AuthenticationFilter</filter-class>
 <init-param>
 <param-name>casServerLoginUrl</param-name>
```

```
 <param-value>https://yourcollabserver:8443/cas-server-webapp-3.5.2/
login</param-value>
 </init-param>
 <init-param>
 <param-name>serverName</param-name>
 <param-value>https://yourcollabserver:8443</param-value>
 </init-param>
 <init-param>
 <param-name>gateway</param-name>
 <param-value>false</param-value>
 </init-param>

 </filter>
 <filter-mapping>
 <filter-name>CAS Authentication Filter</filter-name>
 <url-pattern>/ui/*</url-pattern>
 <url-pattern>/go/*</url-pattern>
 </filter-mapping>

 <filter>
 <filter-name>CAS Validation Filter</filter-name>
 <filter-class>org.jasig.cas.client.validation.
Cas20ProxyReceivingTicketValidationFilter</filter-class>
 <init-param>
 <param-name>casServerUrlPrefix</param-name>
 <param-value>https://yourcollabserver:8443/cas-server-webapp-3.5.2/
</param-value>
 </init-param>
 <init-param>
 <param-name>serverName</param-name>
 <param-value>https://yourcollabserver:8443</param-value>
```

```
 </init-param>

 </filter>
 <filter-mapping>
 <filter-name>CAS Validation Filter</filter-name>
 <url-pattern>/*</url-pattern>
 </filter-mapping>

 <filter>
 <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
 <filter-class>org.jasig.cas.client.util.
HttpServletRequestWrapperFilter</filter-class>
 </filter>
 <filter-mapping>
 <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
 <url-pattern>/*</url-pattern>
 </filter-mapping>

 <filter>
 <filter-name>CAS Single Sign Out Filter</filter-name>
 <filter-class>org.jasig.cas.client.session.SingleSignOutFilter
 </filter-class>
 </filter>
 <filter-mapping>
 <filter-name>CAS Single Sign Out Filter</filter-name>
 <url-pattern>/*</url-pattern>
 </filter-mapping>
 <listener>
 <listener-class>org.jasig.cas.client.session.
SingleSignOutHttpSessionListener</listener-class>
 </listener>
```

Remember to change "yourcollabserver" with the actual URL of your Collaborator server

## Configuring Collaborator Server VM Options

In this step we need to change server's VM options to enable SSL connections between the Collaborator and CAS servers and to specify the logout redirect URL.

Open <collab server install dir>/ccollab-server.vmoptions file and add the following lines to it:

```
-Djavax.net.ssl.keyStore=<collab server install dir>/tomcat/conf/collab.ks
-Djavax.net.ssl.keyStorePassword=<the keystore password>
-Djavax.net.ssl.trustStore=<collab server install dir>/tomcat/conf/cacerts
-Djavax.net.ssl.trustStorePassword=<the truststore password>
-Dsmartbear.ccollab.sso.logout.redirect.url=https://yourcollabserver
:8443/cas-server-webapp-3.5.2/logout
```

The latter VM option specifies the URL to which users will be redirected when they press "Logout" link in the Collaborator's web-interface.

Alternatively, you can remove the "Logout" link in Collaborator's Web Client by setting the VM option as:

```
-Dsmartbear.ccollab.sso.logout.redirect.url=hide
```

In that case, users would manually perform logging out from the SSO server.

Restart the Collaborator server to apply changes in VM options.

## Configuring CAS Properties for Logout Requests

In this step we will configure how CAS server should handle logout requests.

1) Open Collaborator Web Client 312.

2) Open the <collab server install dir>\tomcat\webapps\cas-server-webapp-3.5.2\WEB-INF\ folder and wait until a file with name cas.properties is created in this folder. (The cas.properties file is deployed from the cas-server-webapp-3.5.2.war web application archive, so it can take some time til it is created.)

3) Open <collab server install dir>\tomcat\webapps\cas-server-webapp-3.5.2\WEB-INF\cas.properties file and set the value of cas.logout.followServiceRedirects property to true:

```
 # Specify whether CAS should redirect to the specified service
parameter on /logout requests
cas.logout.followServiceRedirects=true
```

## Restarting the Collaborator Server

In order to apply all the changes we need to restart the Collaborator server.

## Known Issues With CAS Server

- Currently, CAS server settings only allow to have coinciding login/password values. That is, the password must have the same value as the login.
- Because of cookie handling issues, CAS server's Single Logout functionality may not work correctly with Collaborator web clients. Namely, logging out from Collaborator web client will not terminate a session on a CAS server. As a result, the subsequent attempts to open any of Collaborator web client's /ui pages will not be redirected to CAS server login page but will open the Collaborator's native login page instead. This issue does not affect /go pages of web client, however major part of Collaborator web client pages are /ui pages and are affected. To workaround the problem:

  1) Open `<collab server install dir>\tomcat\webapps\cas-server-webapp-3.5.2\WEB-INF\spring-configuration\ticketGrantingTicketCookieGenerator.xml` file and set the value of `p:cookieSecure` property to false:

```
<bean id="ticketGrantingTicketCookieGenerator" class="org.jasig.cas.
web.support.CookieRetrievingCookieGenerator"
p:cookieSecure="false"
p:cookieMaxAge="-1"
p:cookieName="CASTGC"
p:cookiePath="/cas" />
```

  2) Logout from Collaborator Web Client and logout from CAS server.

  3) Clear browser cookies.

  Alternatively, you can disable the "Logout" link in Collaborator Web Client and perform logout from the CAS server. See above 154 for instructions.

### 3.2.9 JMX Monitoring

You can monitor many aspects of the Collaborator server via JMX, Java's standard, built-in network monitoring system.

## Setting up a basic JMX system

To enable JMX you must add Java properties to the server startup script. To do that, edit the file `ccollab-server.vmoptions` which is found inside the server installation directory.

Inside this file, add the following lines:

```
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=54321
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

The first line enables JMX. The second line sets the TCP/IP port number for the JMX protocol, and of course you can use a value other than the example `54321`. The last two lines disable JMX authentication and security.

> **Note:** You must restart the Collaborator server before these settings take effect.

You can now use any JMX console software to access the Collaborator JMX variables. Sun's Java installation ships with a simple JMX console which can be found in `[Java-Install-Dir]/bin/jconsole.exe`. If you use Sun's console, select the "Remote" tab in the connection dialog and specify the host name, the port given in the configuration above. Leave "User Name" and "Password" blank.

## Supported JMX data

There are various things you can monitor with JMX, and this manual cannot cover them all.

The data specific to Collaborator can be found under "MBeans," then under the root tree node "`com.smartbear.ccollab`". Under there you will find one child tree node for each installation of Collaborator, named by the web page it is installed under. The default is just a simple forward slash. Under that node is various data.

Of particular note is "`ApplicationStateCache`" and "`ServletState`". This contains basic information about logins, licensing, versioning, and product usage.

## More complex JMX server configurations

A full description of how to configure JMX is outside the scope of this manual. It is completely standardized and documented on Sun's web site. A good resource to get you started is this:

http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html

### 3.2.10 Technical Specifications

This section describes the technical aspects of the Collaborator server. This information is not generally needed for server administration.

## Server Technology

The web server is Apache Tomcat 9.0.34 Collaborator itself is a collection of standard J2EE servlets, packaged in the standard WAR format that can be used inside any J2EE servlet container. Tomcat is the only servlet container specifically supported by SmartBear.

Tomcat is a sophisticated, scalable stand-alone web server with support for HTTPS 111, LDAP authentication 119, advanced HTTP protocol options, database pools, load-balancing, and more.

For more information about how Collaborator is configured under Tomcat, see Tomcat Configuration Reference.

## Starting and Stopping the Server

### Windows

The server is a Windows Service. This means you can start, stop and restart the server just like any other service. The typical way is through the "Services" Control Panel. You can also use "`net start ccollab-server`" and "`net stop ccollab-server`" from the command-line.

### Unix

The server is a shell script in the installation directory called `ccollab-server`. You can use the usual "`start`" and "`stop`" directives, which also means the script can be used directly in an `init.d` system.

If you encounter the error "Error 1: Incorrect function" when starting the server as a service on Windows, or the server fails to start with no error, possible remedies are:

- free up memory for the service by removing `-Xmx` or `-Xms` in the `ccollab-server.vmoptions` file.

- confirm a valid Java version/environment by running '`java -version`' in you Java bin directory.

## Data Storage

Collaborator stores almost all data in the database[59] initially set up for it. It also uses file content storage[97] – a local folder that keeps copies of uploaded files.

## Recommended Hardware

For trials, you can use any hardware available. We have lots of customers who have done pilots on "servers" such as laptops and regular workstations. For small groups (30 users or less) doing small reviews, the server is not demanding on CPU, memory, disk access, or database access; if you run the server on a workstation, you will never know it is there.

For permanent installations, and to support hundreds or thousands of users, please review our hardware recommendations in the System Requirements[57] section.

We run our own scalability tests using similar hardware. Our standard "smoke test" is 500 caffeinated users, most hitting Review Summary[345] and Side-by-Side[377] pages, a few in administrative[180] and reporting[467] pages. This represents far more activity than 500 real users.

## Performance Tuning

General tips for making the server run as fast as possible:

- Use the latest version of the server and client.
  We are constantly improving speed bottlenecks as they are reported. Upgrade to the latest version, or scan the version history[982] to see if any speed improvements were made.

- Increase the "chat refresh interval[184]" setting. This will make the chat pane update more slowly but can reduce stress on the server by many times. The default value is 5 seconds, but 15, 30, or even 60 seconds can work and reduce server load.

- Make sure nothing else is running on the Collaborator Server.
  Often another server or process running on the same machine as our server will take up CPU cycles or slam the hard drive. For small installations it is quite common (and appropriate) for Collaborator to share a server, but for larger installations or when you are trying to squeeze out more performance you should dedicate a server.

- Increase the number of allowed database connections (instructions here[176]).
  Even if you are not getting errors on the web page about running out of connections, the web server threads might still be waiting for database connections to become available. (Error messages do not appear until those waits start timing out.)

- Increase the speed of the connection between the database server and the Collaborator Server.
  Collaborator makes many small database queries, so decreasing network latency should improve performance. We recommend that Collaborator and the database server be connected via gigabit ethernet.

- Increase the number of server threads (instructions here[176]).
  Just as with database connections, more server threads means less waiting for browser connections on a busy server.

- Increase the max number of simultaneous connections in your browser.
  Both Internet Explorer and Firefox default to just two simultaneous connections to a server. You can increase this to 4 or more. This will not help if you have a very slow connection (for example, a modem), because slow connections are probably already maxed out.

- Make sure caching is enabled in your browser.
  Sometimes a browser's static cache is disabled; make sure it is enabled (at least for connections to our server). We have lots of styles, Javascript, and images that normally are cached.

- Tune the memory settings for Collaborator Server (instructions here[1232]).
  More memory will often improve performance, but large documents in particular require a great deal of it, and often also require to change your Garbage Collector settings. If your users perform document reviews frequently, visit this section.

## Monitoring

You can monitor our application like any other. Like many enterprise-ready Java applications, you can monitor using JMX[155].

### 3.2.11 High Availability Best Practices

The Collaborator server is generally very stable and resilient. In most environments it is appropriate to simply try configuration changes or upgrades in-place (backup[99] first, of course).

However if the server is in constant use by a large number of users, or if Collaborator is vital for your business processes, it may be a good idea to be more cautious. To ensure that a change will not cause a problem on your production Collaborator server, it is best to try it first on a test server.

## Creating a Test Server

A test server should mirror your production server as closely as possible so that the results of the change are reproducible. Follow these steps to create a test server:

1) Provision a machine - this should be physically different hardware than the machine that is running your production Collaborator server, and ideally it should have the same specifications. It does not have to be a perfect match, but the closer you get it to the production server the more confident you will be when testing the change.

a) Hardware - this is not that important to get exactly identical unless you are concerned about performance issues.

b) Operating System - this is important. Make sure you install the same operating system version, especially if you are using Windows.

c) Java Virtual Machine - this is vitally important. Make sure that you install exactly the same version of Java.

d) Network Configuration - this is usually not that important unless your change involves communicating with an external system like LDAP[119], a database service running on a different machine, or configuring https[111].

e) Database - this is very important, especially when testing upgrades to new versions of Collaborator.

> **Warning:** The test server *must* have its own database. Do not run the test server with your production database!

2) Install the Collaborator server

a) Download the installer for *exactly* the same version as your production server.

b) Run the installer. You can click "next" through all the screens and accept the default settings, because those will be overwritten anyway in step 4[160].

c) When you click "Finish" in the installer it will open a browser to a page that says "Initialize database". DO NOT press the button, instead close the browser.

d) Shut down the Collaborator server.

3) Copy your content-cache - this directory stores the content of the files being reviewed and can become very large. The directory is configured in the *installation-directory*/tomcat/conf/Catalina/localhost/ROOT.xml file and defaults to *installation-directory*/tomcat/collaborator-content-cache. Copy the entire directory to the test server in the same location.

a) You can skip this step and your content-cache will simply be empty. Everything will work except when you look at the content of files in old reviews you will see a message "*This content has been archived by the Administrator*".

b) Another option is to use a network share to your production server's content-cache, mounted to the test server's file system. While not ideal this is fairly safe because Collaborator only ever writes to the content-cache, never deletes.

4) Copy your configuration

a) Copy the following files from your production server to your test server:

```
installation-directory/ccollab-server.vmoptions
installation-directory/tomcat/conf/collab.ks (may not exist, if not
using HTTPS connection 111)
installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml
```

b) Edit the "*installation-directory*/tomcat/conf/Catalina/localhost/ROOT.xml" file using a text editor and change the database information (url, username, password) to point to the test database.

5) Copy your database data - migrate a copy of the data from your production database to your test database as described in Backup/Migration 99.

6) Start up Collaborator.

a) Immediately disable email notifications 200 so that your users do not get double notifications from your production server and your test server.

b) Click around and make sure everything works. Note that licensing is not an issue - since you have migrated your database your license will work on the test server.

## Testing a Change

Once you have set up your test server you can use it to test any change you are considering to Collaborator, especially upgrades to new versions or global configuration changes.

### 3.2.12 Variable Substitution

In a few areas of server configuration (Notification templates 288, Collaborator triggers 206, Perforce triggers 788) you are allowed to use variable substitution to insert useful live data into a text-based template.

This section details those variables and how they work.

## Syntax

Variable substitution is indicated with a dollar sign followed by curly braces surrounding the variable name. For example, to access the `review.creator.name` variable you would use this:

```
${review.creator.name}
```

When embedding in other text, variables are inserted into the text wherever they should be replaced. An example that is commonly used in the changelist-update Perforce trigger 789 is:

```
ID ${review.id}: "${review.title}" by ${review.participants.rolename}
```

## Context

Not all of the substitution variables are applicable in all contexts. In general, notification templates 288 use `${review.*}, ${role.*}, ${actor.*}` for the sender and `${user.*}` for the recipient. The Review Created trigger 207 uses `${review.*}` and `${user.*}` is the review creator. The Review Phase Changed trigger 207 uses `${review.*}` and `${user.*}` (not `${actor.*}`). The Added File trigger 207 uses `${review.*}`, `${changelist.*}` and `${files.paths}`. The Notifications Sent trigger 207 uses `${review.*}` and `${user.*}`. The Defect Activity trigger 207 uses `${actor.*}` and `${defect.*}`. The User Created trigger 207 uses `${user.*}`. And the Role Changed trigger 207 uses `${review.*}, ${user.*}` and `${role.*}`, where `${role.*}` stands for the new current role.

Also, pay your attention that not all of the variables are available at the moment the trigger fires. In this case they will return "Not available" instead of their value.

## Reference

The following table lists all the variable substitutions.

| Variable | Description |
|---|---|
| `${review.id}` | The numeric ID of a review. Example: 4235 |
| `${review.title}` | The title field of a review. This variable is not available for the Review Created trigger. |
| `${review.workflow}` | The name of the workflow (template) this review is currently in. If you change the template name in workflow configuration 246 this text changes too, so be careful of scripts which depend on this value. |
| `${review.phase}` | The name of the phase this review is currently in, for example, "Inspection" or "Completed" or "Canceled". |
| `${review.phase.previous}` | The name of the previous phase of the review. Valid only in the context of a phase change. |
| `${review.creator.*}` | User information corresponding to the creator of the review. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example, the user's full name would be `${review.creator.name}`. |
| `${review.author.*}` | User information corresponding to the primary author assigned in the review. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example, the user's full name would be `${review.author.name}`. |
| `${review.custom.*}` | The value of one of the review custom fields 256.<br><br>The final variable component should be the full title of the custom field, including whitespace and so forth. |

| | |
|---|---|
| | If this review does not have this field because the user has not entered it yet, if there is a default value it will be returned. If this review does not have this field because the workflow does not include the field, you will get an error message.<br><br>If you supply the name of a custom field that does not exist, you will get an error message to that effect including the list of valid custom field names. You can use this fact to create an error on purpose to get the list of valid custom fields. |
| ${review.datecreated} | The date when the review was created. |
| ${review.deadline} | The date when the review should be completed. |
| ${review.group.title} | Title of the group associated with the review. |
| ${review.group.guid} | GUID of the group associated with the review. |
| ${review.participants.rolename}<br>${review.participants.rolelogin} | List of role/user pairs of the people assigned to the review. In the `rolename` case the user's full names are used; in `rolelogin` the user's system login names are used. Roles are named by the display name as configured[279]; be careful when writing scripts that depend on this text, and remember that different workflows can define different text for the role name. |
| ${review.changesummary} | A multi-line summary of the changes in the review, lines (+add, *mod, -del) for each file. |
| ${review.shortchangesummary} | A short summary of the changes in the review, total lines (+add, *mod, -del) for all the files in the review. |
| ${review.defectlog} | The log of all defects in a review. |
| ${user.id} | The unique ID of the user in our own database. |
| ${user.login} | The system login for the user. If LDAP authentication[119] is being used, this is the user's LDAP login. The login is unique for every user in the system. |
| ${user.name} | The user's full name, as they configured it themselves. This is not guaranteed to be unique in the system. |
| ${user.email} | The user's email address, or blank if it is not configured. |
| ${user.phone} | The user's phone number field, exactly as they configured it themselves. |

| | |
|---|---|
| ${files.paths} | A set of file paths, in no particular order, separated by the pipe symbol. This is typically a set of paths in the SCM server style, not local file paths. This variable is not available for the Review Created trigger. |
| ${changelist.paths} | Same as ${files.paths}. This variable is not available for the Review Created trigger. |
| ${changelist.scmid} | The ID of this changelist according to the SCM system. For example, Perforce changelist ID or Subversion revision number. This variable is not available for the Review Phase Changed trigger. |
| ${changelist.author} | The author of the changelist according to the SCM system.<br><br>Note that although this username *might* correspond to a user in Collaborator, this is not required. Further note that because this is a simple text field and not a user object, you cannot continue with other sub-fields like email or phone number. |
| ${changelist.comment} | The comment given in the SCM system for this changelist. |
| ${defect.id} | The unique ID of this defect in our own database. |
| ${defect.creator.*} | User information corresponding to the creator of the defect. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example, the user's full name would be ${defect.creator.name}. |
| ${defect.file} | The full path to the file this defect is attached to, or a blank string if the defect was review-wide. |
| ${defect.line} | The line number in the file where this defect is attached, or 0 if this defect is attached to the overall file, the overall review, or something else that's not relevant to a specific line.<br><br>Note that this line number refers to the file at the time of review; in the future the file can change and line numbers can change with it. |
| ${defect.text} | The text entered in the comment field for this defect. This can be long and multi-lined. |
| ${defect.isfixed} | Boolean text "true" if the defect is currently marked fixed, "false" if the defect is currently open. |
| ${defect.isexternal} | Boolean text "true" if the defect is currently marked external, "false" if the defect is currently marked as internal. |
| ${defect.externalname} | The external name specified when the defect was marked external. |

| | |
|---|---|
| `${defect.custom.*}` | The value of one of the defect custom fields 256.<br><br>The final variable component should be the full title of the custom field, including whitespace and so forth.<br><br>If this defect does not have this field because the workflow does not include the field, you will get an error message.<br><br>If you supply the name of a custom field that does not exist, you will get an error message to that effect including the list of valid custom field names. You can use this fact to create an error on purpose to get the list of valid custom fields. |
| `${role.title}` | The title of the role assigned to the user in the review |
| `${role.description}` | The description of the role assigned to the user in the review |
| `${actor.*}` | The user responsible for triggering this action. The asterisk should be replaced by a user-specific variable (described elsewhere). |

### 3.2.13 Archiving Reviews

Collaborator can pack reviews to an external Zip archive.

You can use this feature to keep historical records of the reviews (for instance, because of regulatory requirements), or to export review information to a non-proprietary format.

***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

## Creating Archives

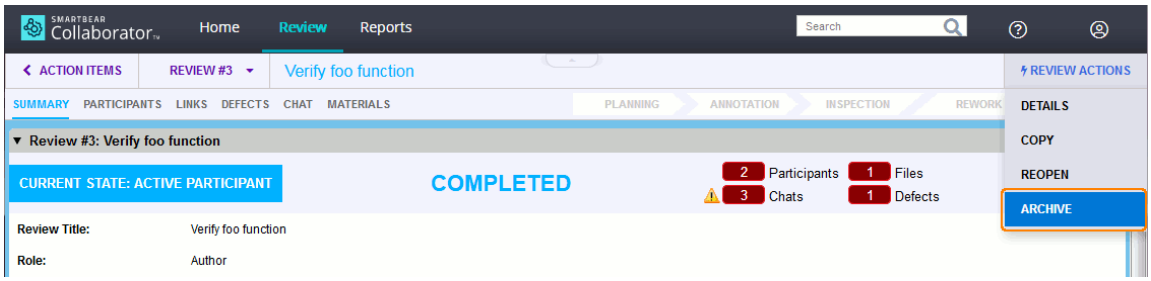
There are two ways to archive reviews: from the Web Client and from the command-line.

To archive a review from the command-line call the `ccollab admin archive` command. See command description for its syntax and examples.

To archive a review from within the Web UI:

1. Open the desired review in the Review Summary screen 345. To find the reviews, you may use Collaborator's search 458 and reporting 467 systems.

2. Select **Review Actions > Archive** in review header:




This will invoke a standard Save As dialog where you can specify the location for the archive.

## Creating Archives Automatically

You may use Collaborator server triggers to archive reviews automatically. The general idea is to install the Command-Line Client on a server machine and call the `ccollab admin archive` command via the server's phase change trigger 206. Below are the detailed steps of how to achieve this:

1. Install the Collaborator client 486 on the same machine where Collaborator server is installed.
2. Specify the connection parameters 494 for the client.
3. Create a directory to store the archived reviews. In this example that will be `c:\archived-reviews` on Windows platforms and `/archived-reviews` on Unix platforms.
4. Login to the Collaborator web interface as an administrator.
5. Go to the Admin->Triggers 206 page
6. Find the "Review Phase Changed" trigger, and configure it as follows:

| Windows platforms | Unix platforms |
|---|---|
| **Executable:** `c:\program files\collaborator client\ccollab.exe`<br><br>**Parameters:** `admin archive ${review.id} --zip-path c:\archived-reviews` | **Executable:** `/usr/local/ccollab_client/ccollab`<br><br>**Parameters:** `admin archive ${review.id} --zip-path /archived-reviews` |

You nay need to change paths to the Collaborator command line client and to the archives folder as necessary.

Note: In the example above the trigger does not check if the phase has transitioned to completed, cancelled or rejected. Since the `ccollab admin archive` command ignores in-progress reviews (by default) you can simply call it on every phase changed event. If your team is using the tool heavily, then you should create a shell script or batch file, and check the review phase before calling the ccollab admin archive command. This will result in improved performance, but is not really necessary for small or medium sized installations.

## The Archive Contents

For each review a separate archive is created. Generated files have the names in the format "review-*ID*-archive.zip", where *ID* stands for the identifier of the review.

Each archive contains the following information about the review:

- The `ReviewDetails.pdf` file contains a PDF version of the Review Detail Report 467.
- The `material_conversations` subfolder contains all comments, defects and conversations that were made during the review. For each of the commented files a separate PDF version is created that holds the contents of the file and the comments and defects related to this file.
- Review materials.
  - If the archive was created from Web UI, it contains all revisions of the materials that were uploaded to the review. If a file was uploaded multiple times (as it typically happens during rework), the archive will contain a copy of each file revision.
  - If the archive was created with the `ccollab admin archive` command, the amount of review materials is specified by the `--zip-option` parameter.

The naming structure of subfolders that store review materials, depends on how the materials were uploaded. For changes uploaded via version control system, the folder name will contain the name of SCM, the name of the repository and (if available) the ID of the changeset ("git-git.example.com_foo_repo.git-85d2f5ccb7f27e1628c09e4d1373f"). For uploads of arbitrary files, the folder names will contain the "upload-file-" portion followed by the date and time ("upload-file-2015-08-18-02-58-59").

## Remarks:

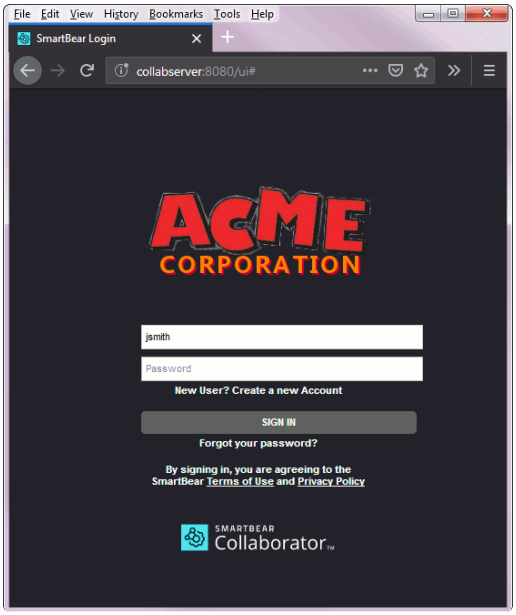
- The "Allow Archive to zip 188" setting controls who can use this feature. The possible values are: administrators 224, group administrators 231, or all users (considering the access permissions 188 of the chosen review).
- By default, only the reviews in the Completed, Cancelled or Rejected phases 17 can be archived. Yet, the "Allow Archive to Zip for Open Reviews 189" setting may allow to archive reviews on any phase.

- The built-in PDF viewer of Windows 8, 8.1 and 10 do not support overlay layers in PDF files. Collaborator uses such layers to render coordinate comments (pushpins) 425 in PDF versions of review materials for archived reviews. Because of this, pushpins will not be displayed if the PDF file from the archive is opened in the built-in PDF viewer of Windows 8, 8.1 and 10. As a workaround, please install the Adobe Acrobat Reader or any other full-featured PDF viewer.

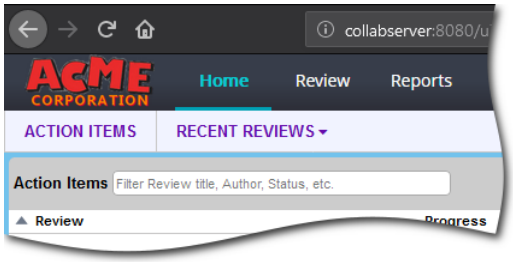
## 3.2.14 Branding Your Server

Administrators of Collaborator Enterprise servers can add custom company logotypes to server's Web interface. Namely, they can enable logos on Login 315, Home 313 and Review Summary 345 screens.

***Note:*** *Ability to display your company logos on Login and Home screens is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3 *.*




*Login screen with customized logo*




*Home screen with customized logo*

## Preparing Company Logos

Collaborator uses two image files for displaying your company logo: one for the Login screen and another for Home and Review Summary screens. You may have a separate image for each screen, or use the same image file for both screens.

The company logo images must be in PNG format. Transparent background is recommended. Image resolution is not limited. Large images will be resized to fit into 640x480 area on Login screen and into 250x50 area on Home and Review Summary screens.

## Technical Details

Custom logo files are stored in the `logos` sub-folder of server's content storage 97 ( `installation-directory/tomcat/collaborator-content-cache/logos`). Thus, they will be retained when backing up/migrating 99 your server together with content storage.

## Login Screen Branding

To display custom logo on Login screen

1. Open System 180 category of administrator settings,
2. Switch to the Display tab,
3. Enable the Company Logo on Login 185 setting,
4. Click the placeholder/current logo and upload image with your company logo.
5. Click **Save** button to apply the changes.

## Home Screen Branding

To display custom logo on Home and Review Summary screens:

1. Open System 180 category of administrator settings,
2. Switch to the Display tab,
3. Enable the Company Logo on Home 185 setting,
4. Click the placeholder/current logo and upload image with your company logo.
5. Click **Save** button to apply the changes.

# 3.2.15 Logging

## Log Files

Collaborator server logs messages to several files.

| Log File | Description |
|---|---|

| | |
|---|---|
| <installation-directory>/tomcat/logs/**collab.log** | Captures application level logging and (if enabled 171) license usage events, such as user login and logout. Configurable. |
| <installation-directory>/tomcat/logs/**remoteSystem.log** | Captures actions of remote repository integrations. Configurable. |
| <installation-directory>/tomcat/logs/**remoteSystemAPI.log** | Captures all outgoing API calls and usage statistics of API cache. Configurable. |
| <installation-directory>/**output.log** | Captures the standard out stream from the application libraries. Non-configurable. |
| <installation-directory>/**error.log** | Captures the standard error stream from the application libraries. Non-configurable. |

**Note:** Normally, Collaborator does not write to standard out or error, but some of the libraries that it uses do. Occasionally, output.log and error.log could be helpful for debugging issues in one of those libraries.

## Configuring Logging Using Configuration Files

Collaborator uses a sophisticated logging system, log4j, to manage the logging. Because of the breadth of configuration options available in log4j, it is not possible to document everything here (the log4j documentation is available at http://logging.apache.org/log4j/1.2/manual.html). However, in this section, we will cover some common scenarios.

The configuration file for the `collab.log, remoteSystemAPI.log and remoteSystemAPI.log` files can be found at `<installation-directory>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.properties`. The file is a plain text Java properties file and can be edited with any text editor. Changes will take effect when the server is restarted.

**Note:** Edits to the log4j.properties file will not be saved between updates to Collaborator. Upon editing this file, it is a good idea to save a backup of the file in a location outside the Collaborator installation tree.

## Basic Debug Logging

The most common configuration change that system administrators will make is to turn on debug logging. This can be useful when working with SmartBear technical support to resolve issues (though this is not always required). To enable debug logging, find the first non-comment line in the file:

```
log4j.rootLogger=info, CodeCollaborator
```

This line configures the base level logging to "info" level and attaches an appender named "CodeCollaborator" that will be configured later in the file. To enable debug logging, simply replace "info" with "debug" and restart the server. Do not change the name of the appender as this will result in the log configuration not being available via JMX 172. Running a production system at debug logging level can cause performance issues, so it is not recommended unless specifically directed by SmartBear technical support.

## Log Rolling

By default the `collab.log` file is rolled when it reaches 10Mb and saves 10 generations of files, for a total of 110Mb of disk space potentially consumed. Both of these values can be configured in the log configuration file. The following lines specify these values:

```
log4j.appender.CodeCollaborator.MaxFileSize=10MB
log4j.appender.CodeCollaborator.MaxBackupIndex=10
```

System administrators may change these values as necessary to capture additional logging information or conserve disk space as necessary.

## Enabling License Logging

Collaborator can log license events (users logging in and out) to help you understand your license consumption. By default this is disabled as it is normally not necessary. The system status 212 page includes graphs which are helpful for understanding the total license consumption. However, drilling down to find what users are using the system when requires turning on license logging.

To enable license logging:

1. Open the `<installation-directory>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.properties` file.

2. Locate the `log4j.logger.com.smartbear.ccollab.license line and` change the license logging level from "warn" to "debug":

```
log4j.logger.com.smartbear.ccollab.license=debug,
CodeCollaboratorLicense
```

3. Restart the Collaborator server to apply changes.

The license log appender is also configured to roll like the `collab.log` and this configuration can be changed in a similar manner. However, it is important that you not change the appender name from "CodeCollaboratorLicense" as this will result in the log configuration not being available via JMX 172.

### Enabling Remote System Logging

To diagnose issues with repository hosting integrations, you may enable detailed logging of remote repository actions and/or logging of outgoing API calls and usage statistics of API cache. By default those are disabled as it is normally not necessary.

To enable detailed logging of remote repository actions:

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations**
3. Set the **Enable detailed logging** option to **Yes**.
4. Click **Save**.

To enable logging of outgoing API calls and usage statistics of API cache:

1. Open the `<installation-directory>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.properties` file.
2. Locate the `log4j.logger.rsApiLog line and` change the license logging level from "warn" to "debug":

```
log4j.logger.rsApiLog = debug, RemoteSystemAPI
```

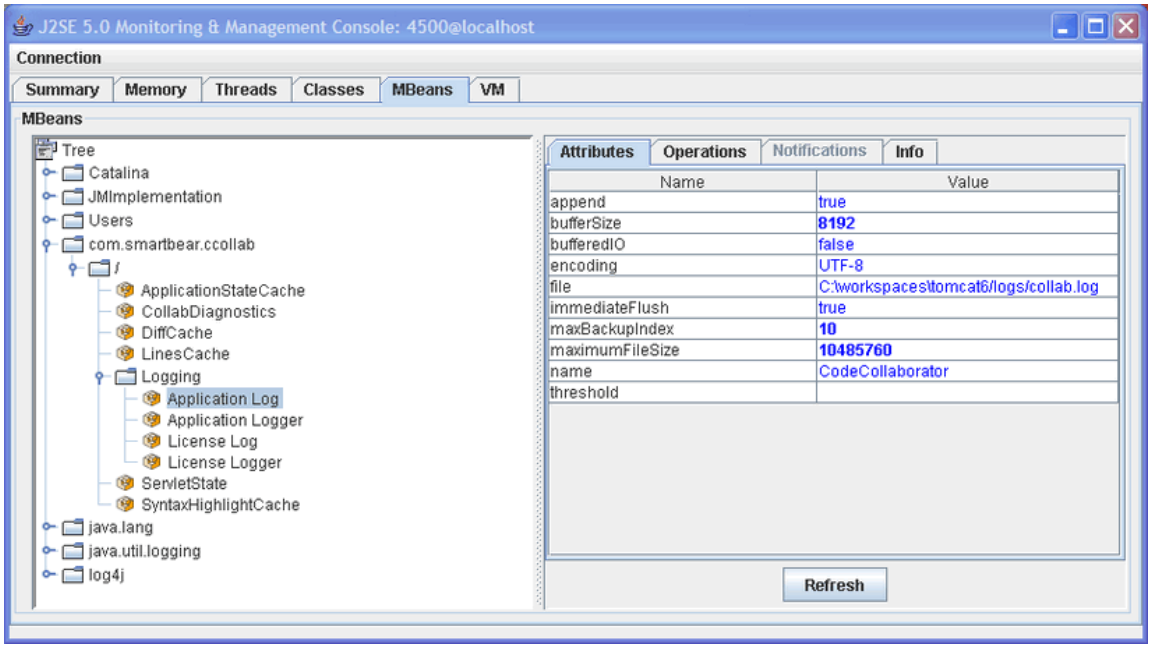
3. Restart the Collaborator server to apply changes.

## Controlling Logging with JMX

If the Collaborator server has JMX management enabled 155, the logging level can be changed at runtime. All of the log configuration MBeans are available at com.smartbear.ccollab:/:Logging. Changing the "priority" on "Application Logger" or "License Logger" will change the base level of logging for the application and the license log respectively.

The appenders are found under the names "Application Log" and "License Log". The max size, number of backups, and even the file names, can be changed in JMX. The changes will not take effect until the "activateOptions" operation is invoked. In jconsole operations are found in a separate tab.

> **Note:** Changes to logging made via JMX are not persisted across server restarts.




*Editing the Application Log Configuration in jconsole*

## 3.2.16 Troubleshooting

When something goes wrong, refer to this section.

### Known Issues

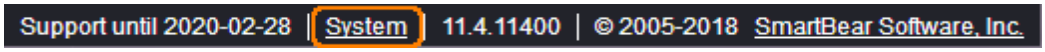
Check the Known Issues Appendix 978 to see whether SmartBear already knows about this issue.

### Version History

Check the Version History on the SmartBear website to see if this issue has been resolved in a later release.

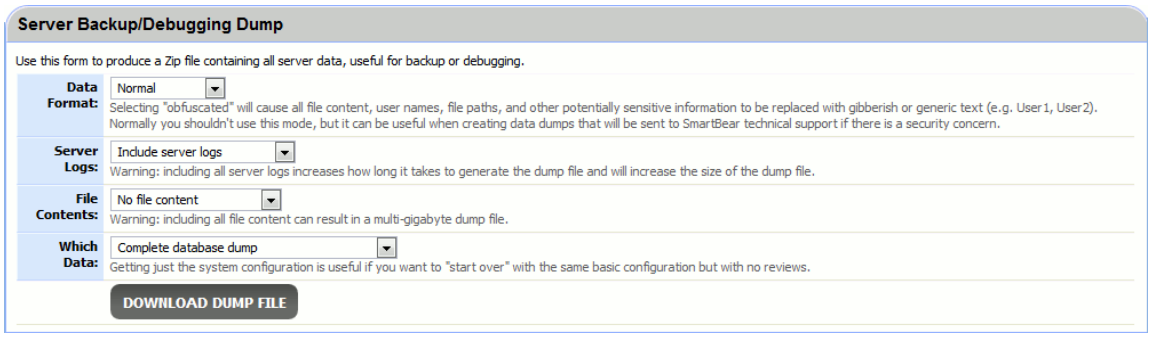
## Getting server information

To get debugging information about the server, go to the Collaborator web server and click the System link at the bottom of the screen:




## Downloading server dumps for SmartBear technical support

Sometimes it will be useful for SmartBear technical support to get a "dump" of your server configuration and database data. You do this by filling out the form at the top of the System page.




### Data Format

You can dump the entire database in a normal format, or set the "Data Format" field to "Obfuscated," which will dump the data after passing through a filter that converts potentially sensitive information into unhackable text. For example, users are renamed "user1," "user2," and changelist text is replaced by the MD5 of that text. If file contents are included in the dump, each line of file content is replaced with the MD5 of that line to enable us to reproduce exactly the right line numbers and diffs you are seeing. This can be used to send us data when you are worried there might be sensitive information.

If you are worried about whether we are really cleaning all the data, you can check out the dump file yourself. It is just a ZIP file containing XML with information, so it is easy to inspect. In fact, if you still see some data you would like to change or delete, you can just do that in the XML, re-zip the file, and send it to us.

### Server Logs

You can choose to include all server logs or none at all.

### File Contents

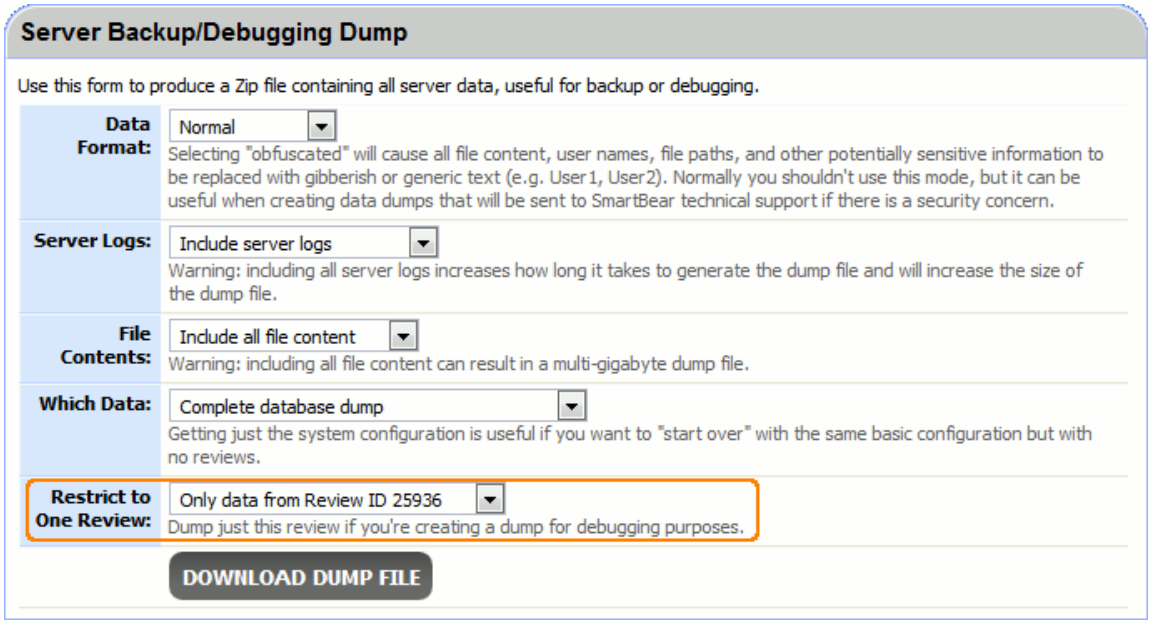
You can choose to include all file content or none at all.

### Which Data

Here you can choose whether you would like to perform a complete database dump or just include the system configuration.

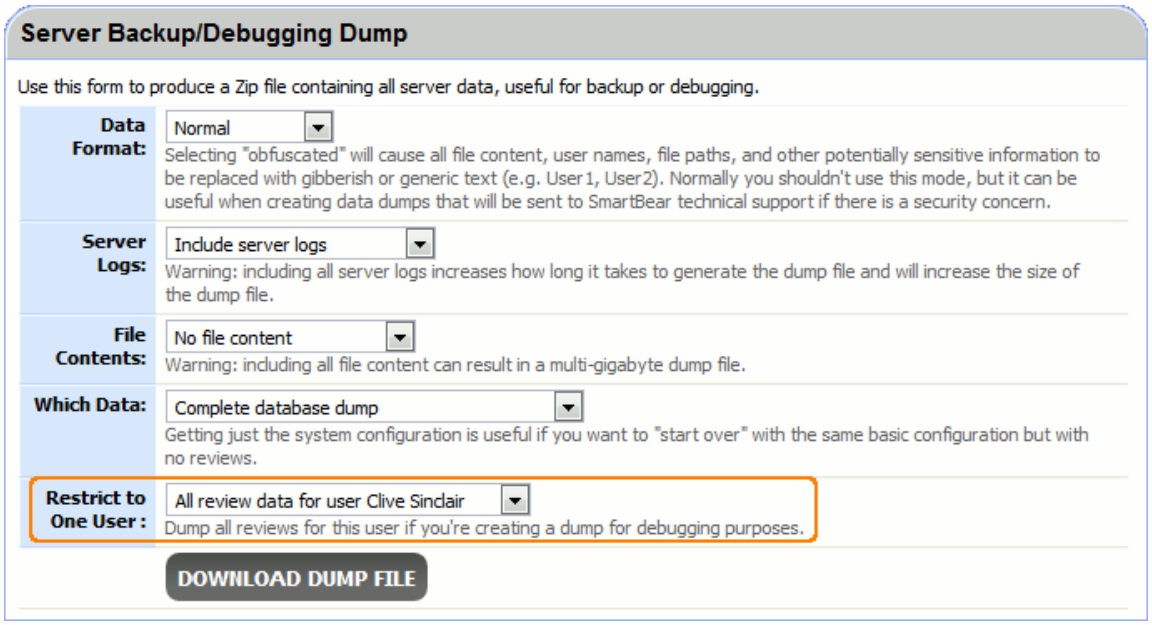
### One Review Dumps

If just one review is broken, you can send us a dump of just the one review. To do that, **first go to the review in question**, then click the `System` link as before. Now you will see an additional field in the form for downloading review data (which includes all the file content as well).




### One User Dumps

If reviews of some particular user are broken, you can send us a dump with reviews of this particular user. To do that, **first go to the Admin | Users 219 section**, then click the [Review data] link next to the desired user. Now you will see an additional field in the form for downloading review data for that particular user (which includes all their reviews and all the file content as well).




## The server is running slowly

Check out our performance tuning[158] suggestions and use recommended hardware[158].

## Large files consistently fail to upload or render

In order to review large artifacts, we recommend increasing the memory available to Collaborator server (instructions here[1232]).

## Email is not working

If email is not working, here are some things to try:

- Make sure the SMTP Host and SMTP Port settings[199] are correct.
- Try sending a test email[199]. Look in the server logs[177] for error messages.
- If you just performed a database migration/restore[100], email is automatically disabled. You need to enable it in the settings[199].

## Getting "Idle Timeout Waiting for Object" Error

This error is shown when the database connection pool runs out of connections to the database.

The fix is to increase the number of connections allowed in the pool. The server is capable of running with many more connections than the default; we keep the default fairly low so that in smaller installations we are not taking up too many database resources.

Fix this by going into the Tomcat servlet context XML file located here:

```
installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml
```

Edit the property called `maxActive` and increase the number of connections. Even doubling this number is normal. A rule of thumb is to have 3 times the database connections (and server threads) as the number of simultaneous users you have under the biggest load.

The Collaborator server must restart for this change to take effect. You do not need to do anything to your database server.

**Warning:** Modifying the ROOT`.xml` file will cause Tomcat to dynamically reload the Collaborator application, terminating any active sessions. Changes to ROOT`.xml` should be done in the context of stopping and restarting the Collaborator service (that is, in a production environment coordinating the restart with user activity), regardless of whether the service itself is actually stopped and restarted, or just reloaded by Tomcat.

Frequently this problem also implies that you should have more web server threads in addition to more database connections. To increase that, edit the `maxThreads` property from this XML file:

```
installation-directory/tomcat/conf/server.xml
```

Stop and restart the server to load new settings from the `server.xml` file.

## "NoHttpResponseException: The server ... failed to respond" Error

This error can occur if a client connection times out after the default 20 seconds. A heavily loaded Collaborator server or limited resources can be the root of this problem, and investigating the server performance and network connectivity may be warranted. Regardless, the connection timeout can be increased by editing the millisecond value of the `connectionTimeout` variable in the `server.xml` file described above. For example, for a timeout of 60 seconds, use:

```
connectionTimeout="60000"
```

## The Server Log

Collaborator is configured by default to perform a very limited amount of logging. The primary purpose of the logs is to record information about error conditions that could potentially arise. When working with SmartBear Technical Support to diagnose issues often it will be helpful to send the logs (or a portion thereof) for investigation.

The logs are located in the following directory:

```
installation-directory/tomcat/logs
```

## Enable server-side debug logging

To enable verbose server-side logging, edit this file:

```
<installation-dir>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.
properties
```

At the bottom you will find a line called:

```
log4j.logger.com.smartbear=info
```

Change the "info" to "debug". You will have to restart the server for changes to take effect. Remember to revert these changes after reproducing the problem and copying off the log file because this creates a large number of logging messages and degrades server performance.

## Change the settings so the server can be accessed remotely

To change the Windows Firewall settings:

Start > All Programs > Accessories > System Tools > Security Center

Click the "Manage security settings for: Windows Firewall" link. On the General tab, make sure the "do not allow exceptions" box is not checked. On the Exceptions tab, click "Add Port" For "Name", use "Collaborator", for "Port number", 8080. Leave TCP selected. If you want, you can click the "Change scope" button to get more options about who can and cannot connect to the port. Click "OK" in each dialog to save the settings and try connecting again.

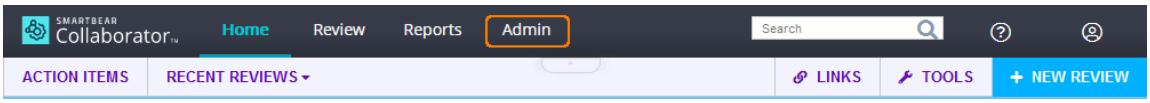
## Contacting Technical Support

For technical support and general inquires, contact us 32.

# 3.3 Collaborator Settings

As a system administrator, you can configure Collaborator's settings to however you would like. This may be useful in better managing users and how they perform reviews.

These settings can be found and changed by simply clicking on "Admin" at the top menu bar. This button is available to users that are Collaborator administrators and to users with advanced user permissions 225. Administrators can modify all administrative settings, while users with advanced permissions could modify a limited number of settings depending on what permissions they have.




The administrative settings are grouped into a number of categories:

- The System 180 category holds system-wide settings that control Collaborator look and feel, abilities and behaviour.
- The Email 199 category holds SMTP server settings and other settings related to e-mail notifications of review activity.
- The Version Control 204 category allows to integrate Collaborator with your version control system.
- The Triggers 206 category configures Collaborator to run an external script or application when certain events occur.
- The Licensing 208 category manages Collaborator licenses.
- The System Status 212 category displays various data about server, database, connection and allows to perform server backup and debugging.
- The Single Sign-On 215 category allows to establish integration between Collaborator and single sign-on authentication services 130.
- The Remote System Integrations 216 category allows to establish integration with third-party systems, like JIRA, GitHub, GitLab or Bitbucket.
- The Users 219 category allows to create users, view user statistics and edit user accounts.
- The Groups 226 category allows to create and organize user groups. Groups of users are used for reporting, filtering and for assigning reviews to multiple users.
- The Review Templates 246 category allows to create and manage review templates: sets of roles, custom fields, and other options that determine the behavior and rules of a review.
- The Custom Fields 256 category allows to create your own fields for users to annotate reviews, participants and defects with necessary information.
- The Checklists 272 category allows to create and manage a list of items that should be checked in every review.
- The Roles 279 category allows to specify the rights and privileges of review participants.
- The Automatic Links 286 category specifies regular expression patterns to treat certain text fragments as links.
- The Notification Templates 288 category allows to customize the content of notification e-mails that Collaborator sends to review participants.

- The Archive 299 category allows to archive content cache files which are no longer used.
- The Savings Report 301 category displays how much money you have saved by using Collaborator compared to non-code reviewed process.
- The Syntax Highlighting 304 category allows you add, manage and delete syntax highlighting schemas for various computer languages.

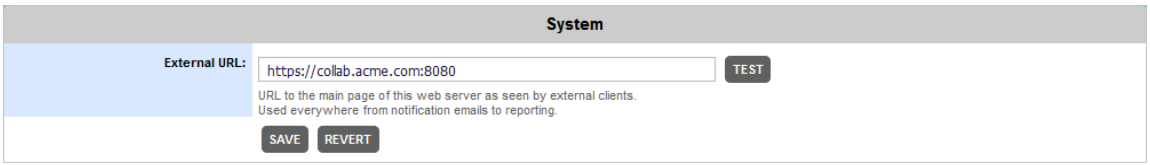
### 3.3.1 System

General system settings are organized into the following tabbed sections:

- Settings 180
- Display Options 182
- Access Restrictions 186
- Review Process 189
- Electronic Signatures 193
- Bug-Tracking Integration 194
- External Clients 195
- File Types 196
- LDAP Settings 198

## Settings Tab

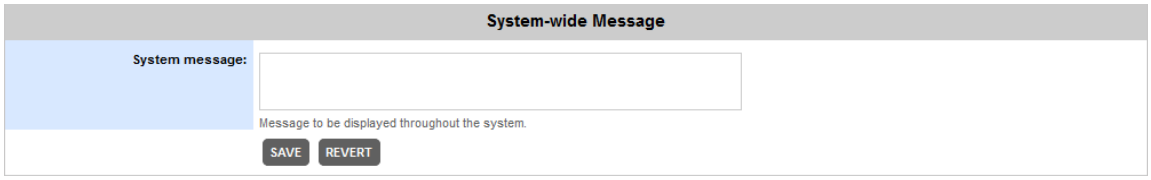
This tab contains settings that affect the entire system.




**External URL** The URL to use as the valid externally accessible URL of the Collaborator server. If the port number is other than default 8080, then specify it as well.

This URL will be reported to clients, used in notification emails, webhooks and so forth.

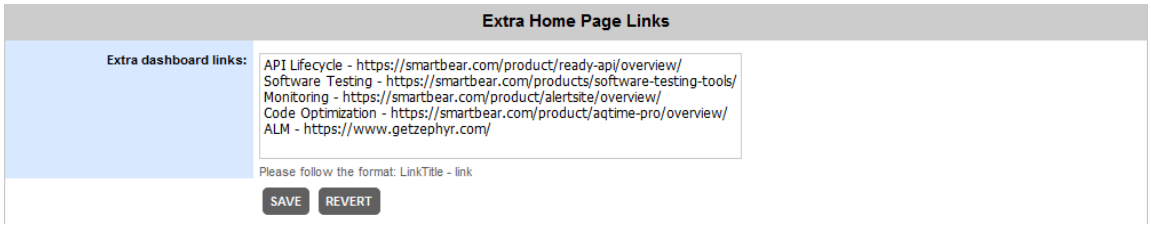
The server's External URL should be accessible to all Collaborator clients, database server, repository hosting services, issue-trackers and other remote systems. You may need to configure your firewall or enable tunneled connections to expose it.




**System message**

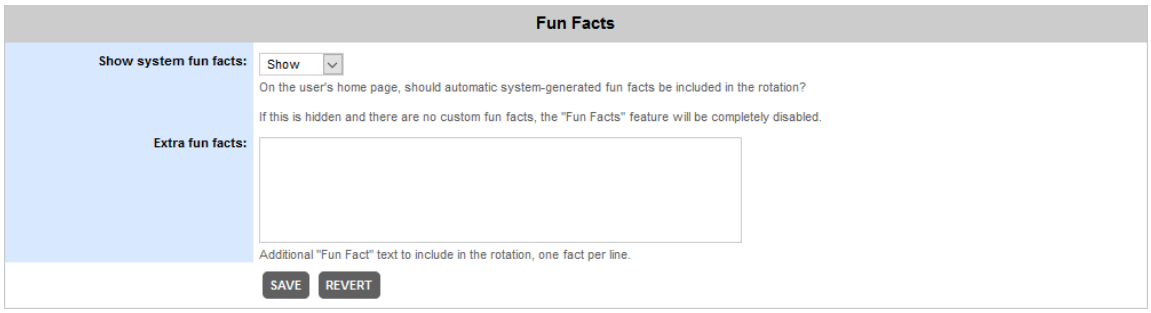
Administrators can optionally insert a message to be displayed throughout the system.

This can be useful for reminders, Collaborator announcements, or general information.




**Extra dashboard links**

List of additional links to be displayed on the Home page. One link per line. Links should be specified in the following format: LinkTitle - LinkURL

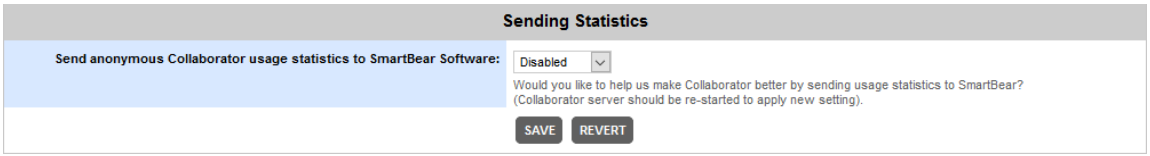



**Show system fun facts**

Fun facts are lines of information displayed on the home page. Collaborator rotates through a number of internal facts based on aggregate review metrics.

If this is set to "Hide" and there are no extra fun facts listed, this feature is completely disabled.

**Extra fun facts** Additional lines to be displayed can be entered in this field.




**Send anonymous Collaborator usage statistics to SmartBear Software** Specifies whether Collaborator server should collect and send usage statistics to SmartBear. Collaborator server should be re-started to apply new setting.

To learn about our privacy policy, visit https://smartbear.com/privacy/.

## Display Options Tab

This tab contains settings that affect the visual appearance of web-interface, its labels, prompts and so forth.


**Company / Unit** Arbitrary text that will be displayed at the top of the Collaborator web page. This distinguishes the server from other servers. This personalizes the server and also makes it easy for users to distinguish between different installations you might have.

**User login prompt** The text to display on the front page when the user logs in. This text is displayed just below the "Username" field.

The default text suggests using the same login as the version control system, however you might want to change this to be more specific, or in the case of LDAP authentication 119, you could instruct the user to use that login.

**Global "Create User"**

If you are not using LDAP authentication 119, every user in the system must be explicitly created. The system administrator can create users at any time, but an easier technique is to allow users to create their own accounts from the "Login" screen 314.

When this option is enabled, this create-account form is displayed, otherwise it is hidden.

This option does not make sense if you are using LDAP authentication because in that case user accounts are created automatically when a user first successfully authenticates on the Login screen.

**Default tab width**

Width of a tab (in spaces) for all users to use when displaying reviewed files 379.

**Chat refresh interval**

The minimum time (in seconds) to wait before auto-updating chat conversations 429 in the Diff Viewer 377.

If increase the refresh time, new chat display will become less responsive, but server load will decrease. This also reduces contention for connections on the browser side because most modern browsers limit the number of simultaneous server connections to 2.

**Name abbreviation length**

The minimum number of letters to use when abbreviating user names during a review.

**Defect label (singular)**

User-visible label for "defect" (lower case). Shows in phrases, like "Marked defect fixed". If the word 'defect' has a negative connotation in your environment, specify a different word (for instance, "item" or "finding" and so on).

**Defects label (plural)**

User-visible label for "defects" (lower case). Shows in phrases, like "Can not finish because there are open defects". If the word 'defects' has a negative connotation in your environment, specify a different word (for instance, "items" or "findings" and so on).

**Group label (singular)**

User-visible label for "Group". Shows in phrases, like "This Review is associated with that Group".

**Group label (plural)**

User-visible label for "Groups". Shows in phrases, like "Select one of the Groups to associate with the Review".

**Compact view** When this option is enabled, some user interface elements will be collapsed to occupy less vertical space.

**Allow users to move comments** Should users be able to move comments in documents 387, PDFs 418, presentations 395, images 413 and vector graphics 400. Possible values are: *Don't allow*, *Allow for creator only* and *Allow*.

**Company logo on Login** and **Company logo on Home** Whether to display your company logo on Login screen and on Home and Review Summary screens, respectively. See Branding Your Server 168.

Once enabled, click on the placeholder/current logo and upload new image with your company logo.

The company logo images must be in PNG format. Transparent background is recommended. Image resolution is not limited. Large images will be resized to fit into 640x480 area on Login screen and into 250x50 area on Home and Review Summary screens.

***Note:*** *Ability to display your company logos on Login and Home screens is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

**Select default WebUI theme** Specifies the default pre-defined theme of WebUI client 312. The specifies the visual style of Web client for new and existing users (unless they explicitly specify their own preferred WebUI theme 322 in User Preferences).

**Display list of logged in users when denied access** Whether to display a list of currently logged-in users on the Login page 315 when a new user fails to login because of insufficient licenses.

**Show active (floating-seat) users list in the license page** Whether to display a list of active floating-seat users on the Licensing page 208.

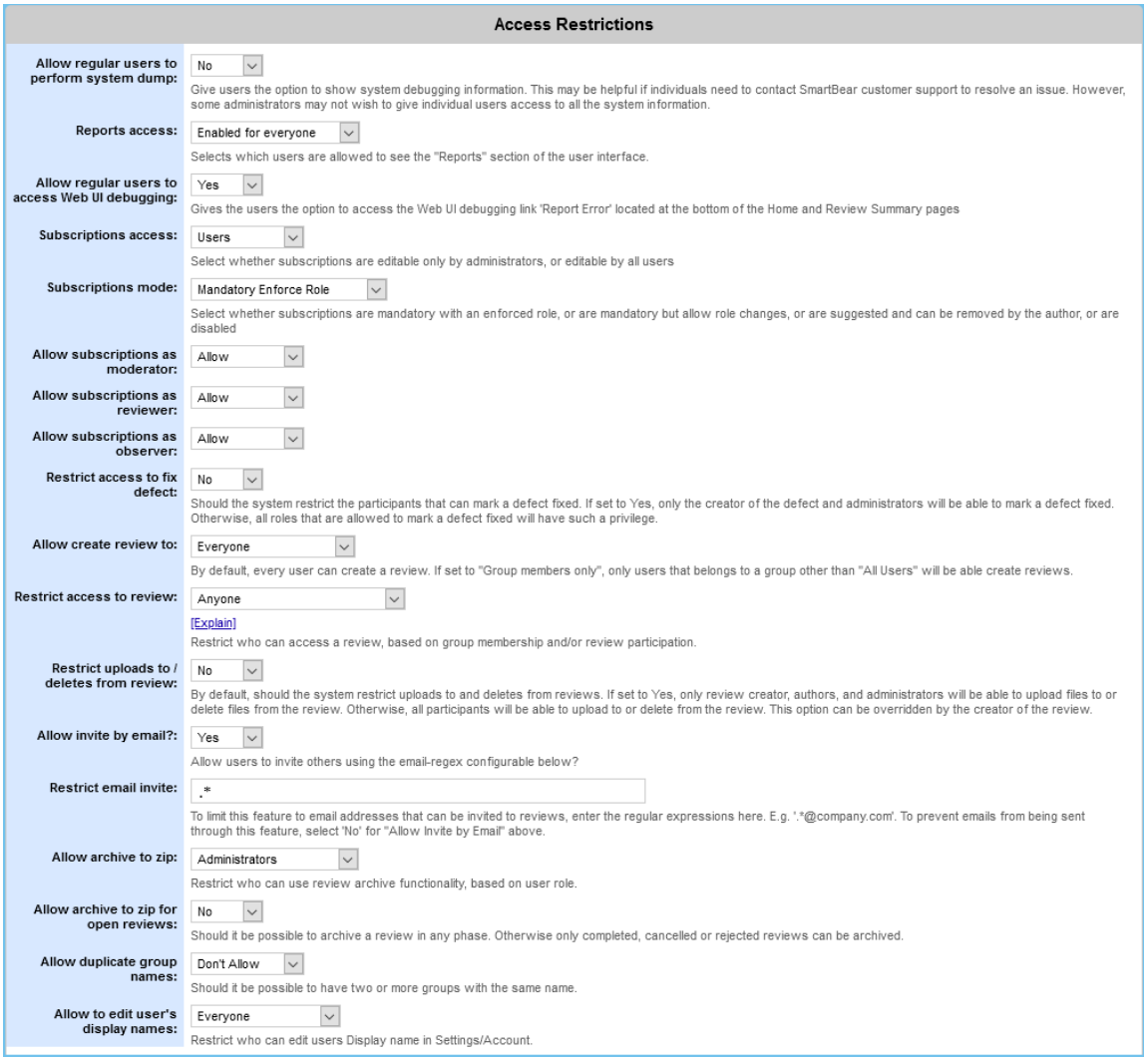
**Enable Markdown in comments** Enables support for Markdown formatting 448 for comment and defect fields. This setting affects newly created comments and defects. Existing comments and defects will retain their current formatting.

**Enable "Unavailable font" warnings** Specifies whether DiffViewer should display or hide a warning when some fonts used in a document were not available on a Collaborator server. This setting affects DiffViewer behavior only. Warnings about missing fonts would be appended to server logs in any case.

| | |
|---|---|
| **Allow to edit/ delete comments** | If enabled, participants would be able to edit[434] or delete[434] their own comments. If disabled, participants could only cross out their comments, making them much more difficult to read. |

## Access Restrictions Tab

This tab contains settings that control different types of access restrictions.



**Allow regular users to perform system dump**

Should regular users be allowed to perform system dump?

Selecting "Yes" will give users, as well as administrators, access to the system debugging information.

**Reports access**

Who should be allowed to view the "Reports" section of the user interface? You can choose everyone, everyone but with review permissions applied, only administrators, or disabled (no one).

Enabled for everyone: Everyone will see the "Reports" section; no restrictions are enforced on what information is returned.

Respect permissions: Everyone will see the "Reports" section; users can only see report content for reviews that they can access. Note that since administrators can always access all reviews, this setting does not affect the content of their reports.

Administrators only: Only administrators will see the "Reports" section.

Disabled: No one will see the "Reports" section.

**Subscriptions access**

Selecting "Users" will allow users, as well as administrators, access to edit subscriptions. Choose "Administrators" to give access to only administrators.

**Subscriptions mode**

Mandatory Enforce Role: Subscribed users must be participants in a review, and they must use the role specified on their subscription page.

Mandatory do not Enforce Role: Subscribed users must be participants in a review, however they can change their role once added to the review.

Requested: Subscribed users are automatically added to a review but can be manually removed.

Disabled: Subscriptions are not used.

**Allow subscriptions as ...**

The Administrator can choose whether users that subscribe to a review can subscribe as a Moderator, Reviewer, or Observer by choosing "Allow" or "do not Allow" from the drop downs below.

**Restrict access to fix defect**

Should the system restrict the participants that can mark a defect fixed?

If set to Yes, only the creator of the defect and administrators will be able to mark a defect fixed. Otherwise, all roles that are allowed to mark a defect fixed will have such a privilege.

**Allow create review to**

Specifies who is able to create new reviews. By default, every user can create a review. If set to "Group members only", only users that belongs to a group other than "All Users" will be able create reviews.

**Restrict access to review**

Should the system always restrict access to review such that only certain users are allowed to view the review? In each review[338] the review creator can select to further restrict access. Administrators can always access all reviews.

Anyone: No global restrictions on who can access reviews

Group Based: Users must have access to the review's associated group[227] to access reviews

Participants: Users must be participants in the review to access reviews

Participants and Group Based: Users must be participants in the review and currently have access to the review's associated group to access reviews

Participant or Group Based: Users may be participants in the review or have access to the review's associated group to access reviews

**Restrict uploads to / deletes from review**

Should only creators of a review be allowed to upload files to and delete files from a review?

If set to "yes," only review creators and administrators will be allowed to upload files to or delete files from the review. However, the option will be available on the Create Review screen[336] to override at the review creator's discretion. If set to "no," all participants will be allowed to upload and delete files.

***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page*[3].

**Allow invite by email?**

Should it be possible to invite a colleague by e-mail[968]?

**Restrict email invite**

Used for the "Invite a colleague" feature, the regular expression in this field is used to match email addresses that can be invited to Reviews. For example, '[\w\.-]+@company.com'.

**Allow archive to zip**

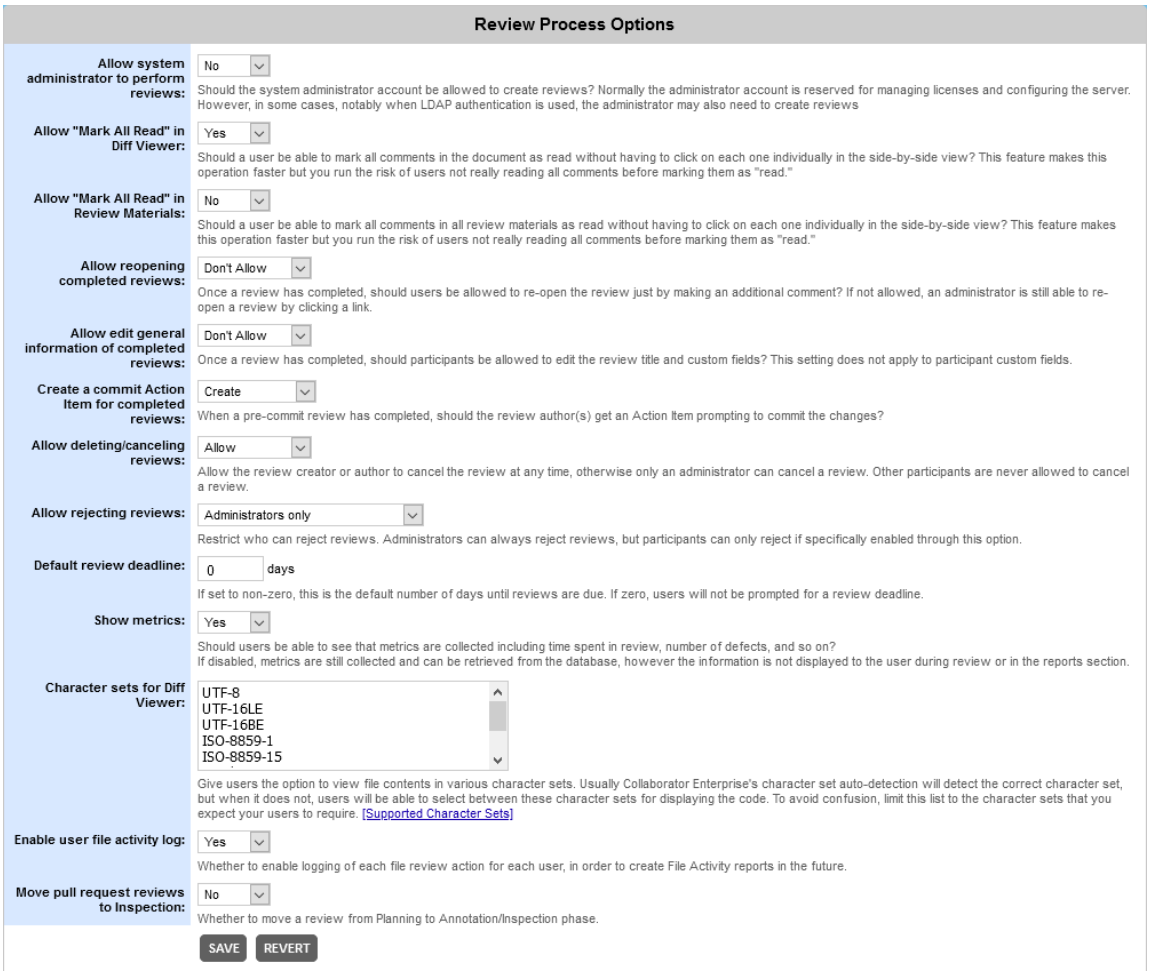
Should the system restrict access to archiving of reviews[165]?

Administrators: Only users with administrator[224] privileges can use this feature.

| | |
|---|---|
| | Group Administrators: Only users with administrator[224] or group administrator[231] privileges can use this feature.<br><br>All users: All users, who can access the review (based on the review permissions[188]), can use this feature. |
| **Allow archive to zip for open reviews** | Should it be possible to archive a review in any phase. Otherwise only completed, cancelled or rejected reviews can be archived. |
| **Allow duplicate group names** | Should it be possible to create groups having the same names? |
| **Allow to edit user's display names** | Specifies who should be allowed to modify **Display name**[317] field in user account settings. You can choose between everyone (default) or administrators only. |

## Review Process Tab

This tab contains settings that control different aspects of review process.




**Allow system administrator to perform reviews**

Should the main system administrator be allowed to participate in reviews?

If you are using internal authentication the setting is typically "no" because the main "admin" account is special and should be used only for system configuration and not to actually do reviews.

If you are using LDAP authentication[119] the setting is typically "yes" because the main administrator is usually an actual human being who will also want to participate in reviews.

This setting affects whether the system account is allowed to create a new review[336] and whether it appears in, for example, drop-down lists for review participants[338].

**Allow "Mark All Read" in Diff Viewer**

Should users be given the option to "mark all conversations read" in one click in the conversation area of the Diff Viewer 433?

This is a convenient operation, so most administrators leave this enabled. However this makes it easy for someone to not actually read a lot of comments because they are not forced to visit each conversation individually.

**Allow "Mark All Read" in Review Materials**

Should users be given the option to "mark all conversations read" in one click in the Review Materials section of the Review Summary Screen 357?

Use with caution, as this operation makes it extremely easy for someone to mark all comments in all review materials as read without actually reading them.

**Allow reopening completed reviews**

Once a review has completed, should participants be allowed to re-open the review just by making an additional comment?

If not allowed, an administrator is still able to re-open a review by clicking a link.

**Allow editing general information of completed reviews**

Once a review has completed, should participants be allowed to edit the review title and custom fields? This setting does not apply to participant custom fields.

**Create a commit action item for completed reviews**

By default, when a pre-commit review is completed, the author(s) are given an action item as a reminder to commit the changes to the underlying source control system. The author then has the option of using the Tray Notifier to commit the changes. In some workflows, this is not the correct behavior. In that case, choose "Do Not Create" to avoid this prompt.

**Allow deleting/ canceling reviews**

Should the review creator or author be able to cancel the review at any time?

If "do not allow" is selected, only an administrator will have the permissions to cancel a review. Other participants are never allowed to cancel a review.

**Allow rejecting review**

Reject review allows a user to stop a review which has passed the planning state but yet in a terminal state such as completed or cancelled. Rejecting a review is different than cancelling or deleting because the administrator can require the user to specify a reason for the rejection. The reason values can be configured in the Review Custom Fields section of the administration pages.

Administrators can always reject reviews. Select Administrators and Reviewers to allow non-observer participants to reject reviews.

**Default review deadline**

If set to non-zero, this is the default number of days until reviews are due. If zero, users will not be prompted for a review deadline[337] when creating a new review.

**Show metrics**

Should users be able to see that metrics are collected including time spent in review, number of defects, and so on?

If set to "No", metrics are still collected and can be retrieved directly from the database[908], however the information is not displayed to the user during review[362] or in the reports[467] section.

**Character sets for Diff Viewer**

Give users the option to view file contents in various character sets.

Usually Collaborator's character set auto-detection will detect the correct character set, but when it does not, users will be able to select between these character sets for displaying the file contents. To avoid confusion, limit this list to the character sets that you expect your users to require.

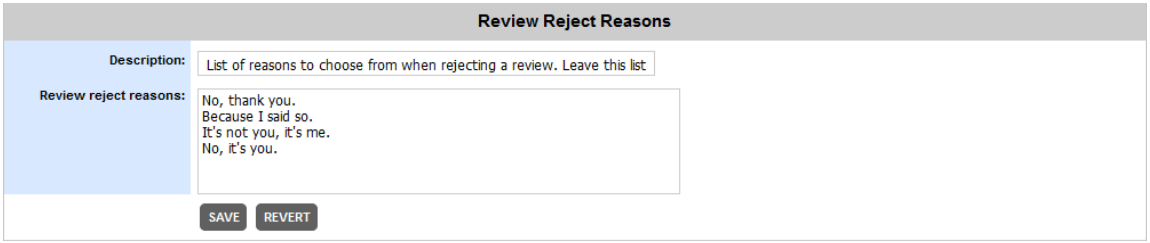
Click [Supported Character Sets] to view all supported character sets.

**Enable user file activity log**

Specifies whether to log file activity of review participants. The gathered data is used when generating File Activity section of Review Detail Reports[473].

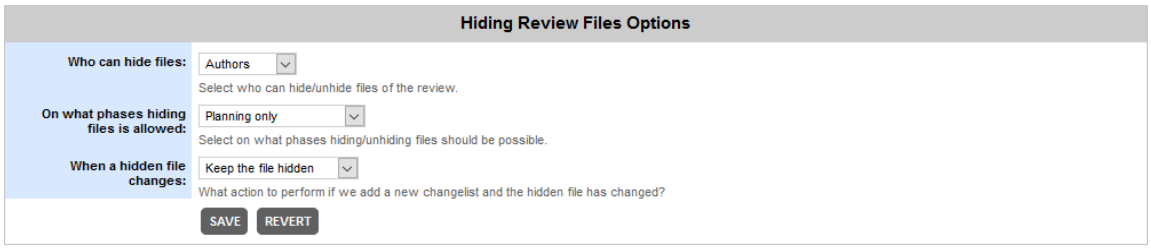
**Move pull request reviews to Inspection**

Specifies whether to automatically move reviews created by repository hosting integrations[826] from Planing to Inspection phase (if the review meets workflow requirements).




**Review reject reasons**

List of reasons to choose from when rejecting a review[347]. Leave this list empty if providing a reason is not required.




**Who can hide files** — Specifies who is able to hide and unhide files in reviews. Available options are: *Authors*, *Participants*, *Nobody*.

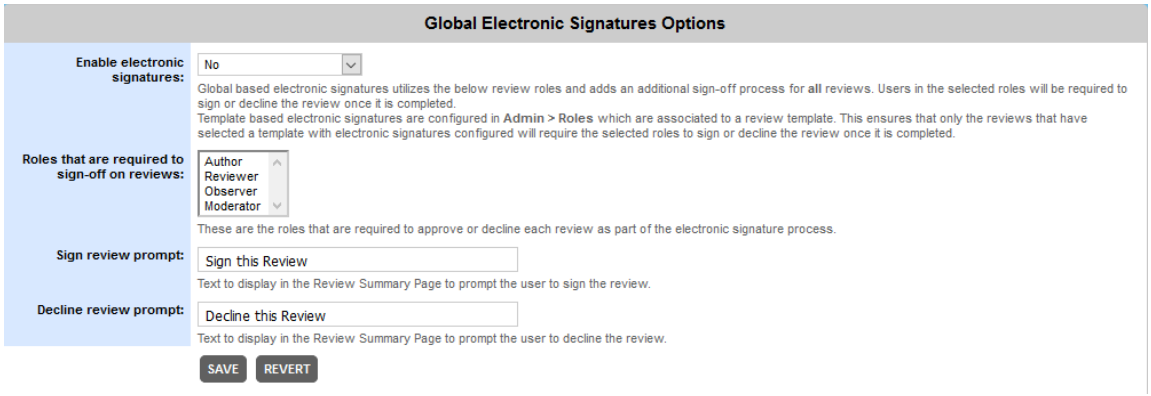
**On what phases hiding files is allowed** — Specifies on what phases users could hide and unhide files. Available options are: *Planning only*, *Planning and Annotating* and *At any time*.

**When a hidden file changes** — Specifies what action to perform in case the hidden file has been changed. Available options are: *Keep the file hidden* and *Un-hide the file*.

## Electronic Signatures Tab

This tab holds settings that control electronic signature process.

***Note:*** *Electronic Signatures are only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.




**Enable electronic signature** — Electronic Signatures allow users in the selected role(s) to either sign off on or decline to sign off on a completed review.

**Global based electronic signatures** adds an additional sign-off process for all reviews. Users whose roles were selected in the "Roles that are required to sign-off on Reviews 194" setting will be required to sign or decline the review once it is completed.

**Template based electronic signatures** add an additional sign-off process only to reviews with specific templates. Users whose roles were selected in the Role Configuration 285 screen will be required to sign or decline the review once it is completed.

**Roles that are required to sign-off on reviews**

Here, the admin can select one or more roles that will be prompted to sign off on Completed reviews.

This setting takes effect when global based electronic signatures are enabled.

**Sign review prompt**

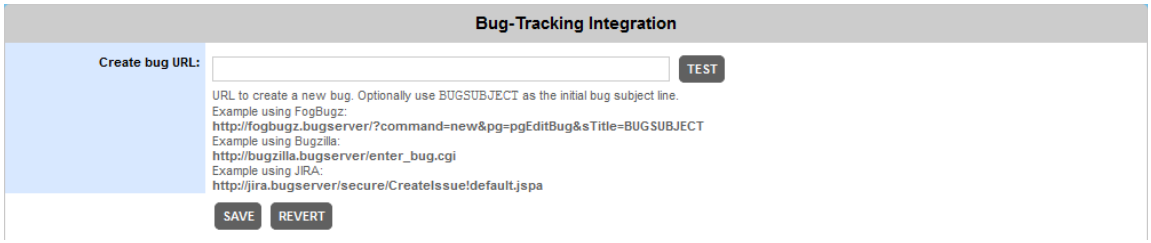
This is the text that will appear on the review summary screen next to the Sign button.

**Decline review prompt**

This is the text that will appear on the review summary screen next to the Decline button.

## Bug-Tracking Integration Tab

This tab holds settings that allow create items in your bug tracking system directly from Collaborator reviews.




**Create bug URL**

The URL to your bug tracking system that creates a new bug. Optionally use the special text BUGSUBJECT as the starting subject line for the bug. This setting is used when the user is prompted to create a bug in your external issue tracking system.

To track a defect externally, the user must first create the defect in Collaborator, then select and edit that defect to choose the 'Track Externally' option.

Below are examples of URL, for some popular bug tracking systems:

FogBugz: http://bugserver/default.php?

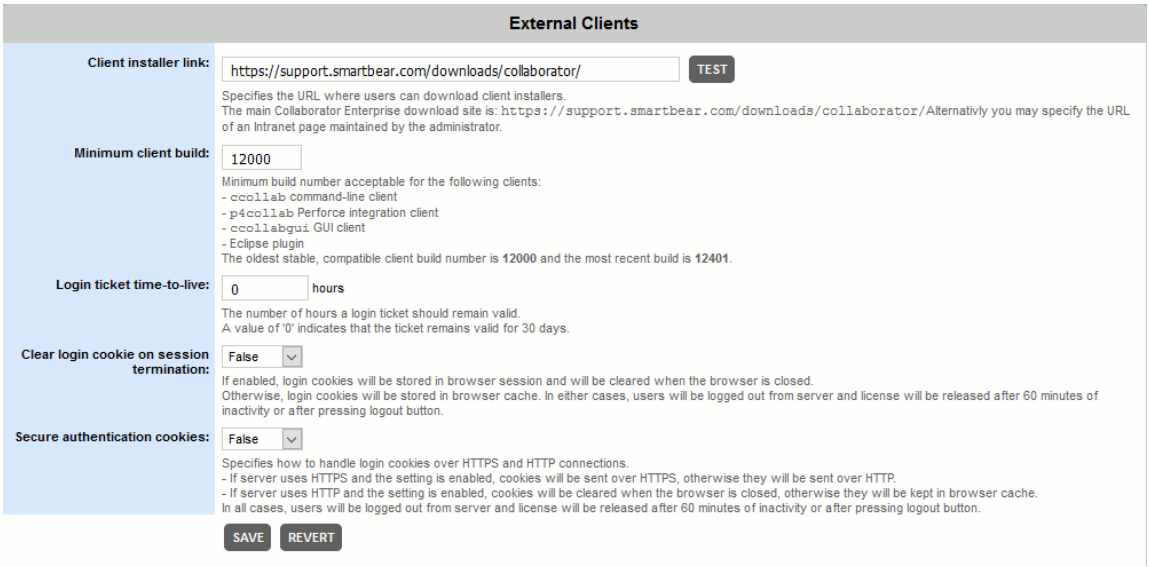
command=new&pg=pgEditBug

Bugzilla: http://bugserver/enter_bug.cgi

JIRA: http://bugserver/secure/CreateIssue!default.jspa

## External Clients Tab

This tab holds settings that relate to Web Client, Command-Line Client, GUI Client and other desktop clients.




**Client installer link**

URL to the version of the client installers that you want presented to users of the system.

This is typically redirected to an intranet page maintained by the administrator.

**Minimum client build**

Minimum allowable build number for the various client applications including the Command-Line Client 506, the GUI Client 496, the Tray Notifier 613, and IDE Clients 524.

Use this to ensure your clients are reasonably up to date. This is especially important if there is a feature or bug-fix you know is necessary for your system.

The help text in the GUI will identify the oldest stable, compatible build number and will list the most recent known build number for your reference.

**Login ticket time-to-live**

Login tickets are special alpha-numeric identifiers that act as user credentials for a limited time period. This setting defines how long a login ticket should remain valid. A value of '0' indicates that the ticket remains valid for 30 days.

**Clear login cookie on session termination**

If enabled, login cookies will be stored in browser session and will be cleared when the browser is closed. Otherwise, login cookies will be stored in browser cache.

In either cases, users will be logged out from server and license will be released after 60 minutes of inactivity or after pressing logout button.

**Secure authentication cookies**

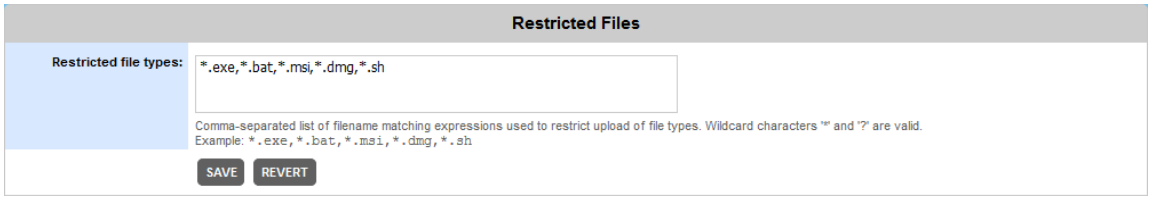
Specifies how to handle login cookies over HTTPS and HTTP connections.

- If server uses HTTPS and the setting is enabled, cookies will be sent over HTTPS, otherwise they will be sent over HTTP.
- If server uses HTTP and the setting is enabled, cookies will be cleared when the browser is closed, otherwise they will be kept in browser cache.

In all cases, users will be logged out from server and license will be released after 60 minutes of inactivity or after pressing logout button.

## File Types Tab

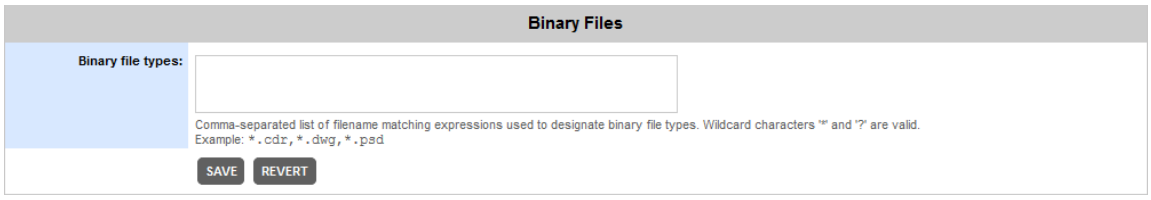
This tab controls how Collaborator server should treat certain types of files: images, binary, executables.




**Restricted file types**

Specifies which files cannot be uploaded to reviews (to avoid malicious file uploads). By default Collaborator blocks the following file types: executable files (.exe), batch files (.bat), Windows Installer files (.msi), Mac OS disk images (.dmg) and Unix script files (.sh).

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.




**Binary file types**

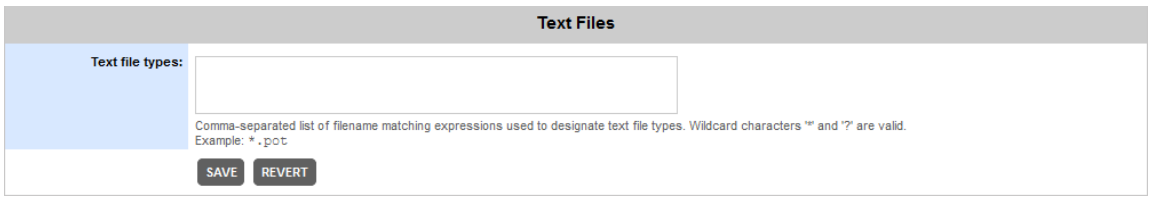
Binary files attached to reviews are not displayed in the Diff Viewer 377, and instead must be opened by external applications. Here you can configure which files are to be treated as binary.

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.


**Image file types**

Image files are reviewed in the browser in the Diff Viewer using a special image Diff Viewer 410. Here you can configure which files are treated as images. By default, GIF, JPEG, and PNG images are handled as images. The content of the files must be renderable by the users' browsers for images to actually be reviewable. Images that require special software to view should be treated as Binary Files 197 so they can be rendered by that software.

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.



**Text file types**

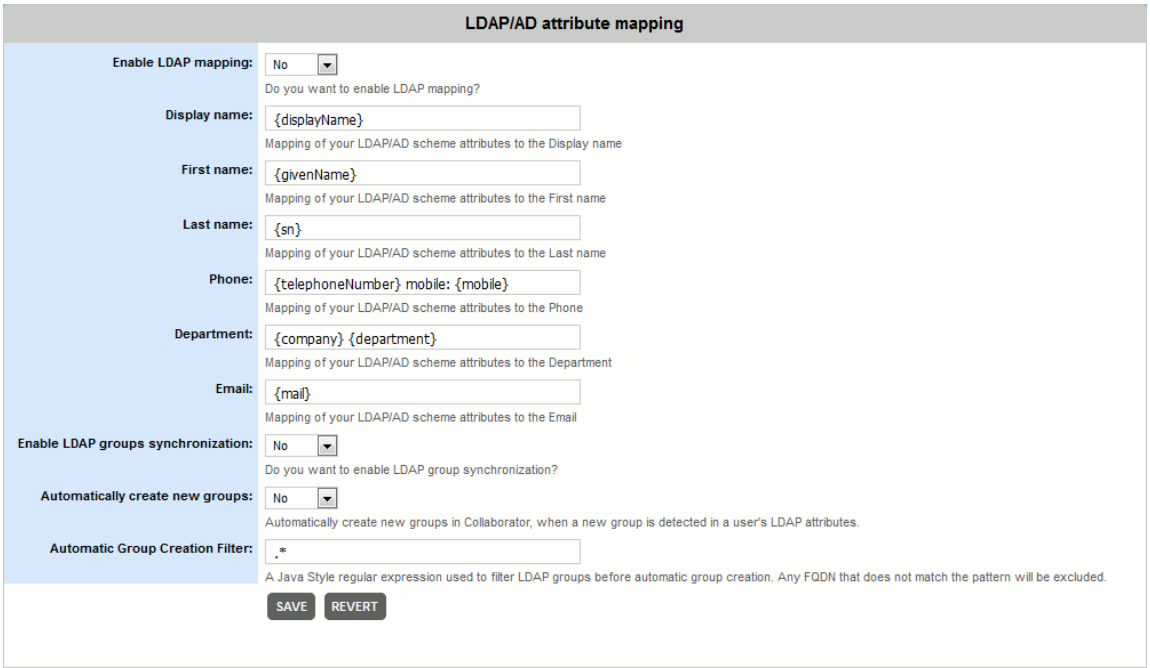
Diff Viewer detects type of review materials[363] based on the extension of the uploaded files. For example, `.java` files typically stand for Java source code, `.rtf` and `.doc` are typically word-processing documents and so on. However, some extensions could stand for multiple types of data and this could mislead the Diff Viewer. For instance, the `.pot` extension could be either a PowerPoint template file or a portable object file. In this case, you can use this setting to specify explicitly which file types should be treated as text-based files.

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.

## LDAP Settings Tab

This tab is visible only if LDAP or Active Directory authentication[119] is enabled.

This tab specifies whether Collaborator should retrieve user settings and group members from the LDAP or Active Directory and defines the mapping between Collaborator user settings and LDAP/AD scheme attributes. For LDAP directories, mapping users requires additional configuration, as described in LDAP or Active Directory Authentication[126].

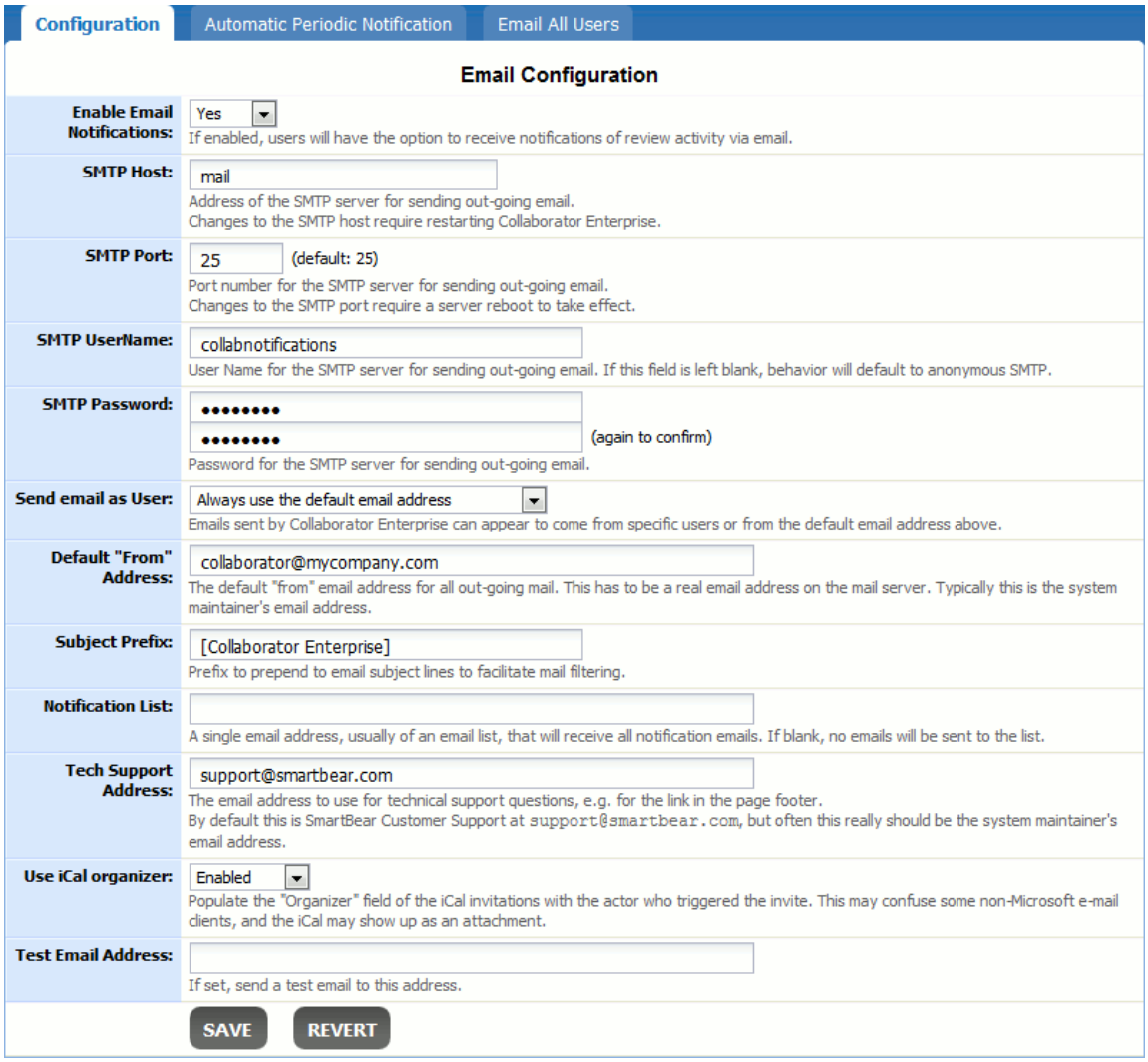


| | |
|---|---|
| **Enable LDAP mapping** | Specifies whether Collaborator should retrieve user settings from the LDAP or Active Directory?<br><br>Once enabled, Collaborator will automatically populate user settings when a user logs into the server. |
| **Display name**<br>**First name**<br>**Last name**<br>**Phone**<br>**Department**<br>**Email** | Defines the mapping between the respective user settings 317 and the LDAP or Active Directory scheme attributes.<br><br>May contain attribute placeholders (attribute names enclosed in curly braces) and plain text: "{telephoneNumber} mobile: {mobile}". Placeholders will be replaced with the matching attribute values for the particular user: "+1987456 mobile: +28467913".<br><br>If the attribute does not exist, or its value is not specified for some particular user, the placeholder will be replaced with blank value: "+1987456 mobile: ".<br><br>To learn about possible LDAP and Active Directory attributes, see this comprehensive list |
| **Enable LDAP groups synchronization** | Specifies whether Collaborator should synchronize its native groups with group information from the LDAP or Active Directory?<br><br>Once enabled, Collaborator will check user membership in groups in the LDAP/Active Directory and automatically add this user to the corresponding groups on the Collaborator server. If a group with the specified name does not exist, Collaborator will act according to the "Automatically create new groups" setting below.<br><br>See Syncing Groups 239 for details. |
| **Automatically create new groups** | Specifies whether to create new groups on the Collaborator server when a group with the specified name does not exist. If enabled, Collaborator will create new group, if disabled Collaborator will only synchronize membership of existing groups. |
| **Automatic group creation filter** | A Java-style regular expression used to filter LDAP groups before automatic group creation. Any FQDN that does not match the pattern will be excluded. |

## 3.3.2 Email

This category holds SMTP server settings and other settings related to out-going notifications of review activity.

In order to establish e-mail, poke and calendar notifications for Collaborator users, administrator must properly configure the SMTP server settings below. Otherwise, notifications cannot be sent.




**Enable Email Notifications**

Email notifications are now optional. If enabled, users will have the option to receive notifications of review activity via email.

**SMTP Host**

The domain name or IP address of the server to use to send SMTP messages. This machine must be configured to accept mail from the server on which Collaborator is installed.

If your email server is Exchange you will need to get your Exchange administrator to enable anonymous SMTP.

If this setting is not established properly, no emails will be sent.

**SMTP Port**

The port number to connect on when sending SMTP messages, typically 25. The port number might be different if a spam filter is set up in front of the primary server; in this case you probably want to bypass the spam machine, or else put a rule on the spam machine to allow Collaborator to send messages.

If this setting is not established properly, no emails will be sent.

**SMTP UserName** and **SMTP Password**

User name and password to use for SMTP servers that require authentication. If this field is left blank, behavior will default to anonymous SMTP.

Do not specify these values if your SMTP server uses secure connection but does not require authentication. Otherwise an error will occur.

**Send Email as User**

Determines which user will be listed in the "From Address" in emails sent by Collaborator.

If you select to use the "default address," emails will come from the default address given elsewhere in this configuration screen. Otherwise the system will attempt to send email from the user that caused the message to be sent. For example, if a user causes a "new review" notification to be sent, that user's email address will be used as the "from" address.

Some messages cannot be associated with a particular user and will still be sent from the default address.

**Default "From" Address**

The email address to use in the "From Address" field in notification emails. Typically this is the system maintainer's email address.

**Subject Prefix**

This text will be pre-pended to the subject line of any email sent by Collaborator. This assists end users with mail filtering and helps to train or configure spam-filters to identify Collaborator notifications as non-spam.

**Notification List**

Specifies an e-mail address or mailing list that will receive a copy of *every* notification 329 that Collaborator sends.
This field can be blank.

**Tech Support Address**

The email address to use for technical support questions. For example, this is used in the `[Support]` link in the web page footer.

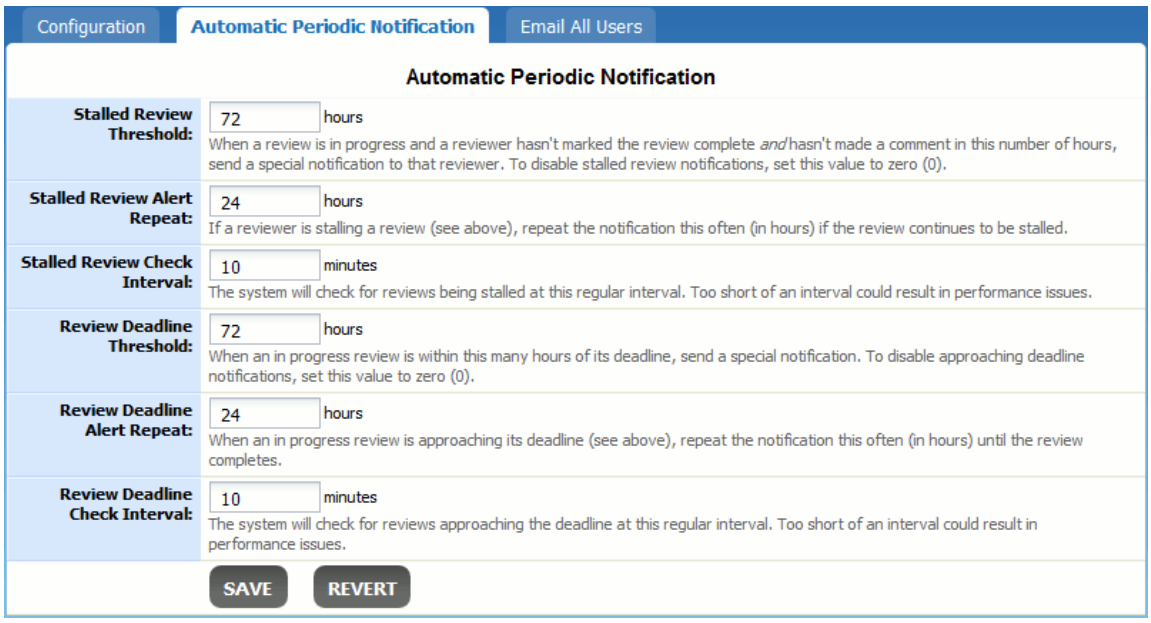
By default this is the email address for Collaborator Technical Support, but often the system maintainer wants to get these emails first for internal resolution.

| | |
|---|---|
| **Use iCal organizer** | Specifies whether to populate the Organizer field in iCalendar invitations[451]. If enabled, the user who has pressed "iCal Everyone" will be set as organizer. Otherwise, the Organizer field will be blank.<br><br>**Important:** Populating the Organizer field may confuse some non-Microsoft e-mail clients, and may result in iCalendar invitations be interpreted as file attachments. |
| **Test E-Mail Address** | Usually left blank; if an address is supplied here a test message will be sent to this address when the form is submitted. This is used to test the email system. The email address is not saved. |

***Note**: If the E-Mail address includes non-Latin symbols, type it inside quotation marks. For example, if you need to use the ädress@example.com address, input it as "ädress"@example.com.*

## Automatic Periodic Notification

On this tab you can set-up email notifications about stalled reviews.



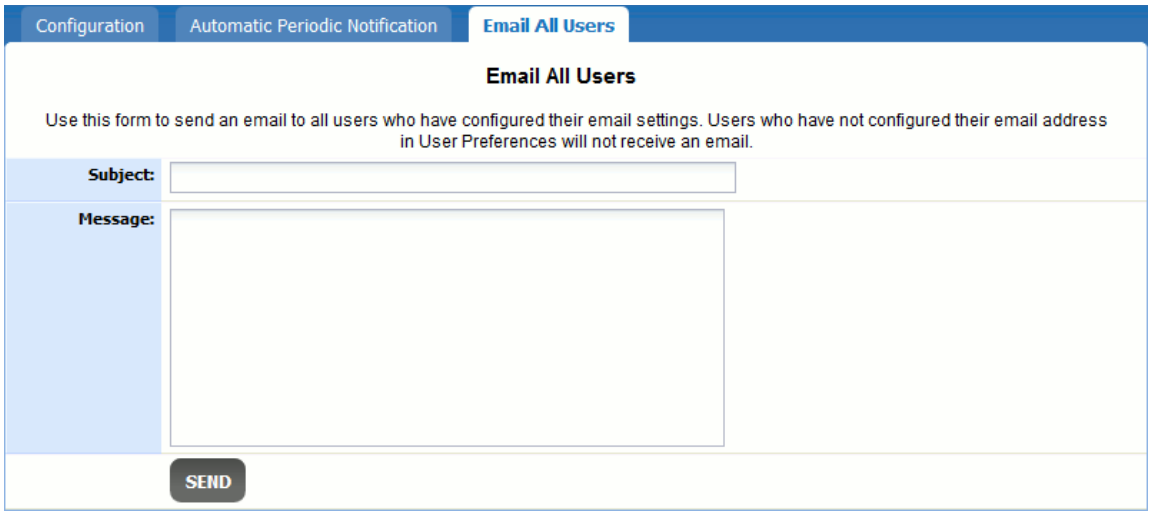

| | |
|---|---|
| **Stalled Review Threshold** | When a review is in progress and a reviewer has not marked the review complete and has not made a comment in this number of hours, send a special notification to that reviewer. To disable stalled review notifications, set this value to zero (0). |

| | |
|---|---|
| **Stalled Review Alert Repeat** | If a reviewer is stalling a review (see above), repeat the notification this often (in hours) if the review continues to be stalled. |
| **Stalled Review Check Interval** | Specifies how often to check for stalled reviews. |
| **Review Deadline Threshold** | When an in progress review is within this many hours of its deadline, send a special notification. To disable approaching deadline notifications, set this value to zero (0). |
| **Review Deadline Alert Repeat** | When an in progress review is approaching its deadline (see above), repeat the notification this often (in hours) until the review completes. |
| **Review Deadline Check Interval** | Specifies how often to check for reviews approaching deadline. |

## Bulk-Email Facility

The Email All Users tab allows the administrator to send an email to all users 219 in the system. This is easier than maintaining a separate email mailing list for users.

A typical use is to broadcast scheduled maintenance of the system, especially before and after a system upgrade 83.



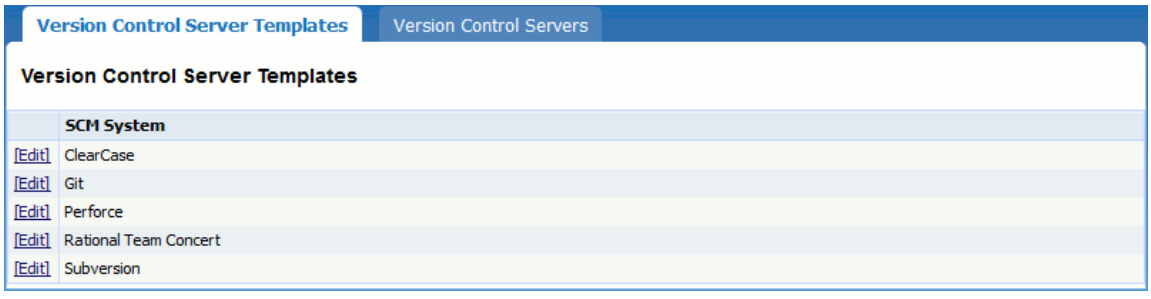

## 3.3.3 Version Control

Collaborator integrates closely with your version control server to select which files to review. The administrator can configure the Collaborator server to be able to connect directly with version control. This allows users to select checked-in changelists to review from the web UI without having to install and configure client-side tools. Users can also configure their connection to the various version control servers using the client tools 485 so they can review changes before committing them.

If you use more than one version control server Collaborator recognizes this and displays the files from the different servers separately.

### Configuring Version Control Server Templates

Each version control system that Collaborator supports for server-side integration has a configurable "template". The configuration in this template is copied in to a version control server entry when you click ***Create*** in the **Configure a new version control server** form, or when a version control server entry is created automatically because a client uploaded files from a server that does not map 206 to any of the existing version control server entries.

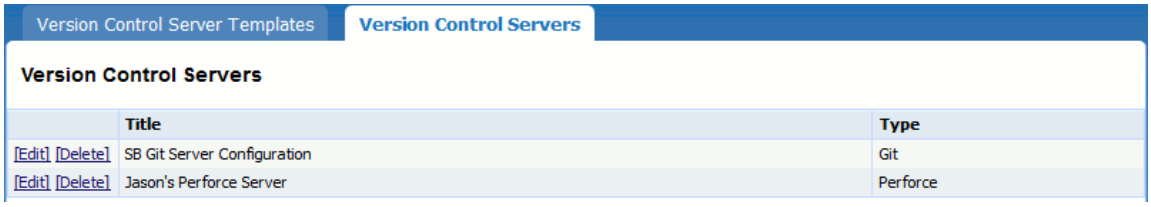



Each version control system that supports server-side version control integration includes different options in the Version Control Server Template:

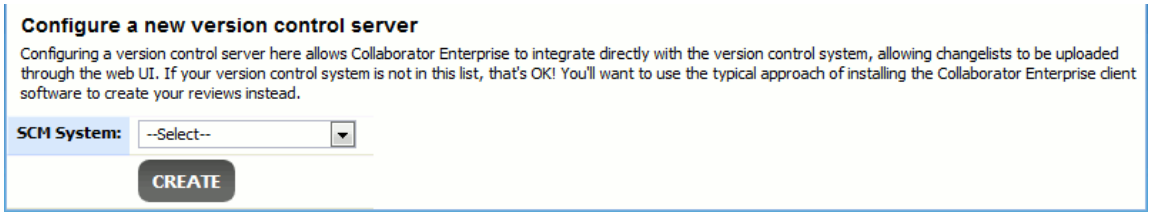
- Git 647
- Perforce 767
- Subversion 801
- ClearCase 668
- Rational Team Concert 701

## Configuring Servers

Next tab lists all currently configured servers, or shows a message if you have not created any yet:



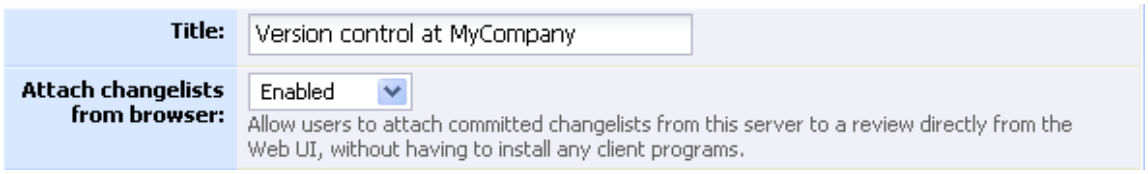

Entries in this table are created automatically when one of the client tools uploads files from a server that does not map 206 to any of the servers we currently know about. You can edit or delete entries in this list. A form to create a new entry always appears at the bottom of the screen:




## Editing Server Configuration

The exact fields you need to configure the connection to a version control server depends on the version control system:

Most of the version control server entries have at a minimum these two fields:



| | |
|---|---|
| **Title** | The title is used to indicate this version control server to other users of Collaborator, so it should be something that everyone will understand, even if they are going through proxies, VPNs, or other such things. |
| **Attach changelists from browser** | If enabled, this feature lets users select committed changelists to review directly from the web browser, without having to install any client programs. |

All of the version control systems also require that you specify the path to the command-line client executable for that system. This is a good example of something you might want to specify in the version control server template 204, since it is probably going to be the same for all the version control server entries you configure.

If you want to enable the Attach changelists from browser 206 feature, then the server will also require some sort of authentication. This is the version control server account that Collaborator should use when querying that version control server. Note that Collaborator will use this account no matter who is logged in!

You can click ***Test Connection*** at the bottom of the form to make sure that Collaborator can successfully communicate with the server. NOTE: This test simply verifies that specified server appears to be a valid repository; it does NOT verify that your username and password can update the repository.

## Client Configuration Mapping

An unfortunate thing happens when client tools are separately configured for version control access: they can have different names for the same physical version control server. It is hard or impossible for the Collaborator server to automatically figure out whether these two names represent the same server or two different version control servers.

The administrator can fix this using the **Client Configuration Mapping** section of the server configuration form. You can configure Java-style regular expressions to match against each of the fields that our client tools upload. If the version control server entry was created automatically then these fields are already filled out and you probably do not need to change them. The fields available are different for each version control system:

When you submit this form, the list of configurations updates to indicate which configurations match and which do not match the current patterns. This makes it easy to iterate until you have covered exactly the right cases.

# 3.3.4 Triggers

Collaborator can execute an external script or application when certain events occur. This is useful when integrating Collaborator with other systems such as issue-tracking, ALM, or builds.

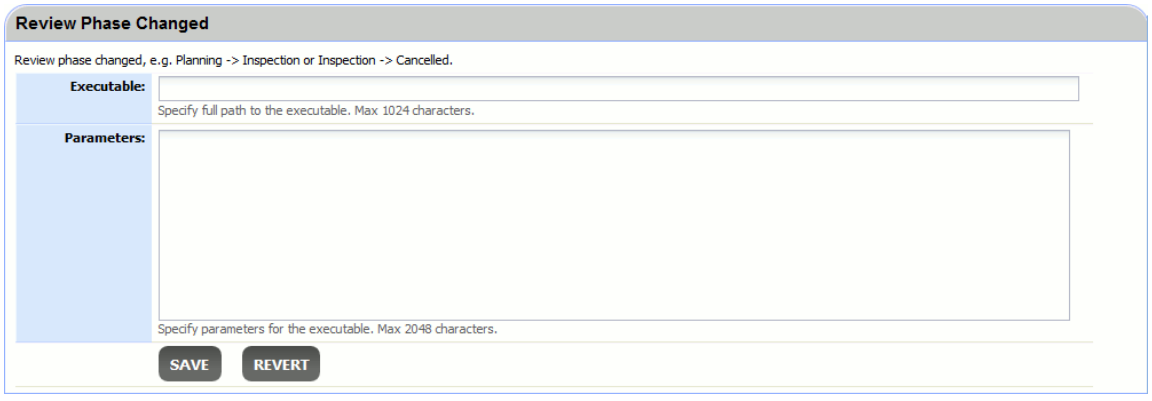
***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

Possible uses of triggers include:

- Mirror defects created in Collaborator into an external issue-tracking system.
- Set a flag in a system when a review completes.
- Intercept all notifications sent from Collaborator to log them or to send the notification in a way other than standard email or RSS.
- Send notifications to a mailing list when certain events occur.

The Triggers section of the Administration page lets you configure the various triggers:

- **Review Created** - Review created. No files or participants yet.
- **Review Phase Changed** - Review phase changed. For instance from "Planning" to "Inspection" or from "Inspection" to "Canceled".
- **Added Files** - Files and/or changelists attached to a review.
- **Notifications Sent** - A email, RSS or other type of notification was sent to someone.
- **Defect Activity** - A defect was added, modified, or marked open/fixed. Use the defect unique ID to determine the difference.
- **User Created** - A new user was created. This runs whether or not the creation was automated or manual.
- **Role Changed** - A participant's role has changed. For instance, a participant was added to a review, removed from a review or promoted from one role to another.




The **`Executable`** field specifies the application to run. If this is a script, use the script executable here (for example, Perl, PHP, Python). This must be an absolute path to the application; the form will complain if the file cannot be found or executed. To specify path on Windows, use double slash. For example: `C:\\Windows\\System32\\cmd.exe`

The **Parameters** field is a list of arguments to supply to the executable. If this is a script, the absolute path to the script itself is typically the first argument.

The **Executable** field could be up to 1024 characters long, and the **Parameters** field could be up to 2048 characters long. However if your character encoding system uses more than 1 byte per character, the file path/parameters could be shorter (to fit into 1024/2048 bytes).

There are many special arguments you can supply that will be replaced by Collaborator when the application is executed. For example, `${review.title}` will be replaced by the title of the review associated with the given triggering event. See the Variable Substitution 161 chapter for details.

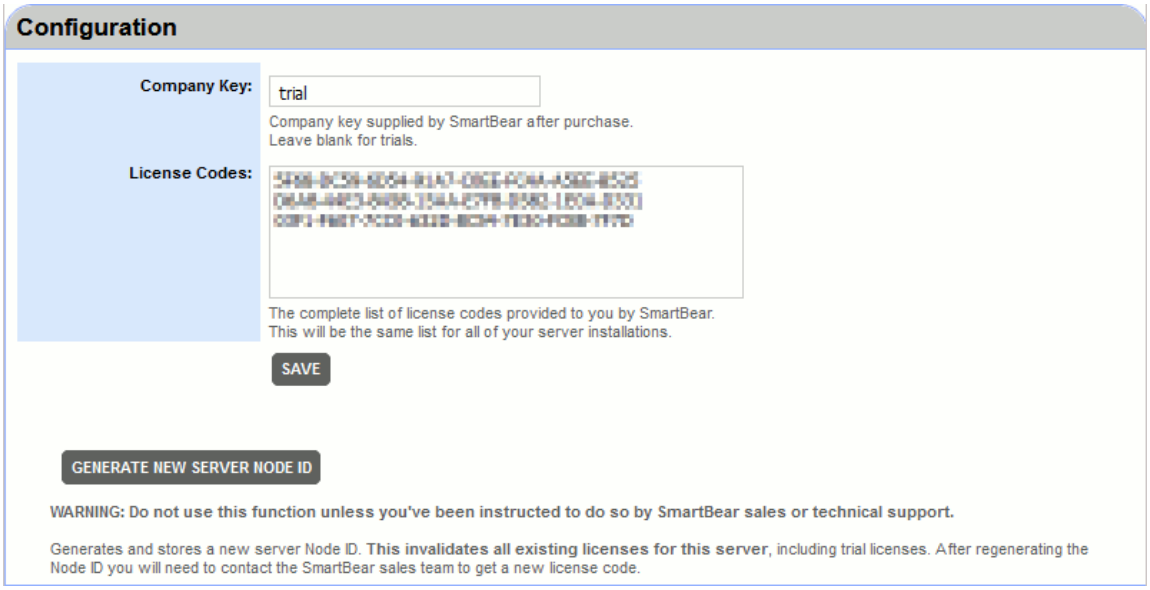
Please note, that not all of the variables are available at the moment the trigger fires. In this case they will return "Not available" instead of their value. For example, the `${review.title}` is not available for the Review Created trigger as the review does not yet has a title when the trigger fires.

## 3.3.5 Licensing

This category holds two tabs, **Collaborator license** tab displays information about your current Collaborator license, while the **Simulink integration** tab displays information about additional licenses required for reviewing Simulink models 404.

The licensing is done by the administrator on the Collaborator server. Collaborator client software doesn't have to be licensed.

### Collaborator License




In **Configuration** section you can configure your Collaborator license. See this topic 94 for instructions on how to configure your license. You should specify the following data:

**Company Key** The company key supplied by SmartBear after purchasing the license.

The company key is "trial" for temporary licenses and a word or phrase for permanent ones.

**License Codes** The list of 32-character license codes for the Collaborator server provided by SmartBear.

If you have the license set up, you can use the **Update from SmartBear** button to request a license update directly from the SmartBear license server.

In order for this feature to work, your computer must have a functioning Internet connection, and proxies and firewalls in your network should allow connection to the Collaborator license server (URL http://licensing.codecollaborator.com, port 80).

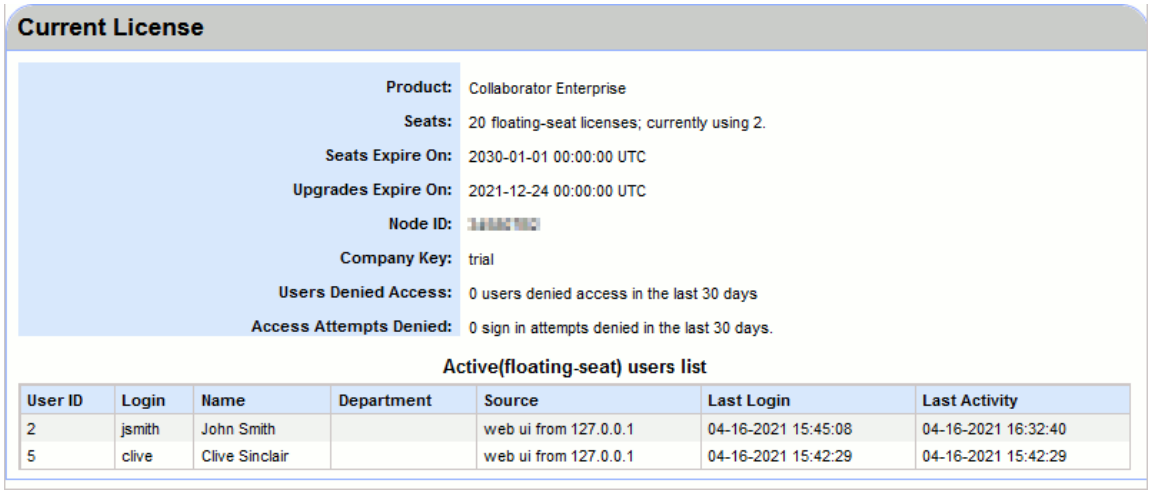
Once you are finished editing the license configuration, select **Save** to save changes.

## Generate New Server Node ID

To generate a new server Node ID, you can use the **Generate New Server Node ID** button.

> Do not use this function unless you've been instructed to do so by SmartBear sales or technical support. It invalidates all the existing licenses for the server, including trial licenses. After generating a new Node ID, you will need to contact the SmartBear sales team to get a new license code.

The **Current License** section displays information about your current Collaborator license.




**Product** The Collaborator edition 3 the license applies to.

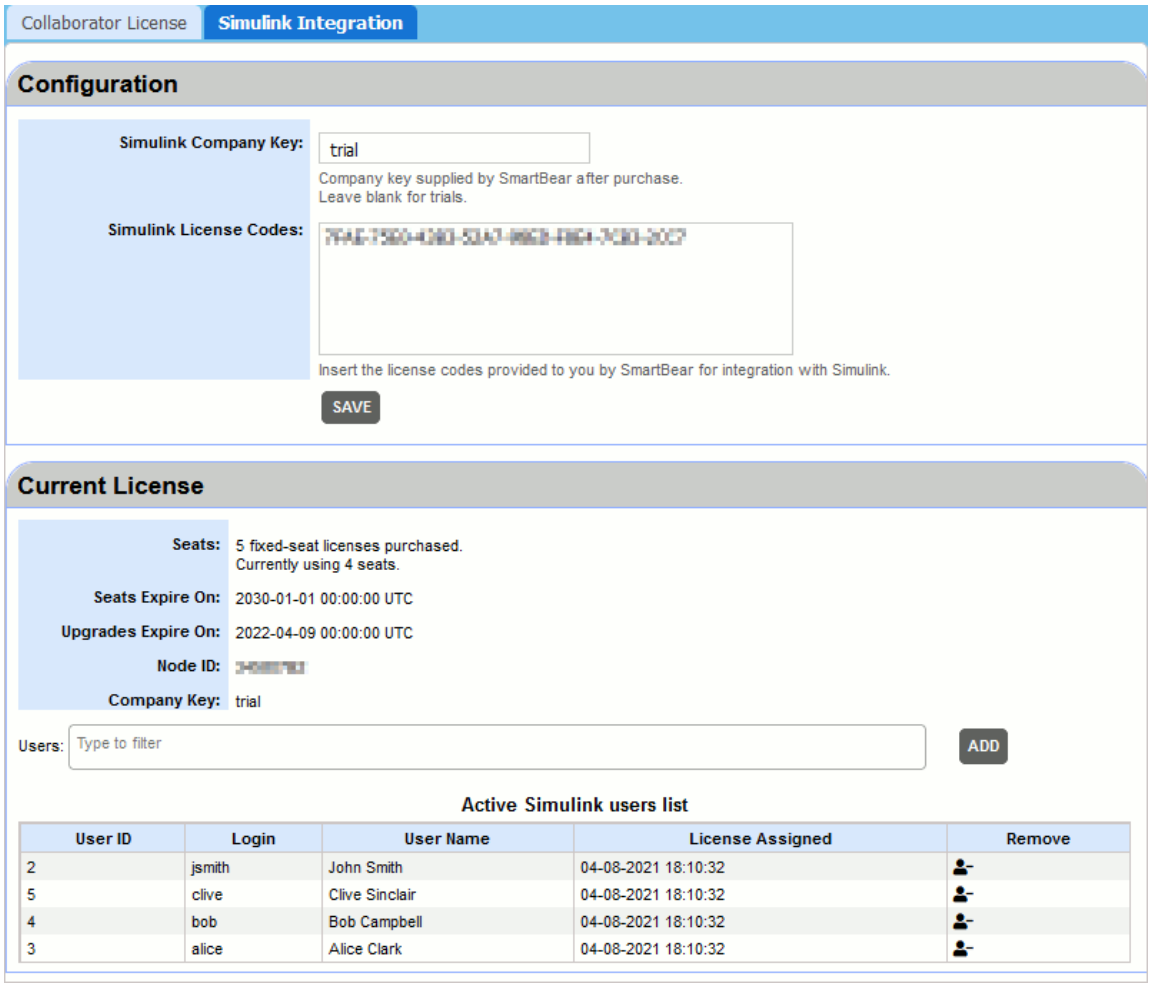
| | |
|---|---|
| **Seats** | The number and type of seats available under the license and the number of seats currently consumed.<br><br>SmartBear offers fixed-seat and floating-seat licenses for Collaborator. See Licensing 90 for more information on the types of licenses and license consumption. |
| **Seats Expire On** | The date and time when the license expires. |
| **Upgrades Expire On** | The date and time when license upgrades expire. |
| **Node ID** | A unique identifier of your Collaborator server. All the licenses issued for the server are linked to the Node ID.<br><br>The Node ID is bound to the database, and not the server hardware or operating system. As long as you are connecting to the same database, the Node ID will not change, and the license will remain valid. During server migration 99, the server's Node ID is carried over, and so are all the licenses linked to it. |
| **Company Key** | The company key supplied by SmartBear after purchasing the license.<br><br>The company key is "trial" for temporary licenses and a word or phrase for permanent ones. |
| **Users Denied Access** | The number of login attempts failed due to licensing reasons 92 in the last 30 days.<br><br>The number is not incremented if a login attempt fails due to the user having been disabled 223 or entering incorrect login or password. |

Additionally, this section can display a list of active floating-seat users. To display it, you need to enable the Show active(floating-seat) users list in the license page 185 setting.

Once enabled, it will display information about floating-seat users who are currently logged-in: their ID, username, source of license consumption, as well as date and time of last login and last activity.

## Simulink integration

In order to review Simulink models 404, you should have Collaborator Enterprise edition license and also obtain additional Simulink integration licenses. Simulink integration licenses are assigned to particular users.




In **Configuration** section you should specify the following data:

**Company Key** The company key supplied by SmartBear after purchasing the Simulink integration license.

The company key is "trial" for temporary licenses and a word or phrase for permanent ones.

**License Codes** The list of 32-character license codes for the Collaborator server provided by SmartBear.

Once you are finished editing the license configuration, select **Save** to save changes.

The **Current License** section displays information about your current Simulink integration licenses and allows to assign licenses to particular users.

To assign Simulink integration licenses to users, specify their login names and press **Add**.

The **Active Simulink users list** displays users who currently have Simulink integration licenses assigned. To revoke a license from a user, click icon in the user list.

Simulink integration licenses can be re-assigned to another user only once in 24 hours.

### 3.3.6 System Status

Collaborator system status reports can be found on the System Status page.

## Graphs

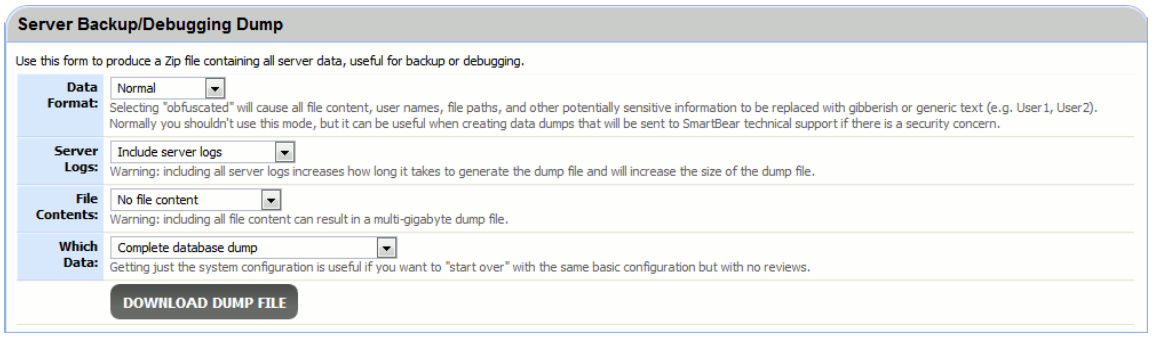
The "Peak Usage" chart plots the maximum number of simultaneous users, measured by day. The "Active Users" chart displays the total number of users during the day:


This is most useful when determining the number of floating seats necessary to handle your Collaborator traffic. In the example above, 5 floating seats would be sufficient.

To learn more about Collaborator licenses: difference between fixed-seat and floating seat licenses, how Collaborator server counts license consumption and so on, see Collaborator Licensing 90.

## Server Backup/Debugging Dump

This tab allows you to create and download a whole system dump:




It contains the following settings:

1. **Data Format:** Normal - Selecting "Normal" will leave the data as is.

Obfuscated - Selecting "Obfuscated" will change the data to conceal sensitive information.

2. **Server Logs:** Include server logs - Selecting this will include server logs in dump. Note: If you are creating a dump file to send to technical support, it is important to include the server logs.

do not include server logs - Selecting this will not include server logs in the dump file.

3. **File Contents:** No file content - Selecting "No file content" will not include file content.

Include all file content - Selecting this will include all file content, but this will also cause a longer download.

4. **Which Data:** Complete database dump - This must be selected for a whole system dump.

System configuration only - no review data - Selecting this will only save the system settings.

Once you have filled out the form, click the "**Download Dump File**" button to download the database dump file. This is a ZIP file containing all your database data in a platform- and database-independent XML format, plus additional files that describes your server environment.

To learn more about creating system dumps, see Server Backup and Migration 99.

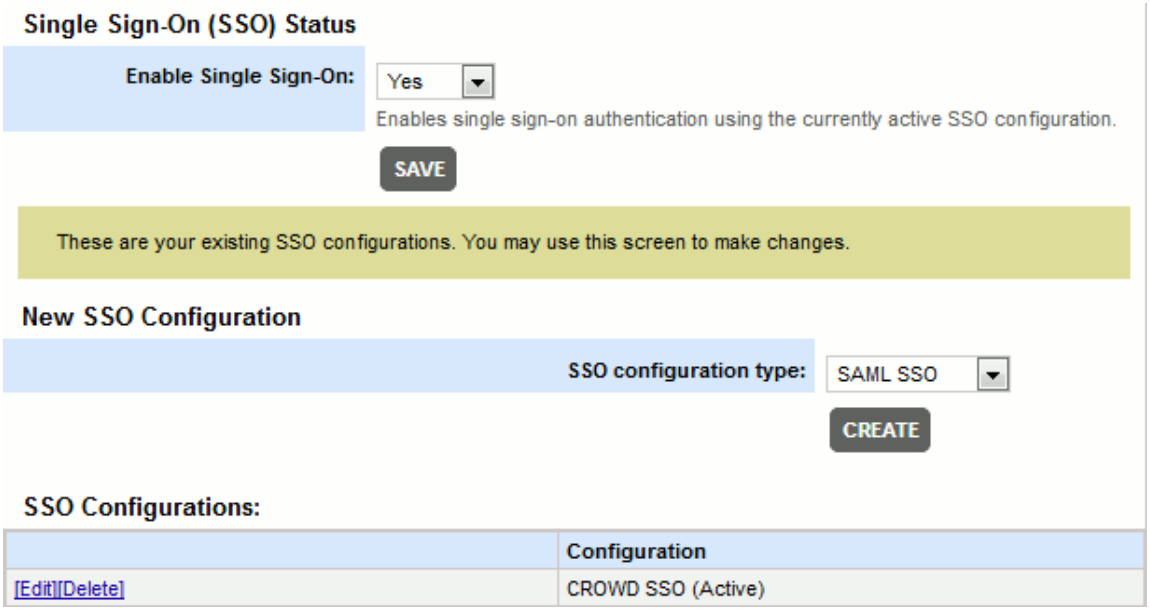
## System Information

Other tabs of System Status page contain multiple server parameters which can be useful when debugging problems. These tabs will be mainly used to gather system information for our Technical Support when trouble should arise.

## 3.3.7 Single Sign-On

These settings allow to configure integration between Collaborator and single sign-on authentication services 130.

***Note:*** *Single sign-on authentication is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.




The **Single Sign-On (SSO) Status** section indicates current state of single sign-on integration and allows to enable or disable pre-configured SSO configurations.

If you experience troubles with single sign-on authentication and cannot login to your Collaborator server (wrong redirect URLs, cannot login as admin and so on), you can also disable single sign-on authentication via the `-Dcom.smartbear.server.sso.disable=true` Java VM option 1232.

The **New SSO Configuration** section allows creating new SAML or Crowd OpenID configurations. You can have only one configuration of each type. On the **SSO Configurations** section you can edit or delete your existing SSO configurations.

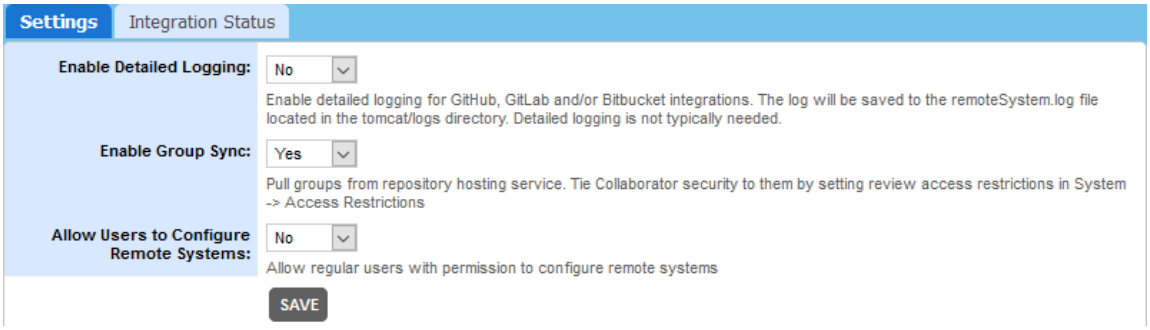
Check the following sections to learn how to configure single sign-on services:

- Configuring SSO via SAML 132
- Configuring SSO via Crowd OpenID 145

### 3.3.8 Remote System Integrations

This section hosts two tabs **Settings** and **Integration Status**.

The **Settings** tab holds a number of global settings affecting remote system integrations:

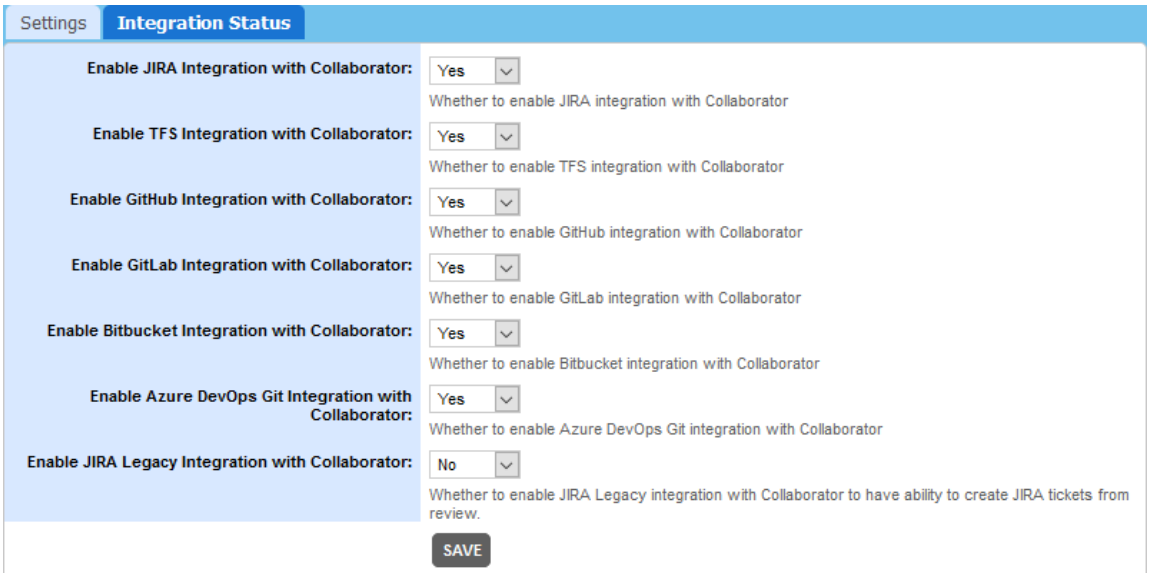



**Enable detailed logging** — If enabled, Collaborator will generate logs of remote system actions and store them to the `<Collaborator Server>/tomcat/logs/remoteSystem.log` file.

**Enable group sync** — Enables system-wide ability to synchronize groups and members 239 with remote repository hosting services. To enable or disable synchronization with particular service, use the **Sync groups** setting of that particular remote repository configuration 217.

**Allow users to configure remote systems** — If enabled, this section and its subsections would be available to regular users having the appropriate permissions 225. Thus regular users could create and modify remote system integrations themselves.

The **Integration Status** tab displays which of remote system integrations are currently operating and allows to enable or disable pre-configured integrations:

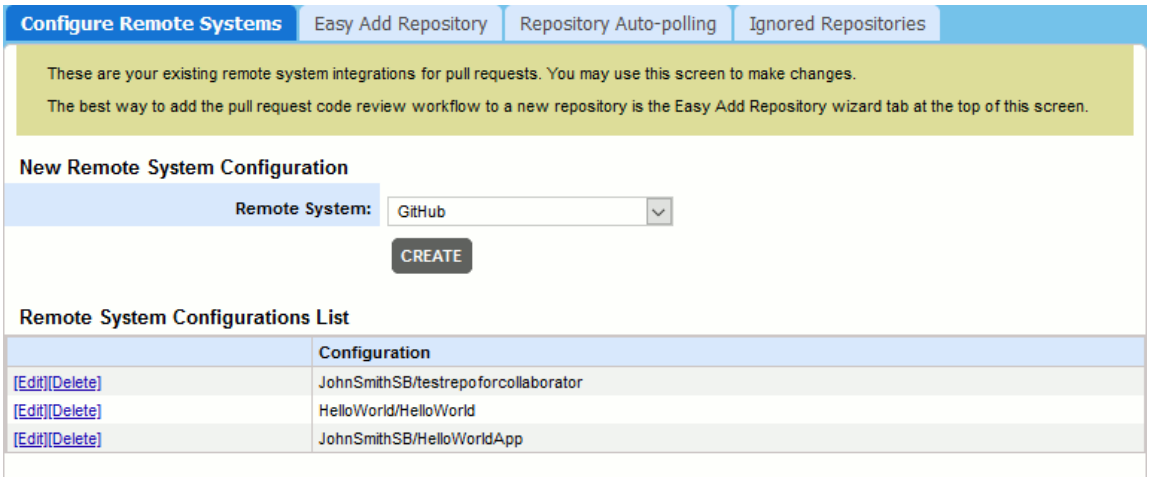



To create new integrations or configure existing integrations, switch to the Repository Hosting Services 217 or Issue-Tracking Services 219 setting pages.

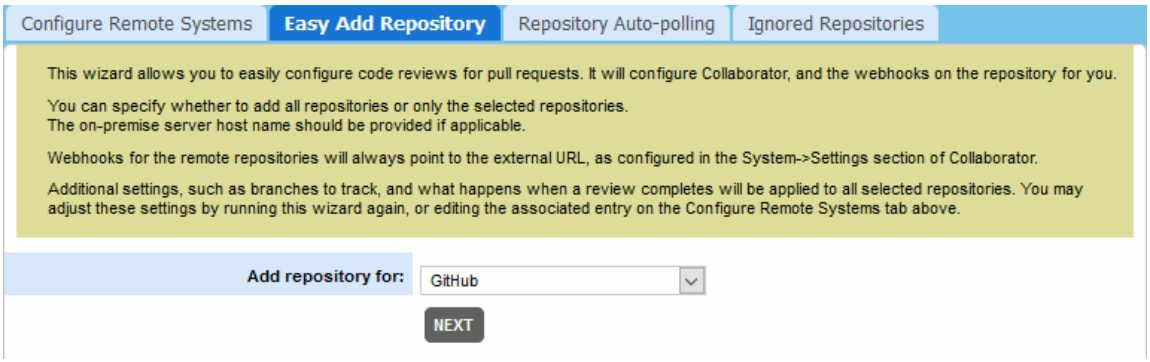
### 3.3.8.1 Repository Hosting Services

These settings allow to configure integration between Collaborator and the remote repository hosting services like GitHub, Bitbucket, Bitbucket Server, GitLab, or Azure DevOps 826.

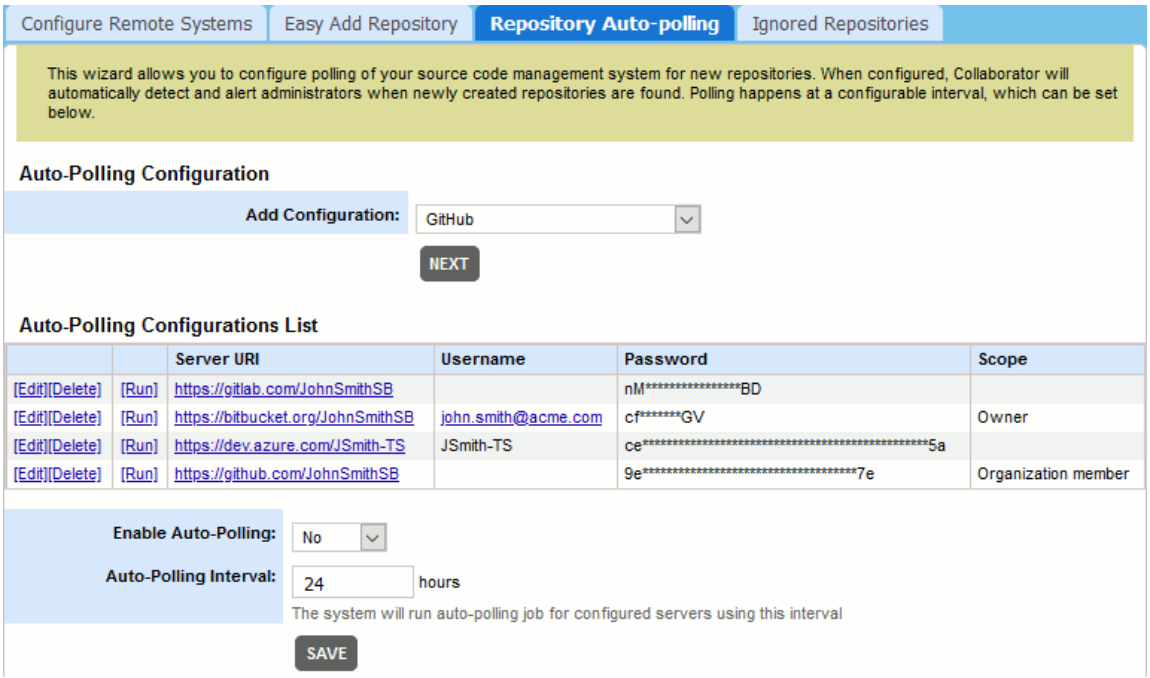
On the **Configure Remote Systems** tab you can edit your existing integrations. You can have multiple configurations for remote repository services.



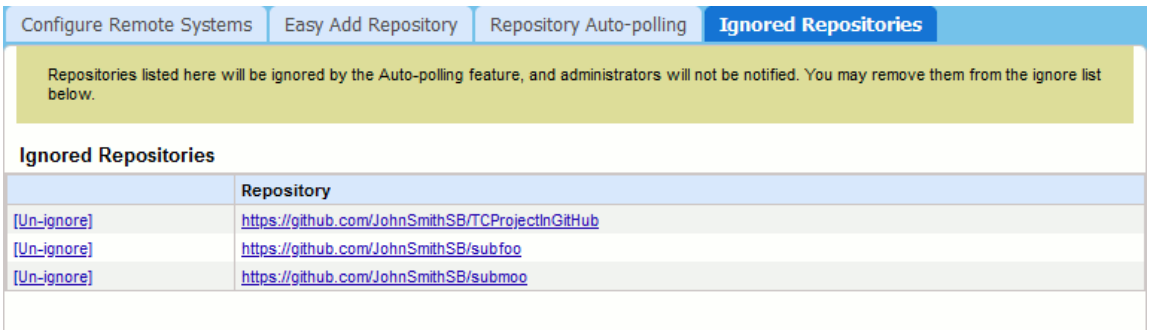

To create a new configuration for a remote system, use either the **Easy Add Repository** tab or the New Remote System Configuration section of the **Configure Remote Systems** tab.




Alternatively, you may configure Collaborator server to automatically check if any new repositories were found on the specified remote hosting servers and notify administrators suggesting to create integrations with these newly created repositories. This can be performed on the **Repository Auto-Polling** tab.




The **Ignored Repositories** tab lists the repositories that should not be affected by the auto-polling feature, so that Collaborator will not suggest creating integrations for those repositories.



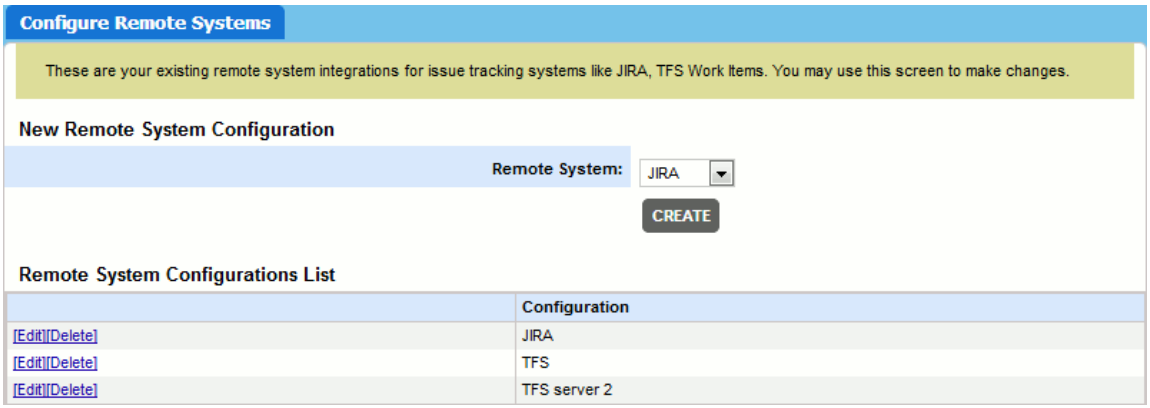
Read the following sections to learn how to create and configure remote system integrations:

- Configuring GitHub Integration 841
- Configuring Bitbucket Cloud Integration 850
- Configuring Bitbucket Server Integration 859
- Configuring GitLab Integration 866
- Configuring Azure DevOps Integration 875

#### 3.3.8.2 Issue-Tracking Services

These settings allow to configure integration between Collaborator and remote issue-tracking services like JIRA and TFS 887.

On the **Configure Remote Systems** tab you can create new configurations and edit your existing integration configurations. You can have multiple configurations for multiple issue-tracking services.




Read the following sections to learn how to create and configure remote system integrations:

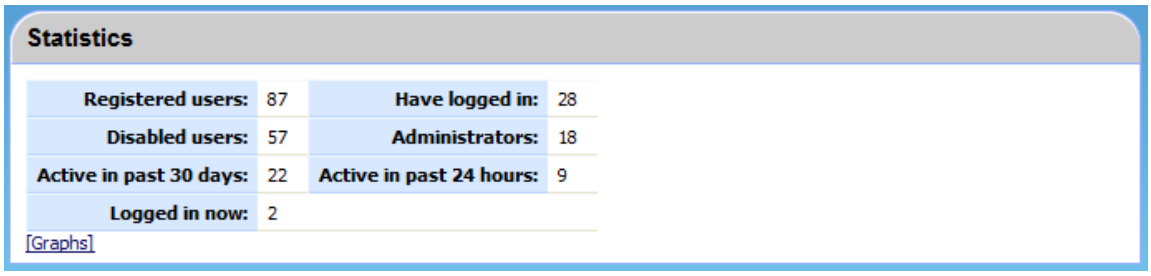
- Configuring JIRA Integration 897
- Configuring TFS Work Item Integration 903

### 3.3.9 Users

The user management page lets you add, disable, and manage user accounts.

## Statistics

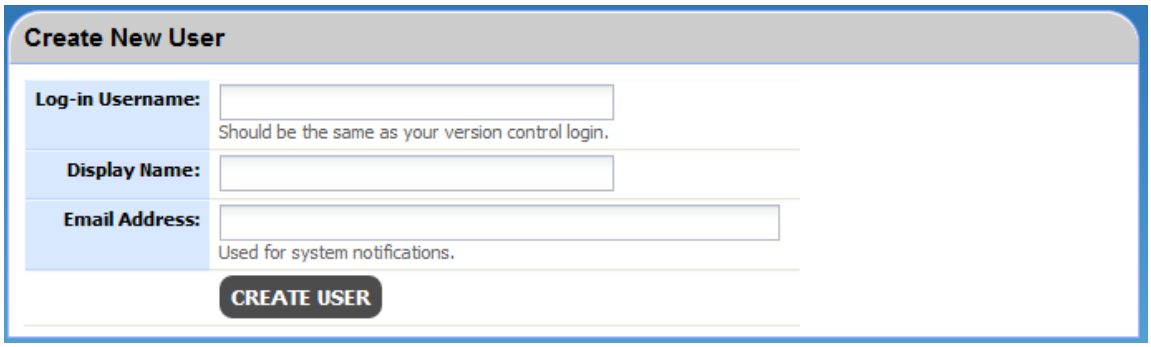
The statistics tab shows information about users and usage of Collaborator:




This can help monitor your usage and determine how many seats are necessary.

## Creating New Users

This form lets you create new users in Collaborator. If you are using LDAP authentication[119] you will not see this form because users are created automatically when they first log into the system.




Login names must be unique in the system; Collaborator will not allow you to create a second user with the same login. The display name is used to denote the user in the interface. Typically, display name contains the user's full name, but may include any other information: initials, nickname, position, department, and so on. Email address is used for sending notification messages.

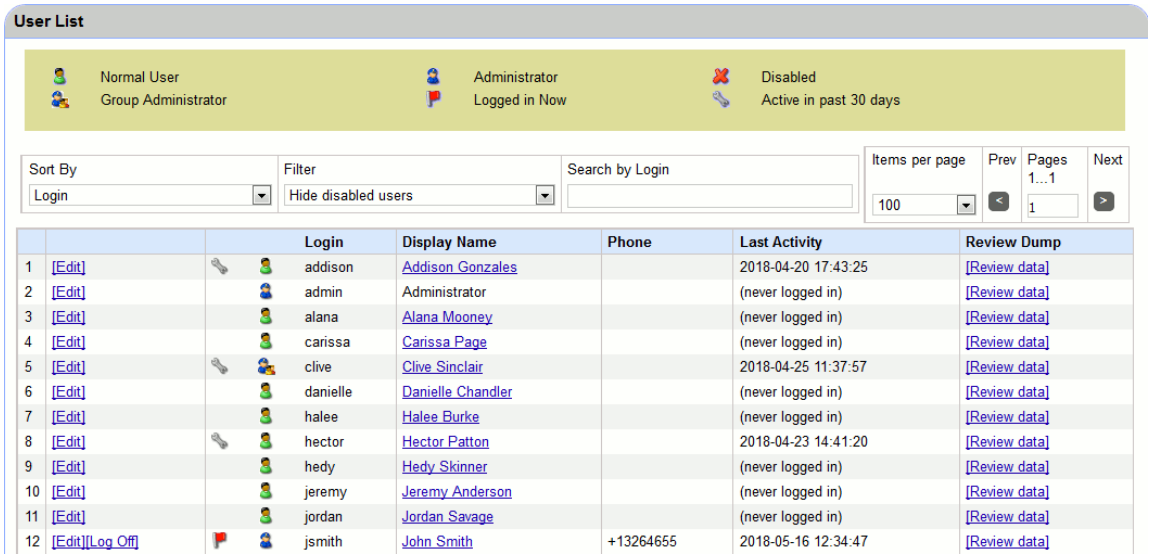
***Note**: If the E-Mail address includes non-latin symbols, type it inside quotation marks. For example, if you need to use the ädress@example.com address, input it as "ädress"@example.com.*

**Important:** When creating new user accounts, it is a good idea to make the login name identical to the user's version control login. One advantage is that the user does not have to remember a new login. Another advantage is that certain features are enabled in Collaborator because it can correlate users from version control changelists with Collaborator users. Finally cross-system reporting becomes much easier when the logins are the same.

After you create a user you are returned to the same screen. This makes it easy to add multiple users quickly.

## The User List

The user list shows every user in the system along with key information about each:




Here, the **Last Activity** column indicates the last time the browser contacted the Collaborator server. The browser contacts the Collaborator server every 5 minutes to see if any new reviews or comments have been added that the user needs to look at. Additionally, the browser contacts the Collaborator server every time the user takes action within the Collaborator web UI.

You can sort the user list by:

- Login
- Display Name
- Last Activity (oldest first)
- Last Activity (most recent first)

You can filter the user list by:

- Show all users
- Hide disabled users
- Show only logged-in users
- Show only administrators
- Show users missing name
- Show users missing email
- Show users missing phone number

Also, you can search for user logins that contain the specified letters.

When your server has multiple users, the user list is divided into several pages. The controls above and below the user list allow to select how many items will be displayed per page and navigate to next and previous pages.

The `[Edit]` link allows you to update any user setting 317 from basic contact information to personal display and notification settings. The `[Log Off]` link allows you to immediately log off a logged-in user 223. The [Review data] link allows to generate a dump with reviews of this particular user 175.

The first column of icons tells you several things about the user's recent activity and how the user is counting towards server licensing 90.

| Icon | Logged in? | Consuming floating-seat license? | Consuming fixed-seat license? |
|---|---|---|---|
|  | Yes | Yes | Yes |
|  | No | No | Yes |
| *(none)* | No | No | No |

The second column of icons tells you about the user's access rights:

| Icon | Description |
|---|---|
|  | User with full administrator privileges |

| | |
|---|---|
|  | User with group administrator 231 privileges |
|  | Regular user |
|  | Disabled user. (Not allowed to log in and don't count in licensing.) |

You will notice that the `admin` account never counts towards licensing. This is one of the special characteristics of the System Administrator Account 224.

## Log User Off

Administrators also have the ability to force a user to log out of Collaborator to make a seat available. To do so, just click the [Log Off] link that will appear to the left of the logged in username. This feature should be used carefully as the user will immediately be logged off. Once a user is forced out, the next time they access Collaborator, they will be required to reenter their username/password to gain access.

## Deleting & Disabling Users

You cannot delete users in Collaborator because the user record is necessary for reporting and for viewing old reviews that they were a part of.

However you can *disable* a user. Disabling a user does several things immediately:

- The user can no longer log in.
- If the user is already logged in, they are automatically and immediately logged off.
- The user does not show up in standard user lists, like in list for inviting to a review.
- The user will not receive any notifications or broadcast messages 199.
- The user immediately does not count towards either floating-seat or fixed-seat licensing 90.

The user will continue to appear in the general user list at the end, and an administrator can re-enable a user at any time.

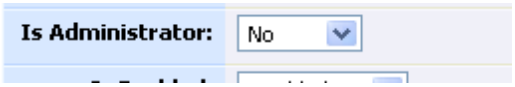
To disable or enable a user, click the `[Edit]` link next to that user on the user list and change the "Is Enabled" flag:

## Administrative Access

Users with administrative access are allowed to access the "Admin" menu from the menubar and all screens therein. They also have a variety of other permissions described elsewhere in this manual, including but not limited to:

- Administrators can look at *all* reviews, regardless of whether they have been marked restricted 336.
- Administrators can do anything with defects in *any* review: create, delete, edit, externalize, and mark fixed, regardless of their role or even non-participation in that review.
- Administrators can edit characteristics of *any* review including changing custom field content, changing the list of participants, and canceling the review.

To set or clear a user's administrative access, click the [Edit] link next to that user on the user list and change the "Is Administrator" flag:




> **Warning:** Be careful if you are an administrator and you are editing your own user information. If you say that you are not an administrator, you are not from that point forward!
> If you do this by accident it can be remedied by another administrator, or the System Administrator in particular (see below).

## The Special "System Administrator" Account

One user account is special: The System Administrator Account. How do you know which account is this special one? If you are using internal authentication, the System Administrator Account is always called `admin` (this cannot be changed). If you are using LDAP authentication 119, the account is determined during installation 70.

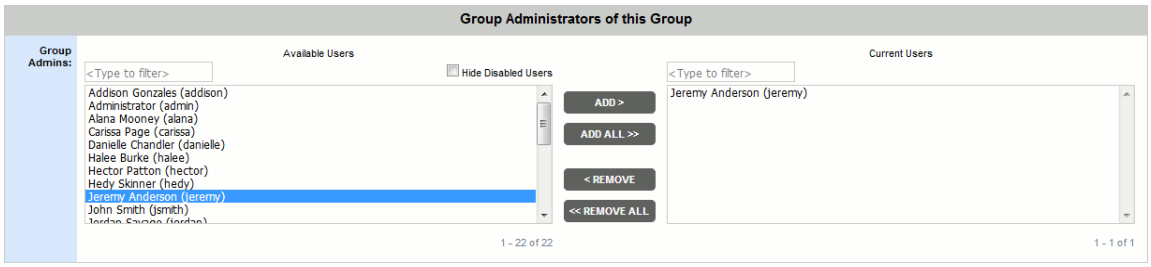
The System Administrator Account has the following special properties:

- There can be only one. One to rule them all.
- The System Administrator is *always* an administrator; administrative access cannot be revoked. (This is prevented in the user interface so you cannot revoke access, even by mistake.)
- The System Administrator can *always* log in. Even if the server license has been exceeded, this account is *still* allowed to log in.

# Advanced User Permissions

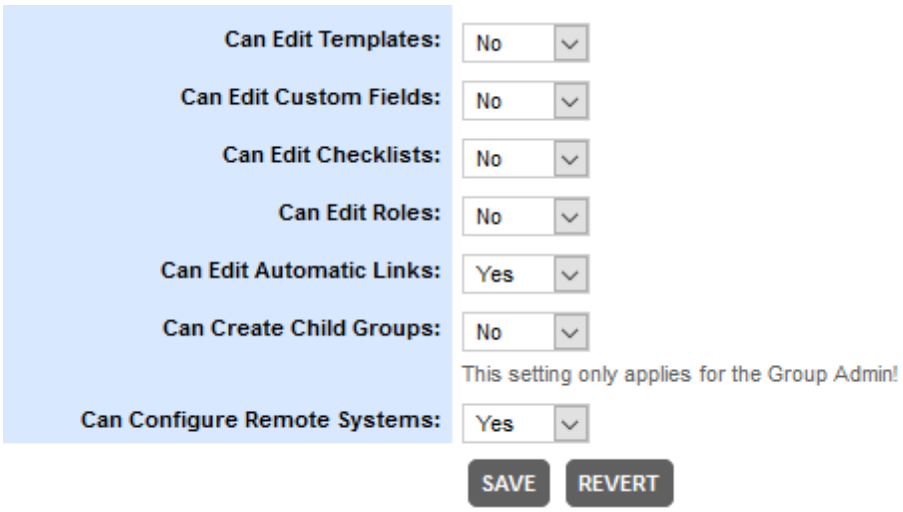
Additionally to giving users full-fledged administrator privileges, you may grant some users with elevated permissions. Users with elevated permissions may perform some of administrative tasks: manage user groups, templates, custom fields, checklists, roles, automatic links.

To assign a user as group administrator 231, open Groups 226 category, click the [Edit] link next to the desired user group, scroll to the Group Administrators section and add the user to the list of group administrators.




Group administrators are regular (not full administrator) users who are allowed to administer this group. Group administrators can edit group settings, add or remove user members 232 of the group.

To set or clear a user's elevated permissions, click the [Edit] link next to that user on the user list, switch to the **Permissions** tab and change the appropriate flags:




Once enabled, the user will become able to modify the respective categories of administrative settings.

## 3.3.10 Groups

User "Groups" is an optional feature that allows you to define groups of users within your organization, so reviews can be associated with a group. Groups can be used to model your organization's hierarchy and/or project assignments. Each group can contain multiple users and can also contain other groups. Each user can be in more than one group. You can specify group administrators who are able to maintain individual groups.

The word "Group" is configurable in the General Settings 180, so you could change it to "Team", or "Project", or whatever else.

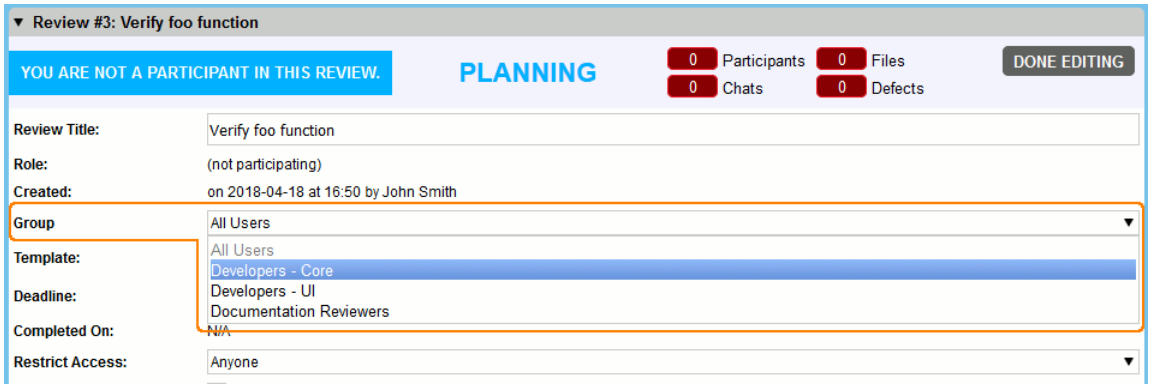
***Note:*** *User groups are only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

In this section:



### Association With Reviews

If a user is a member of a single group that can be associated with a review 230, reviews created by that user are automatically associated with that group. If a user is a member of more than one group that can be associated with a review 230, an additional field is displayed when creating a review:



The list of groups to select from is calculated by first creating a list of all groups that contain the review creator as a member. All groups in that list are then checked to see if they are a child group of any other group(s). Parent groups are added to the list recursively.

In the settings for a group there is an option to disallow associating that group with reviews[230]. This is useful for creating groups that are used for reporting or filtering only. When a group and all its sub-groups cannot be associated with reviews, then this group is not displayed in the Group list. When a group cannot be associated with reviews, but some of its sub-groups can be associated with reviews, then the parent group is displayed in the Group list, but cannot be selected.

Associating a group with a review makes it easier to select participants for that review because the Person list can optionally be filtered by group. The filter is calculated by starting with the users that are members of the selected group. If the selected group also contains child groups, then the users that are members of those child groups will also be added. The search for child groups is done recursively.

## Changing the Group being used in a review

We **do not recommend** changing a group on phases other than Planning phase.

- Changing a group will remove existing participants that do not belong to a new group. If participants belong both to existing group and to new group, they will be retained.
- Changing a group clears the current template, unless the template is available for both groups.
- Changing a group clears currently filled-in checklist items, unless the checklist is available for both groups.

## Restricting Access

Groups can be used to limit access to reviews by setting **Restrict Access to Review Content**[186] in the general settings. When the group access restriction feature is used, the list of users who have access to the review is calculated by starting with the users that are members of the group that is associated with the review. If that group also contains child groups, then the users that are members of those child groups will also be added. The search for child groups is done recursively. Finally, if the group associated with the review is a child of any other group(s) then the users that are members of those parent groups will also be added. The search for parent groups is done recursively.

## Group Hierarchies

It may be difficult to understand how best to use groups based on the rules given above. We have provided a few examples of how they are usually used:

- Using Groups for Organizational Hierarchy[234]

- Using Groups for Projects 235
- Using Groups for Organizational Hierarchy and Projects 237

## Maintaining Groups

Groups can be created and modified in one of two ways: manually (with the web user interface - see the sections below, starting with **Creating New Groups** 228, or with the command-line scripting command ccollab admin group create) or automatically via the command-line scripting ccollab admin group sync command.

If you have only a few groups it is easiest to manage them manually. If you have many groups or if you are mirroring them from an external system like LDAP, it is best to manage them automatically using the "sync" command.

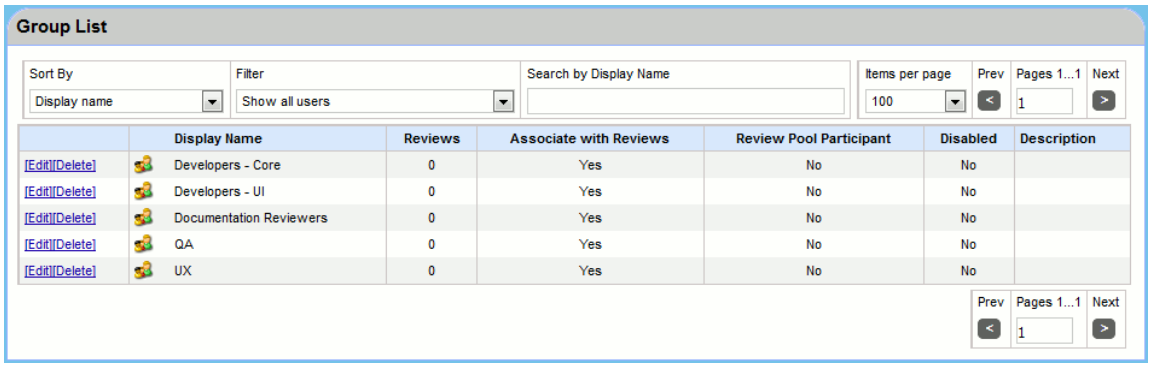
The command-line "sync" command interface is provided to make it easy to mirror group definitions in to Collaborator from an external system. The ccollab admin group sync command uses the contents of an XML file (schema) to update Collaborator's group definitions. See the Syncing Groups 239 topic for more information.

Collaborator tracks which groups were created manually and which were defined via ccollab admin group sync. Groups that were defined manually are not overwritten as a result of using ccollab admin group sync. See the Using Groups for Organizational Hierarchy and Projects 237 example to see how this can be useful.

## Creating New Groups


To create a new user group, first specify its display name, then click **Create Group**.

Once a group is created, the group will appear in the Group List. Group administrators will only see groups listed that they can administer.




You can sort and filter group list, also, you can search for groups that contain the specified letters.

When your server has multiple groups, the Group List is divided into several pages. The controls above and below the list allow to select how many items will be displayed per page and navigate to next and previous pages.

In the Group List, you can edit the configuration details or delete the group by clicking the appropriate link. The Group List also shows the number of reviews associated with the group, whether or not the group can be associated with reviews, whether or not the group is enabled, and any descriptions given to the group.

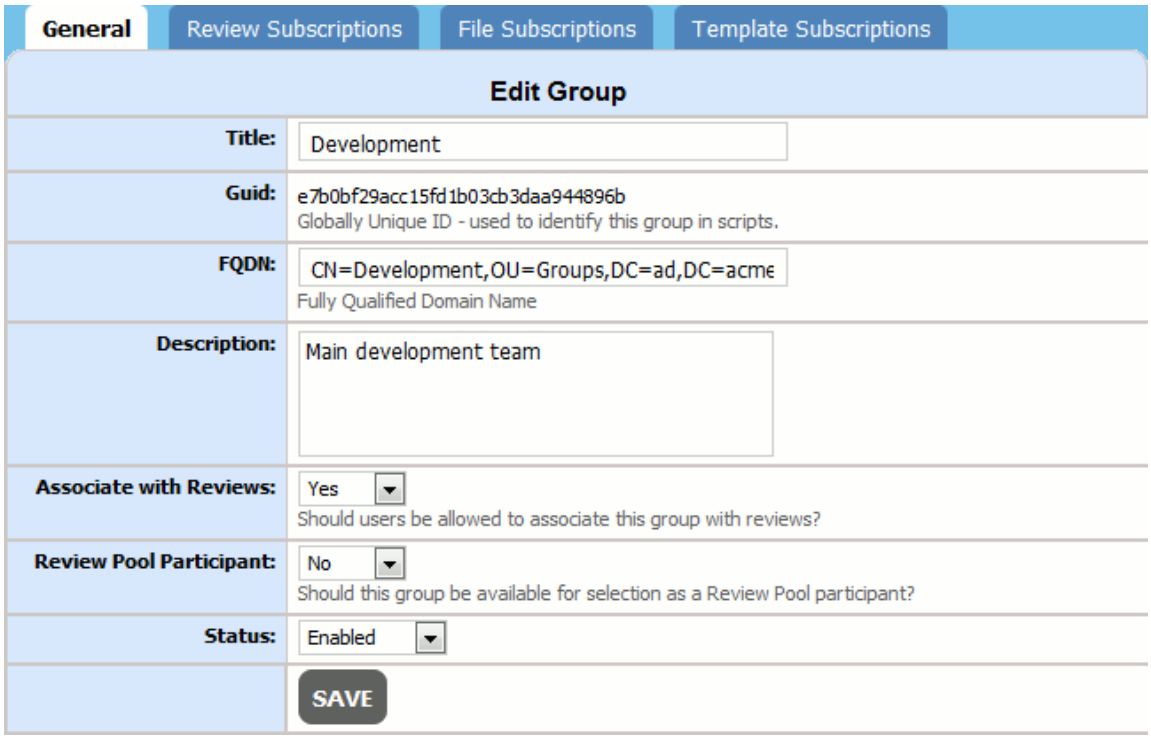
## Deleting a Group

To delete a group, click the [Delete] link next to the group's title in the Group List. You can only delete a group if it is not associated with reviews. If the group is associated with reviews, you must first associate those reviews with a different group before deleting it; alternatively, you could instead disable the group in order to prevent future reviews from being associated with it.

Group administrators are further restricted to only being able to delete groups that they are group administrators of and that have no child groups. This restriction is to prevent group administrators from indirectly creating top-level groups.

## Editing Group Properties

To edit a group, click on the [Edit] link next to the group's title in the Group List. You will be directed to the "Edit Group" page.




**Title** Human-readable title for the group, used in drop-down menus and other UI elements. This does not have to be unique across all groups.

**Guid** Machine-readable ID for group, unique across all groups. If the group was created using the Web Client, this is generated automatically. If the group was created using the sync command, this ID is supplied in the XML file.

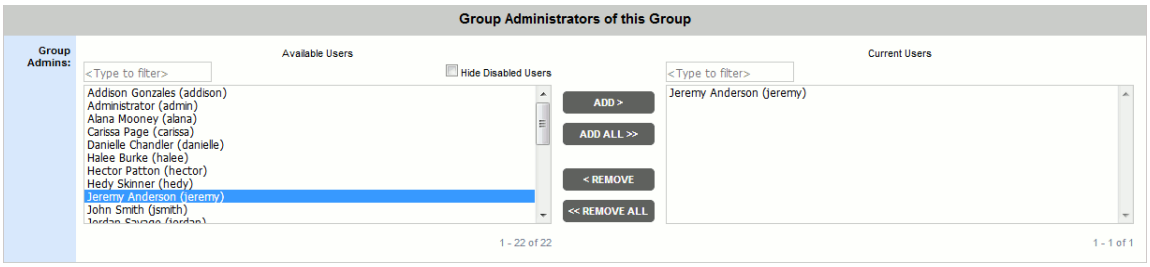
**FQDN** Fully qualified domain name (FQDN) of the group. This name is used when synchronizing groups[126] with LDAP or AD.

**Description** Human-readable description for the group. This is displayed on the "New Review[337]" page when associating a group with a review[226].

**Associate with Reviews** If this is set to "Yes", this group is selectable from the Group drop-down list when associating a group with a review[226]. If "No", this group will be hidden or disabled in the Group list. For an example, see Using Groups for Organizational Hierarchy and Projects[237].

**Review Pool Participant** Specifies whether this group is a Review Pool[969], that is it could be selected as a participant of a review.

**Status** Specifies whether the group is enabled or disabled. Groups can be disabled when they are no longer in use. Disabled groups still show up in reports because reviews may be associated with them.

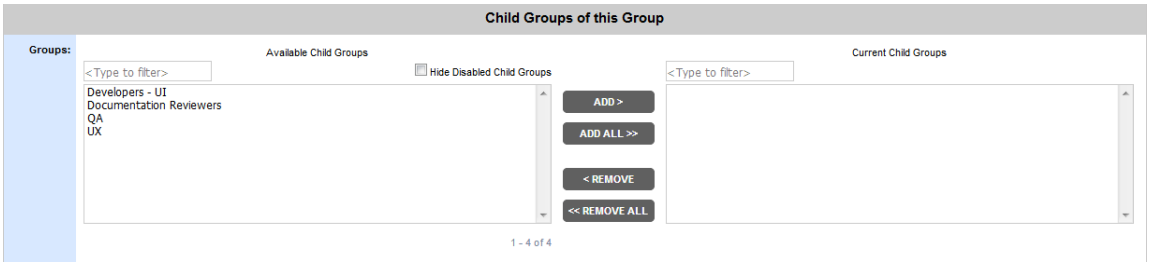



***Group administrators*** are regular (not full administrator) users who are allowed to administer this group. Group administrators can edit[229] all of the fields of the group.

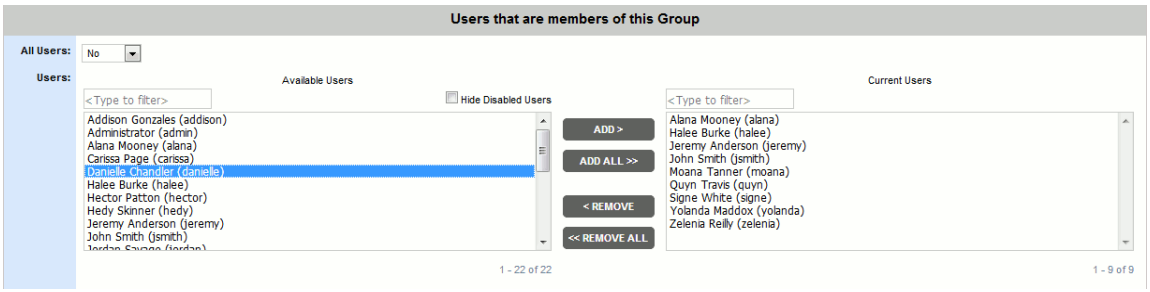
Group administrators can add or remove group administrators for the group, even themselves, but they can not remove the last group administrator from the group.

Group administrators can add or remove user members[232] of the group. Users that are group administrators may or may not also be regular members of the group.

Group administrators are not allowed to directly edit the list of child groups of the group, also they cannot create top-level groups. If the "Can Create Child Groups[225]" option is enabled for the respective user, group administrator can create new child groups of the group. The group administrator that has created a child group automatically becomes a group administrator of that child group.

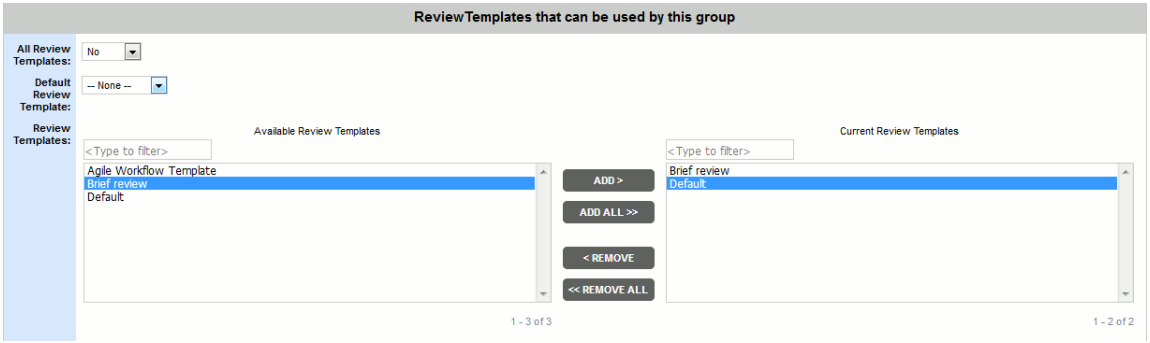



You can add multiple groups to be children of the group. This may be helpful if a group is constituted by a number of sub-groups. The "group hierarchy" created by Group -> Child Group relationships is important for the associate with review[226] and access control[227] algorithms.

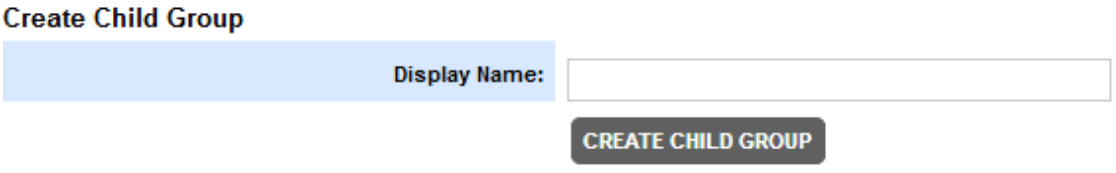



Groups can have specifically named member users, or they can automatically contain all users as members. Group membership is important for the associate with review[226] and access control[227] algorithms.

You can also configure which review templates[246] can be used by this group. That is, specify which workflows, sets of roles, rules will become available to group members when creating reviews. To make all review templates available to the group, set the **All Review Templates** option to Yes. To define a specific list of templates that should be available to the group, set the **All Review Templates** option to No and move the desired templates to the Current Review Templates list. The Default Review Template setting specifies the initial template that will be chosen when creating review (if set to "None", the first template in the list will be chosen).




Besides, on this page you can create child groups of the selected group:

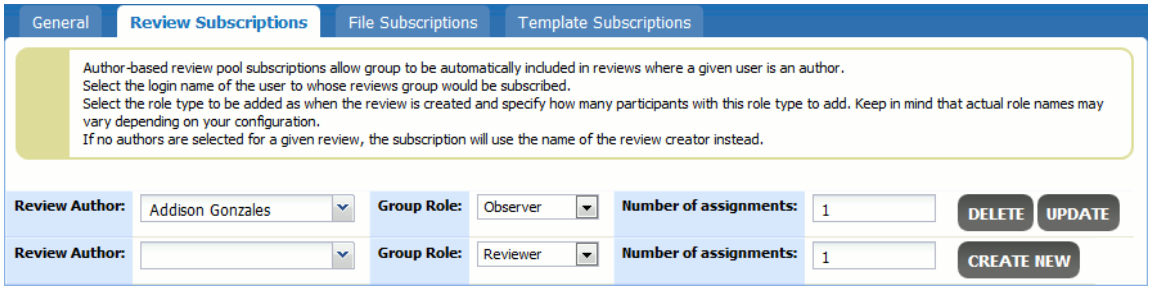



# Review Pool Subscriptions

When the group could be selected as a participant of a review[969] (that is when the Review Pool Participant setting is set to Yes), administrators can also set-up review pool subscriptions. Once a review of particular author is created, or a review contains some particular files, or a review uses some particular template, Collaborator will automatically add a group to this review as participant.

## Author Subscriptions

Author-based subscriptions allow a group to be automatically included in reviews of particular authors.

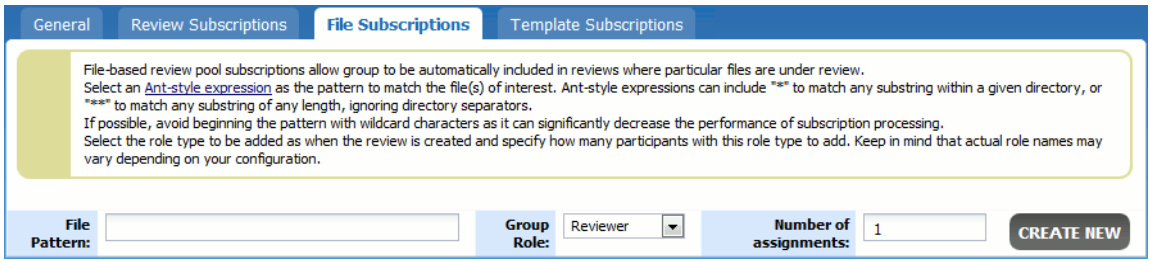



Under the "Review Author" field, select the login name of the user to whose reviews a group would be subscribed. If no authors are selected for a given review, the subscription will use the name of the review creator instead - unless the review creator is flagged as a system admin.

Select the desired role type of group members (**Reviewer**, **Observer**, or **Moderator**) and specify how many participants with this role type to add to the the review. Keep in mind that actual role names may vary depending on your role configuration 281.

## File Subscriptions

File pattern subscriptions allow a group to be automatically included in reviews where particular files are under a review.




Select an Ant-style expression as the pattern to match the file(s) of interest. The expression will be matched to the file path and repository name; it will not look for matches in the complete repository path with URL and other server descriptions. Ant-style expressions can include "*" to match any substring within a given directory, or "**" to match any substring of any length, ignoring directory separators. If possible, avoid beginning the pattern with wildcard characters as it can significantly decrease the performance of subscription processing.

Select the desired role type of group members (**Reviewer**, **Observer**, or **Moderator**) and specify how many participants with this role type to add to the the review. Keep in mind that actual role names may vary depending on your role configuration 281.

# Template Subscriptions

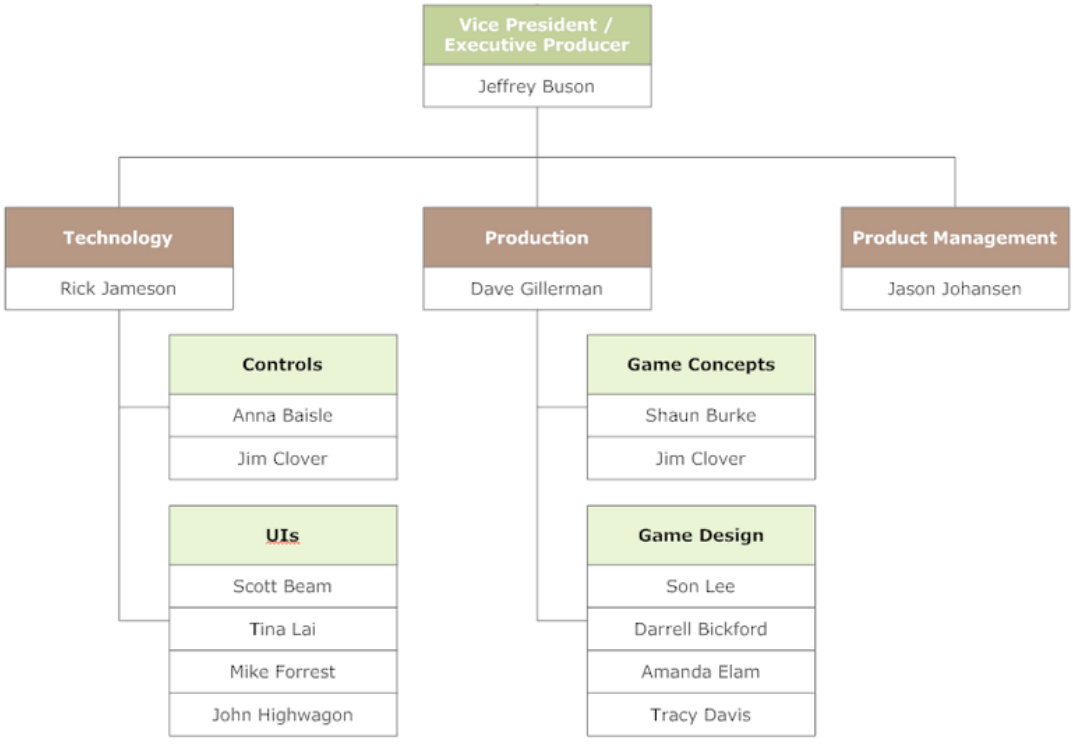
Template subscriptions allow a group to be automatically included in reviews that use particular templates.


Under the "Review Template" field, select the name of the template to which the group would be subscribed.

Select the desired role type of group members (**Reviewer**, **Observer**, or **Moderator**) and specify how many participants with this role type to add to the the review. Keep in mind that actual role names may vary depending on your role configuration 281.

### 3.3.10.1 Using Groups for Organizational Hierarchy

The Groups 226 feature is flexible enough to allow you to model any organization. As an example, suppose your company's **Action Games Division** org chart looked like this:



Companies this large typically store their org chart in an LDAP system. The Collaborator administrator can write a script that synchronizes 239 Collaborator groups with the LDAP system.

For example, the XML to create the organizational structure shown above would look like this:

Sample XML

```
<groups>-<group guid="action-games-division" title="Action
Games Division" allow-associate-with-reviews="false"><member-group
guid="technology"/><member-group guid="production"/><member-group
guid="product-management"/><member-user login="jbuson"/></
group>-<group guid="technology" title="Technology" allow-associate-
with-reviews="false"><member-group guid="controls"/><member-group
guid="uis"/><member-user login="rjameson"/></group>-<group
guid="production" title="Production" allow-associate-with-
reviews="false"><member-group guid="game-concepts"/><member-group
guid="game-design"/><member-user login="dgillerman"/></group>-<group
guid="product-management" title="Product Management" allow-associate-
with-reviews="false"><member-user login="jjohansen"/></group>-<group
guid="controls" title="Controls" allow-associate-with-
reviews="false"><member-user login="abaisle"/><member-user
login="jclover"/></group>-<group guid="uis" title="UIs" allow-
associate-with-reviews="false"><member-user login="sbeam"/><member-
user login="tlai"/><member-user login="mforrest"/><member-user
login="jhighwagon"/></group>-<group guid="game-concepts" title="Game
Concepts" allow-associate-with-reviews="false"><member-user
login="sburke"/><member-user login="jclover"/></group>-<group
guid="game-design" title="Game Design" allow-associate-with-
reviews="false"><member-user login="slee"/><member-user
login="dbickford"/><member-user login="aelam"/><member-user
login="tdavis"/></group></groups>
```

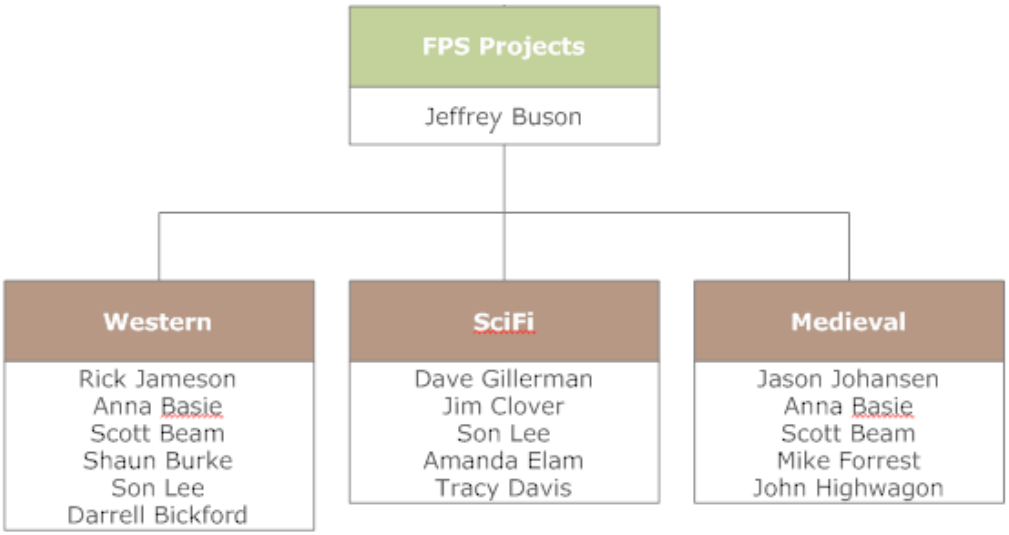
Note - we have assumed that user's logins are created by prefixing the first character from their first names to their last names. All of the groups have "Allow associate with Review 230" set to "No".

## Reporting based on group membership

The reporting system can be used to create reports based on group membership. For example, all reviews where a member of a department was a participant, or all reviews created by a member of a department, and so on.

### 3.3.10.2 Using Groups for Projects

The Groups 226 feature is flexible enough to allow you to model any structure. As an example, if you wanted to model a group of projects from a software gaming company:




Since there is only a few groups we create them manually using the web UI. First create a group called *FPS Projects*. The *FPS Projects* group would have only one user in it: Jeffrey Buson. It would also have three child groups: *Western*, *SciFi*, and *Medieval*. The *Western*, *SciFi*, and *Medieval* groups would each contain the users that are assigned to those projects.

## Associating groups with reviews

We set "Allow associate with Review[230]" to "Yes" for the *Western*, *SciFi*, and *Medieval* groups. In this example we are using groups to represent projects, so we configure the singular group label[184] to "Project" and the plural group label[184] to "Projects".

When a user in one of those groups creates a review it will be associated with a group. Most of the users are only members of one group, so when they create a review they do not have to manually choose a group - the review is automatically associated with their group. The users "Anna Basie" and "Scott Bean" are members of both the *Western* and *Medieval* groups, so when they create a review they are prompted to select one of those "Projects" to associate with the review.

## Filtering the participant list

When a user creates a review they can choose to filter the participant list[227] by that group. When Amanda Elam creates a review, if she selects "Filter possible participants by 'SciFi' Project" then the participant drop-downs will only contain Dave Gillerman, Jim Clover, Son Lee, and Tracy Davis. This is useful at large companies to make the list of users more manageable.

## Restricting Access based on group

Access to reviews can be restricted based on associated group. The restriction can be set for all reviews or just for a particular review. For example, if a review was associated with the *Western* group and group access restriction was enabled, then it would be accessible to Scott Beam and Jeffrey Buson, but not accessible to Jason Johansen.

## Reporting based on associated group

You can display, sort and filter reports by the associated group, so you know which project the review was for.

## Finishing projects

When a project is finished, we mark it is group disabled. This means the group can no longer be associated with reviews, but it is still in the system for reporting purposes.

### 3.3.10.3 Using Groups for Organizational Hierarchy and Projects

The Groups 226 feature is flexible enough to allow you to model any structure. You might want to use groups to represent an organizational hierarchy like the Using Groups for Organizational Hierarchy 234 example *and* use groups to represent projects like the Using Groups for Projects 235 example. This is supported and gives us all the advantages of both approaches.

## Creating Groups

We create the groups to represent organizational hierarchy using the sync process as described in the Using Groups for Organizational Hierarchy 234 example. We create the groups to represent the projects manually as described in the Using Groups for Projects 235 example. Collaborator remembers which groups were created manually and which were created using sync, so the sync process does not delete the manually created project groups.

## Assigning Group Members

Instead of manually naming each user that is a member of each project group like we did in the Using Groups for Projects 235 example, we can indicate that all the members in an organizational hierarchy department are working on a project by making the organizational hierarchy group a child of the project group. This is much easier than assigning individual user members, but we lose some fine-grained control.

## Associating groups with Reviews

All of the organizational hierarchy groups have the "Allow associate with review[230]" option set to "No". The *Western*, *SciFi*, and *Medieval* project groups have the "Allow associate with Review[230]" option set to "Yes". We configure the singular group label[184] to "Project" and the plural group label[184] to "Projects".

When a user creates a review it will be associated with a project group. If the user's organizational hierarchy group is only a member of one project group then when they create a review they do not have to manually choose a group - the review is automatically associated with the project group. If a user's organizational hierarchy group is a member of more than one project group then when they create a review they are prompted to select one of those "Projects" to associate with the review.

## Filtering the participant list

When a user creates a review they can choose to filter the participant list[227] by that group. This lets a user filter the list of participants by only the users working on the project associated with the review. This is useful at large companies to make the list of users more manageable.

## Restricting Access based on group

Access to reviews can be restricted based on associated group. The restriction can be set for all reviews or just for a particular review. For example, if a review was associated with the *Western* group and group access restriction was enabled, then only users working on the *Western* project would be able to access it.

## Reporting based on group membership

The reporting system can be used to create reports based on group membership. For example, all reviews where a member of a department was a participant, or all reviews created by a member of a department, and so on.

## Reporting based on associated group

You can also display, sort and filter reports by the associated group, so you know which project the review was for.

## Finishing projects

When a project is finished, we mark its group disabled. This means the group can no longer be associated with reviews, but it is still in the system for reporting purposes.

#### 3.3.10.4 Syncing Groups and Their Members

Collaborator provides a simple web UI for creating and managing groups, but when groups are used to represent organizational hierarchy[234] for large companies the web UI becomes unmanageable. Such groups are usually already stored in an external system such as LDAP or Active Directory, Single Sign-On server, or repository hosting server, so Collaborator provides the ability to synchronize its users and groups with external system.

Collaborator offers two ways to perform synchronization: you can setup user and group mapping from the administrator's Web interface or run the special command of command-line client. Once established, user and group mapping operates constantly (upon every login), however it supports just several most frequently used systems: LDAP and Active Directory[119], SAML single sign-on[132] servers, Crowd single sign-on[145] servers and GitLab servers[866]. Synchronizing through command-line interface accepts group data in XML format and thus is suitable for any system. However it requires more efforts to create the input XML file and the command should be executed periodically to keep the data synchronized.

Syncing Users and Groups via Web Interface[239]

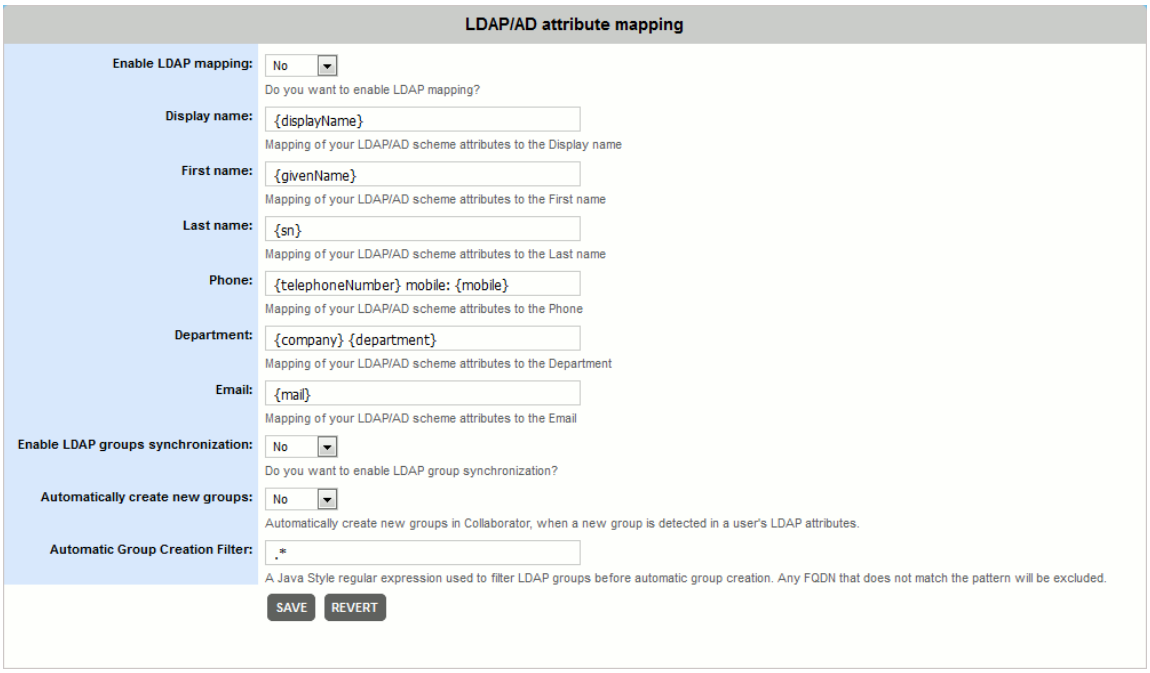
Syncing Users and Groups via Command-line Interface[243]

## Syncing Users and Groups via Web Interface

### Syncing with LDAP or Active Directory

You may configure user and group synchronization between Collaborator and the LDAP directory or Active Directory. In this case, Collaborator will retrieve user properties (name, phone, email, and so forth) and their membership in groups from the LDAP directory or Active Directory when the users login. Additionally, you can select whether to create new groups automatically and specify regular expression for automatic group creation.

To enable synchronization on Active Directory systems, you will need to open the LDAP Settings[198] tab of General settings, enable the respective properties and possibly adjust the attribute mapping configuration.




In order to perform synchronization on LDAP systems, you will need to configure the above-mentioned LDAP Settings[198] and also need to modify the ROOT.xml. Namely, you will need to add the following fields: the `connectionName` and `connectionPassword` fields which define a user account the Collaborator will use to connect to LDAP to find the group membership user records, and the `roleBase` and `roleSearch` fields which define the base entry for the role search and the search filter for selecting role entries.

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionName="cn=read-only-admin,dc=example,dc=com"
       connectionPassword="xxxx"
       connectionURL="ldap://xxxx.com:389"
       userPattern="uid={0},dc=example,dc=com"
       roleBase="dc=example,dc=com"
       roleSearch="(&amp;(objectClass=groupOfUniqueNames)(uniqueMember={0}))"
       allRolesMode="strictAuthOnly"
/>
```

**Technical details**

For mapping user membership in groups Collaborator uses the group's fully qualified domain name (FQDN) retrieved from the LDAP or AD. It checks if some of existing user groups[226] has matching FQDN and adds the user to this group on success. Otherwise, it can create new group (if automatic group creation is enabled and group name matches filter) and adds the user to the new group.

To name the new group Collaborator uses the first entry of group's fully qualified domain name (FQDN). For example, a group having the "ou=ccusers,dc=example,dc=com" FQDN will have the ccusers title. If some other group already has the same title, Collaborator will append the ordinal number to the group title: ccusers1, ccusers2 and so on.

If users have multiple emails specified on the LDAP or Active Directory side, Collaborator will retrieve them and use the first of the emails for notifications. Email addresses should be separated by comma.

Collaborator checks existing groups created via LDAP or Active Directory synchronization and actualizes user membership in those groups on every login. Such algorithm allows to keep a consistency between Collaborator and LDAP or Active Directory.

## Syncing with SAML single sign-on server

You may configure user and group synchronization between Collaborator and your SAML single sign-on server. In this case, Collaborator will retrieve user properties (email, first and last name), permissions and their membership in groups from the single sign-on server when the users login.

To enable group synchronization with SAML SSO server:

1. Open the **Admin > Single Sign-On** page.
2. Locate your SAML SSO configuration in the **SSO Configurations** list and click **Edit**.
3. In the **MemberOf parameter** field specify the name of SAML parameter which stores information about user membership in groups.
4. Set the **Enable SAML group sync** setting to **True**.
5. Optionally, enable the **Allow new group creation** to create new groups automatically.
6. Optionally, use the **Automatic group creation filter** to specify what exactly groups should be created automatically.
7. Press **Save** to apply changes and enable synchronization.

**Technical details**

Collaborator retrieves user membership from the SAML single sign-on server and checks if some of existing user groups 226 has matching name. Once such group is found, it adds the user to this group. Otherwise, it checks if the "Allow new group creation" setting is enabled and if the group name matches the "Automatic group creation filter", creates new group on success and adds the user to the new group.

Along with membership information, Collaborator is able to sync user permissions with SAML SSO server. If the **User Role sync** 140 option is enabled, it will synchronize user roles between SAML and Collaborator servers. SSO accounts having admin privileges (Role = admin) will automatically become administrators on Collaborator server. SSO accounts having empty or any other value of the `Role` parameter will become regular users on Collaborator server.

Collaborator actualizes user properties, their membership in groups and permissions on every login. Such algorithm allows to keep a consistency between Collaborator and SSO server.

When SAML SSO integration is disabled or deleted, synchronized groups will become disabled.

Alternatively, you may configure user membership synchronization between Collaborator, SAML single sign-on server and LDAP directory or Active Directory. In this case, Collaborator will use single sign-on server to manage users, but will query user membership from the LDAP directory or Active Directory. See Configuring Single Sign-On via SAML 143 for instructions on enabling this variant of sync.

## Syncing with Crowd single sign-on server

You may configure user and group synchronization between Collaborator and the Atlassian Crowd server. In this case, Collaborator will retrieve user properties and their membership in groups from the Crowd server when the users login.

To enable group synchronization with Crowd server:

1. Open the **Admin > Single Sign-On** page.
2. Locate your Crowd SSO configuration in the **SSO Configurations** list and click **Edit**.
3. Set the **Enable Crowd group sync** setting to **True**.
4. Optionally, use the **Automatic group creation filter** to specify what exactly groups should be created automatically.
5. Press **Save** to apply changes and enable synchronization.

**Technical details**

Collaborator retrieves user membership from the Crowd server and checks if some of existing user groups 226 has matching name. Once such group is found, it adds the user to this group. Otherwise, it checks if the group name matches **Automatic group creation filter**, creates new group on success and adds the user to the new group.

Collaborator actualizes user membership in groups on every login. Such algorithm allows to keep a consistency between Collaborator and Crowd server.

When Crowd integration is disabled or deleted, synchronized groups will become disabled.

### Syncing with GitLab server

You may configure user and group synchronization between Collaborator and the GitLab server. In this case, Collaborator will retrieve user properties and their membership in groups from the GitLab server when the users login.

To enable group synchronization with GitLab server:

1. Open the **Admin > Repository Hosting Services** page.
2. Set the global **Enable Group Sync** 216 setting to **Yes** and press **Save**.
3. Locate your GitLab configuration in the **Remote Systems Configurations** list and click **Edit**.
4. Set the **Sync Groups** setting to **True**.
5. Press **Save** to apply changes and enable synchronization.

**Technical details**

Collaborator retrieves user membership from the GitLab server and checks if some of existing user groups 226 has matching name. Once such group is found, it adds the user to this group. Otherwise, it creates new group and adds the user to the new group.

Collaborator actualizes user membership in groups on every login. Such algorithm allows to keep a consistency between Collaborator and GitLab server.

When GitLab integration is disabled or deleted, synchronized groups will become disabled.

## Syncing Users and Groups via Command-line Interface

The `ccollab admin group sync` command takes as input an XML file describing all the groups being synced, their relationships, and their member users. The XML format is simple and a full schema is provided: group-sync-xml.xsd (opens in a new window). For an example see Using Groups for Organizational Hierarchy 234. The Collaborator administrator should write a script that queries the external system (that is, LDAP) and creates this XML. For an example of this script, see the Examples section of the `ccollab admin group sync` command.

The Collaborator administrator should run the sync script periodically. Collaborator will examine the group sync xml and make the appropriate changes in the system. Processing the sync is not very resource intensive, so you probably do not need to worry about load on the Collaborator server.

## Adding Users

The group sync xml describes which users are admins and/or members of which groups. There are two options if the XML describes users who do not yet have accounts in Collaborator, controlled by the `--create-user` option of the `ccollab admin group sync` command:

- **Create the member-user** - If the --create-user option is specified, a user account is created for the user. Note the user does not consume a license 90 until they log in.
- **Ignore the member-user** - If the --create-user option is not specified (default behavior), the member-user is ignored and a user account is not created. This is useful if you have a large organization but only a small subset of the employees are using Collaborator.

## Removing Groups

Groups may be removed from the external system. There are two options if the XML no longer describes a group which was created in an earlier sync, controlled by the `--delete-groups` option of the `ccollab admin group sync` command:

- **Delete the group** - If the --delete-groups option is specified, the group is deleted. However if the group can not be deleted because it has associated reviews 229, an error is returned and the sync is canceled.
- **Disable the group** - If the --delete-groups option is not specified (default behavior), the group is marked disabled. This always works, regardless of whether there are reviews associated with the group.

## Renaming groups

Groups are uniquely identified by a GUID which is specified in the group sync xml. To rename the group simply specify a group with the same GUID but a different title.

## Groups that were created manually

Sometimes it is useful to sync groups with an external system but *also* manually specify some groups using the Web Client or the `ccollab admin group create` command. For an example see Using Groups for Organizational Hierarchy and Projects 237. Collaborator keeps track of which groups were created using sync and which groups were created manually. Manually created groups are not deleted or disabled on subsequent syncs just because they are not in the group XML. The web UI displays a warning when manually editing a group that was created by sync. Manually created groups can contain synced groups as children, but not vice-versa.

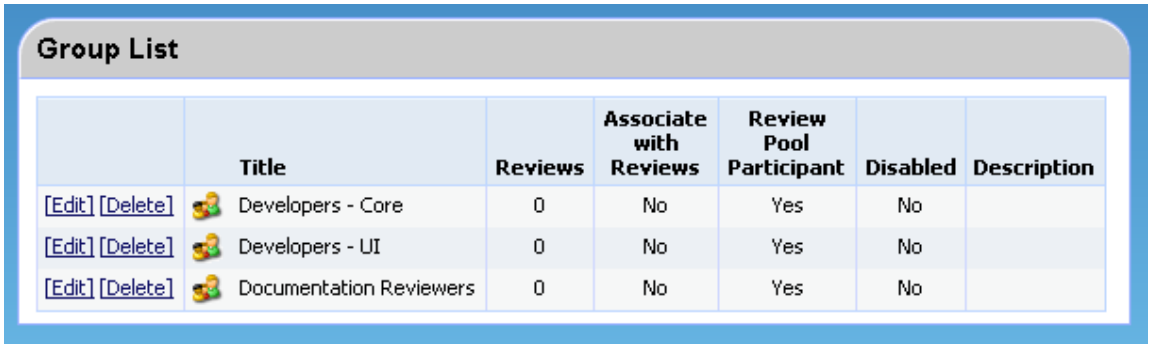
### 3.3.10.5 Using Groups for Review Pools

## Groups for Review Pools

Review Pools are Groups that can be selected as participants in a Review[969]. To enable Review Pools, a Collaborator administrator must define specific Groups to be selectable for Review Pools. You can configure Groups for Review Pools specifically for use as Review Pools, or select Review Pools from an existing group hierarchy.

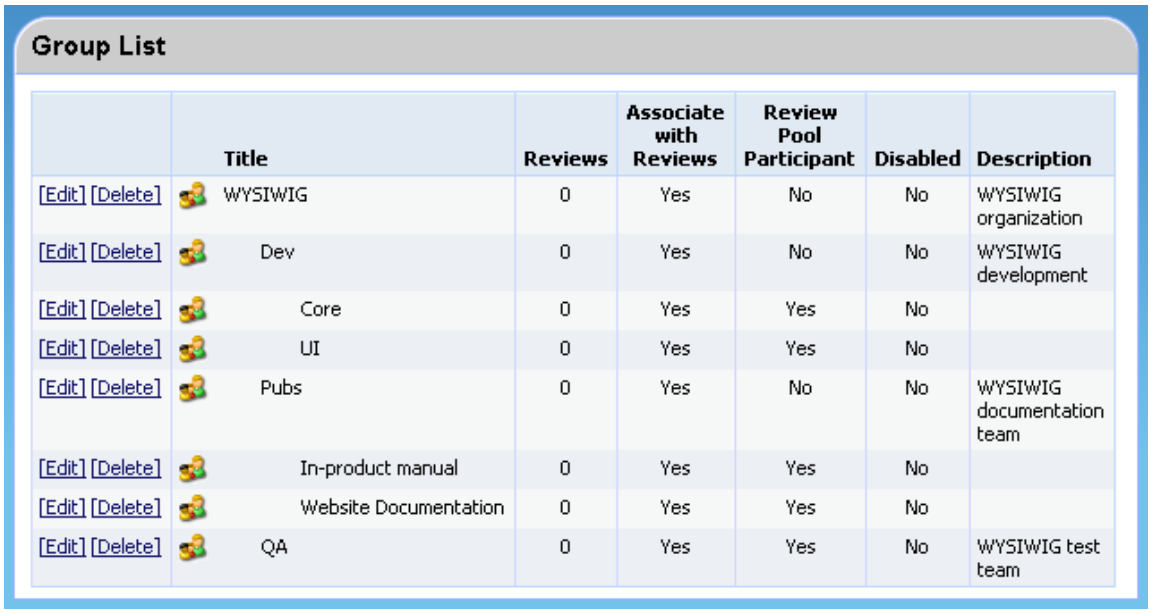
## Using Groups for Review Pools without a Group Hierarchy

As a Collaborator administrator, if you do not already use a group hierarchy, or if it does not make sense to use it for Review Pools, then you should create Groups only for use by Review Pools. Create one or more Groups containing the users who are candidate participants. For each Group to be used as a Review Pool without a group hierarchy, enable **Review Pool Participant** and disable **Associate With Reviews**. Shown below is an example where several Groups have been created specifically for use as Review Pools.




## Using Groups for Review Pools with Group Hierarchy

As a Collaborator administrator, if you have a group hierarchy already defined for your organization, you may designate any existing Groups to also be used for Review Pools. For each group to additionally be used for Review Pools, enable **Review Pool Participant** and leave **Associate with Reviews** enabled. Shown below is an example organization where the leaf groups have been enabled for use as Review Pools.

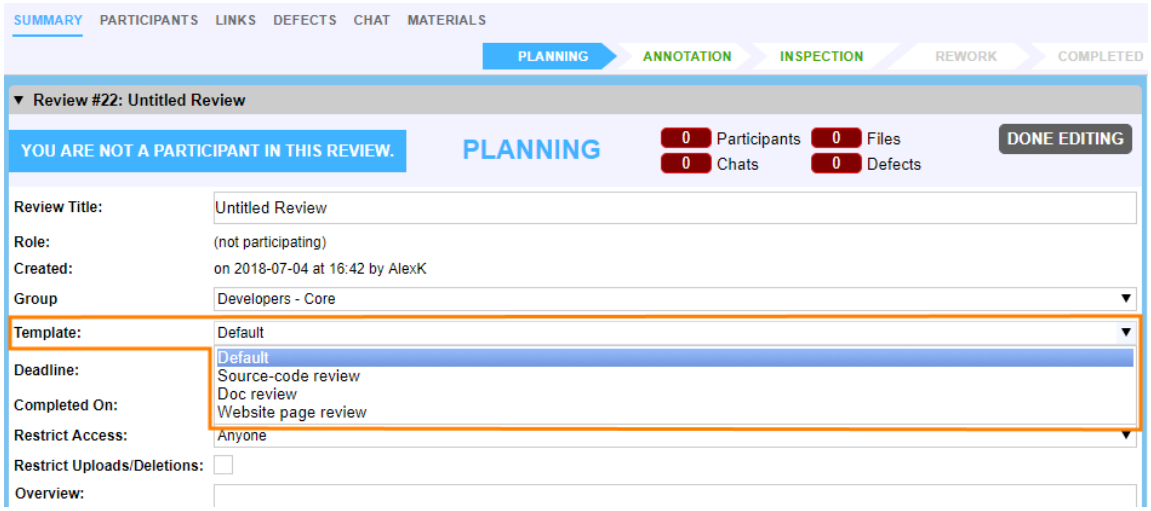



## 3.3.11 Review Templates

### About

As you integrate reviewing into your work process, you may need to create slightly different reviews for different document types or different projects. For example, depending on the document type reviews may require different type-specific fields, and depending on projects they may need slightly different participant roles.

To adjust a review to the form you need, you use review templates. A review template is a configuration of user-defined fields, roles, and other settings that you can quickly apply to a review. You can do this when you are creating a review or editing it:



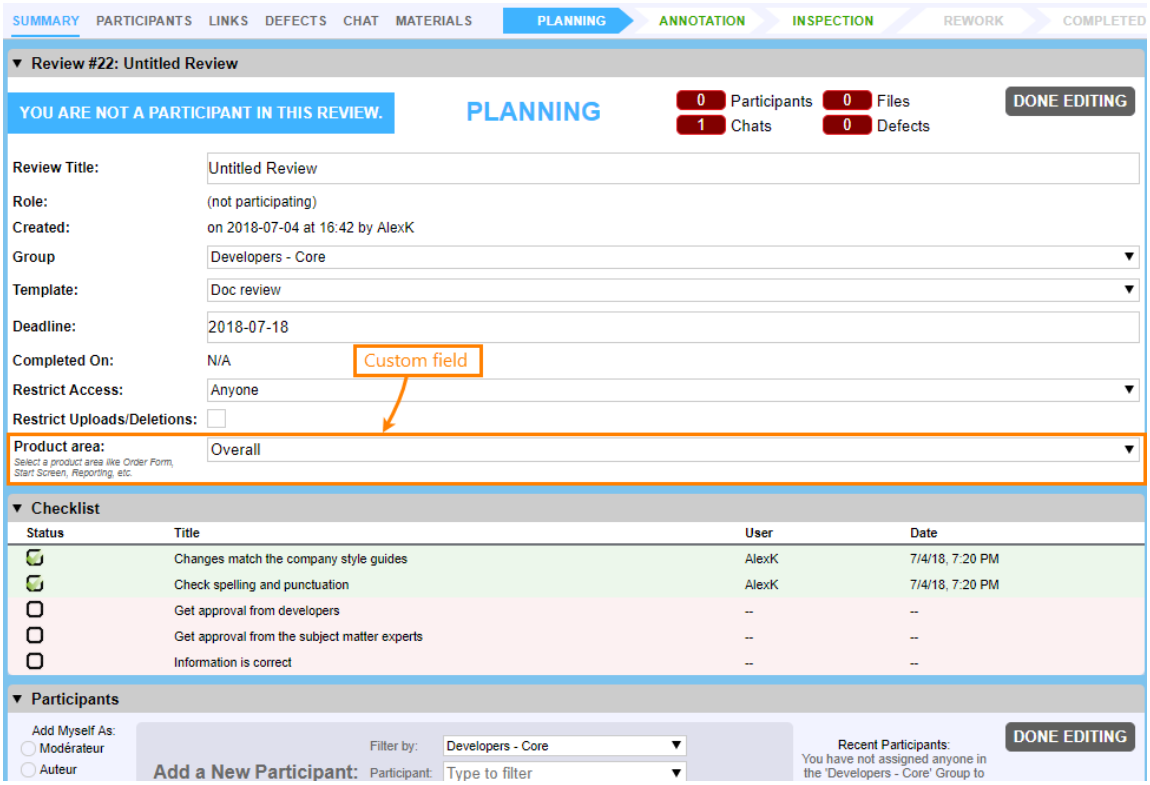
Applying a template to a review may clear custom fields and the list of participants of this review. Existing values of custom fields will be retained upon changing review template if the custom field exists in both templates. Existing review participants will be retained upon changing review template if they are allowed to use new template.

## Template Components

Each review template includes the following settings:

### Custom fields

One or multiple custom fields to be added to a review. You can see these custom fields at the beginning of the review, right after the general settings:

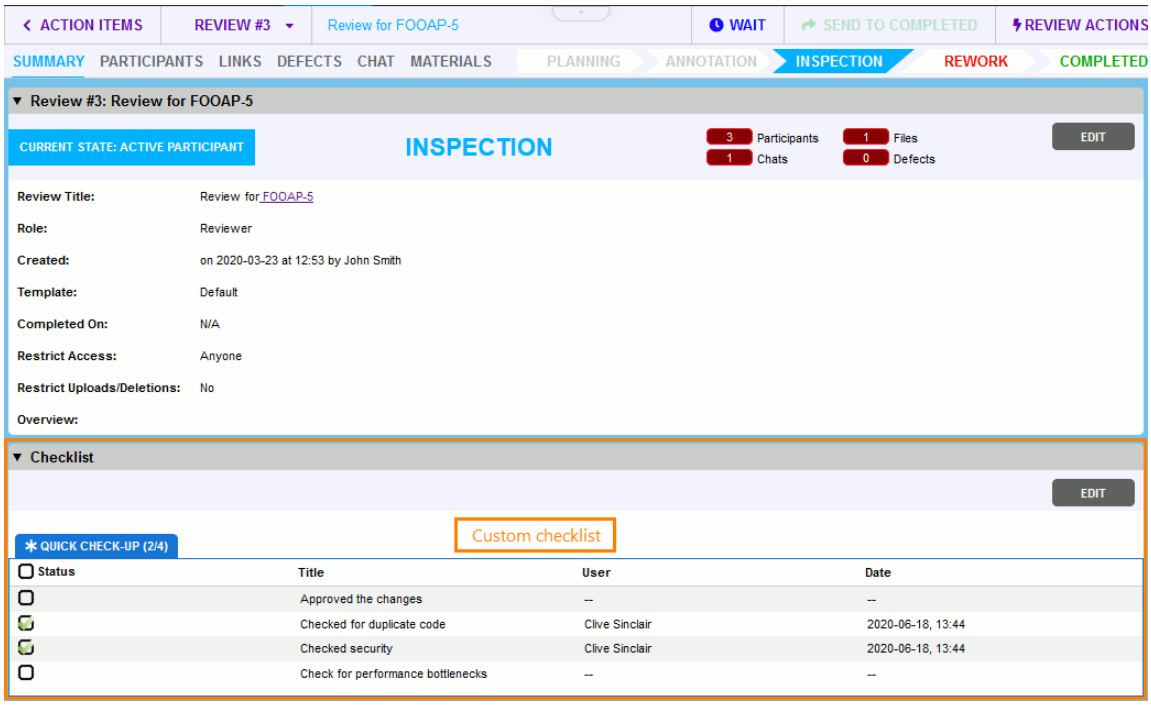



*Custom fields on the review screen*

For information on creating custom fields and on their parameters, see Custom Fields 256.

### Checklists

A checklist is a list with check boxes for its items. You can see it as a list of actions to do or conditions to check for a review. You can create one or multiple checklists for your reviews, and use different checklists for different document types. You can see the check lists on the review screen.




*Checklists on the review screen*

For information on creating and configuring check lists, see Checklists 272.

## Roles

A configuration of participant roles that Collaborator will use in a review. You can use them to change role permissions and names.

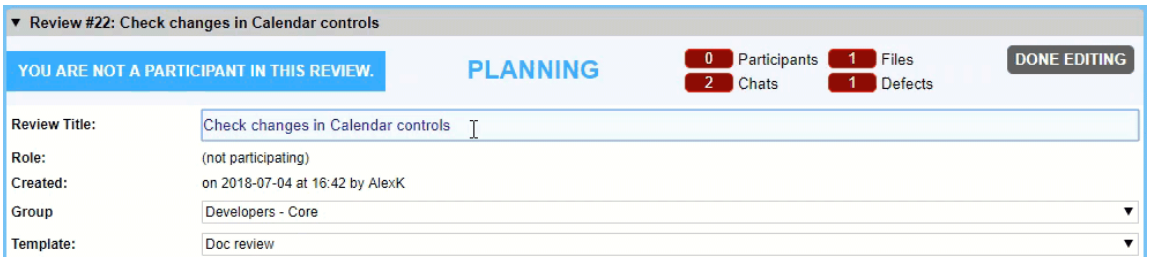

*Custom roles on the review screen*

For information on creating and configuring roles, see Roles 279.

## Automatic links

Review fields may contain links to external resources like website pages or links to items of a remote bug tracking system.

The Automatic links group of settings let you specify rules that Collaborator will use to replace text in multi-line or single line fields with links to remote resources.




*Automatic links in text fields*

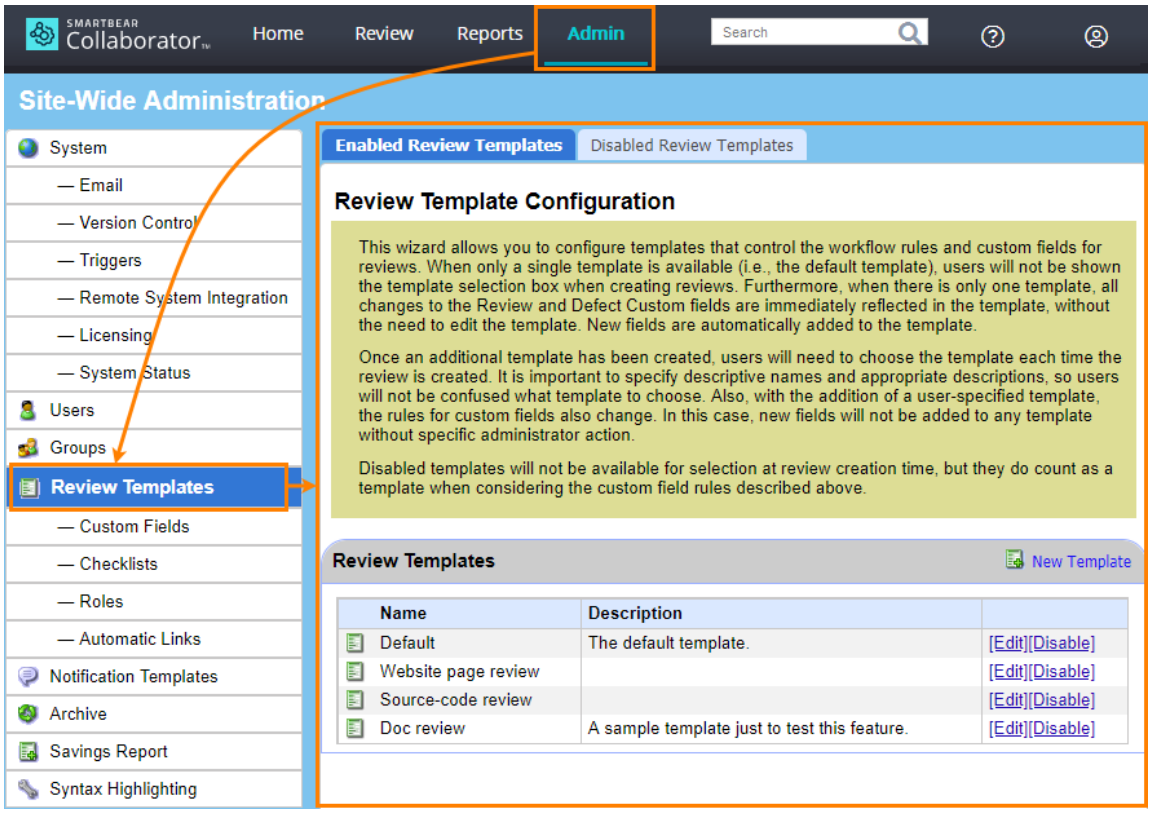
For information on creating and configuring automatic links, see Automatic Links 286.

## Requirements

- Review templates are available in Collaborator Team and Collaborator Enterprise. For a complete list of differences between Collaborator editions, see Collaborator Editions 3.
- To create and edit review templates, you need administrator permissions in Collaborator.

## Create and Edit Template

You create and edit review templates in the **Admin > Review Templates** section of Collaborator settings:

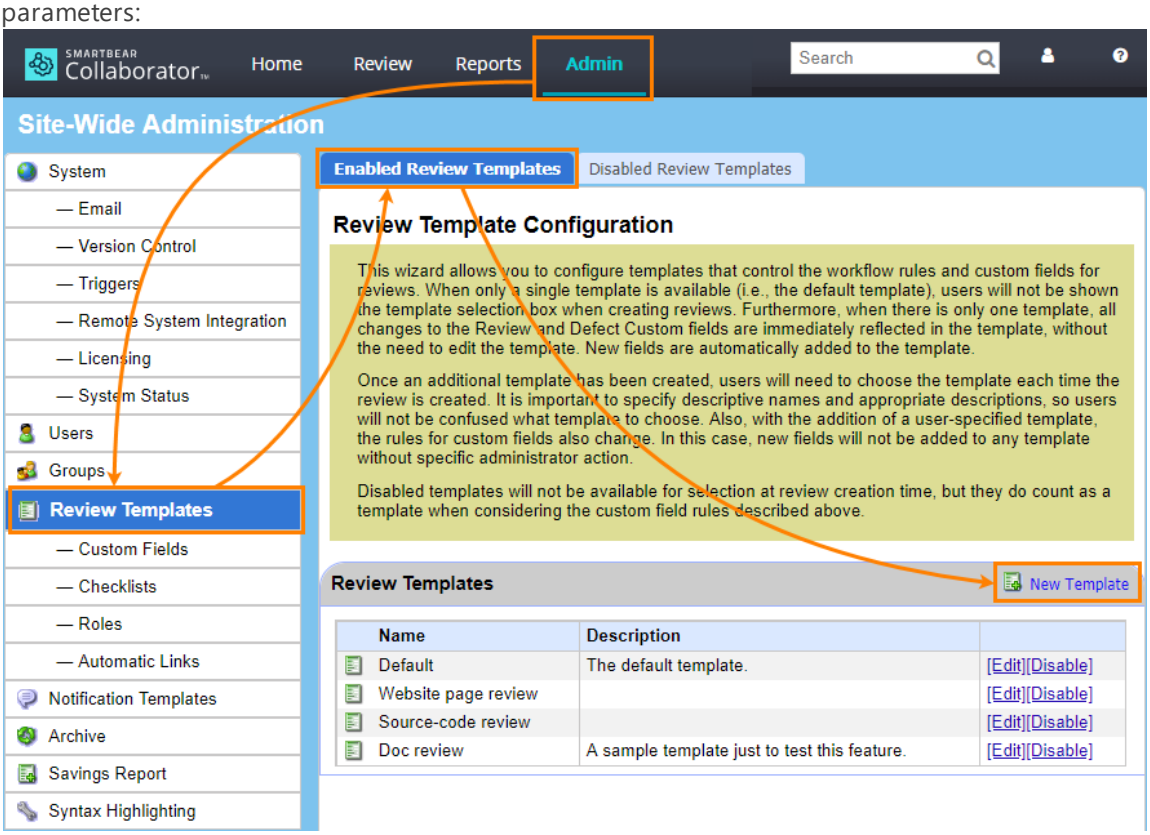



## Create template

1. Log in to Collaborator as administrator.

2. Select **ADMIN** on the main toolbar. Then choose **Review Templates** on the left.

3. On the **Enabled Review Templates** tab, click **New Template** and specify the template-parameters:




*Create a review template*

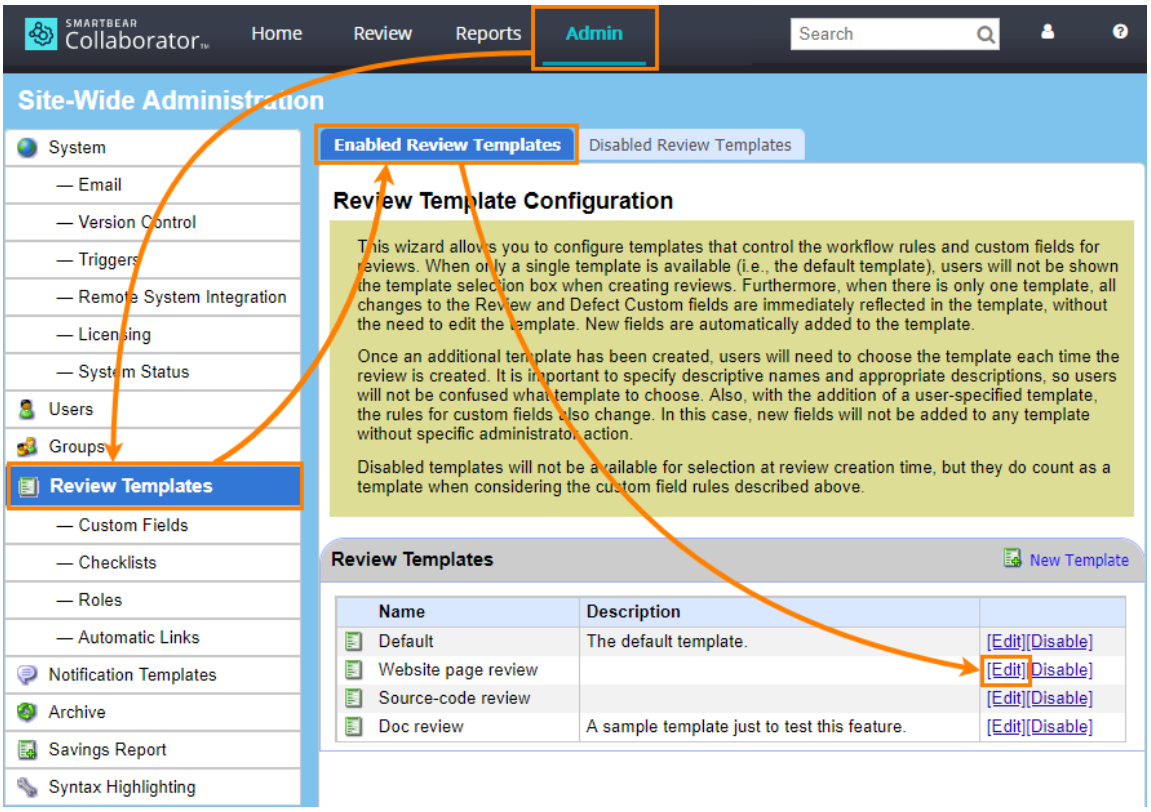
4. Click **Save** anywhere on the page to save the changes.

## Edit template

1. Log in to Collaborator as administrator.

2. Go to **ADMIN > Review Templates**.

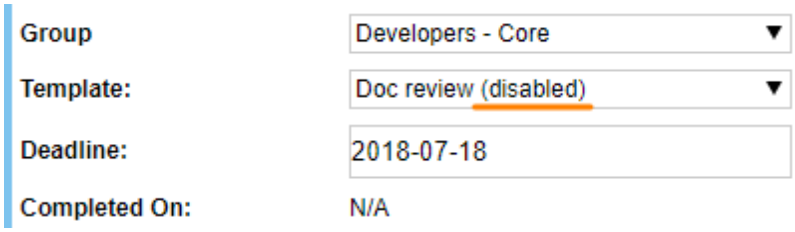
3. Find the review in the list and click **Edit**:




*Edit a review template*

4. Configure template parameters and click any **Save** button to apply the changes.

Collaborator will not apply changes to existing reviews immediately. Instead, it will mark the template as disabled signaling that you need to re-apply the template:

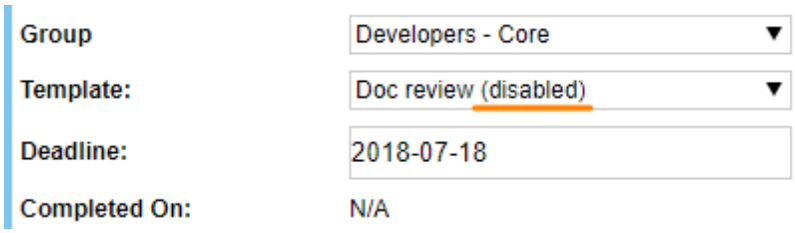



*Disabled templates on the review screen*

To re-apply a template, simply select it again from the Template drop-down list.

## Disable or delete template

Review templates cannot be deleted. If you don't need a template any more, you can disable it. This will hide that template from the Template drop-down box on the Review screen:



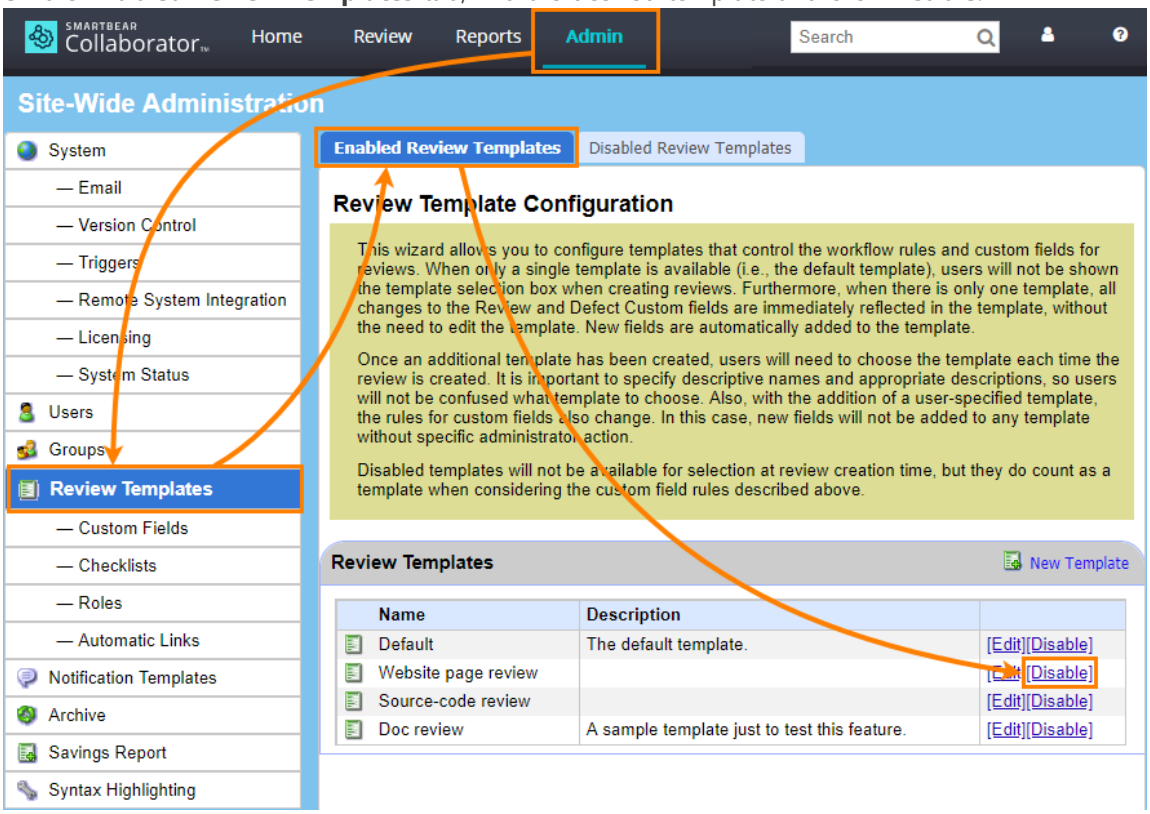

*Disabled templates on the review screen*

To disable a template:

1. Log in to Collaborator as administrator.

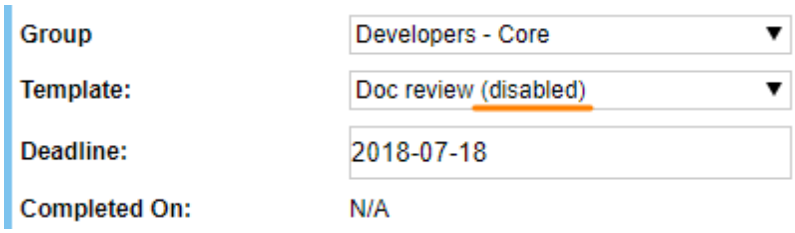
2. Go to **ADMIN > Review Templates**.

3. On the **Enabled Review Templates** tab, find the desired template and click **Disable**:




*Disabling a review template*

Collaborator will not update reviews that use the disabled template immediately. Instead, it will mark the template as disabled on the review screen, so that you know you need to apply another template:



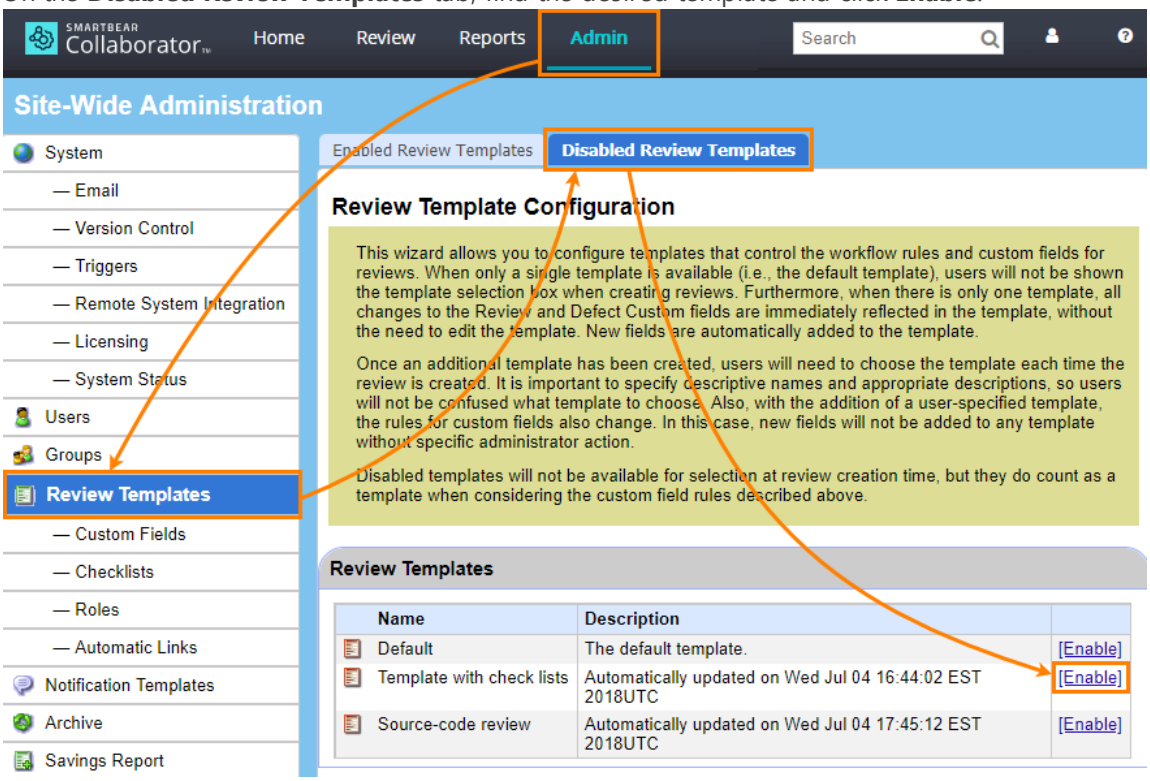

*Disabled templates on the review screen*

## Enable template

1. Log in to Collaborator as administrator.

2. Go to **ADMIN > Review Templates**.

3. On the **Disabled Review Templates** tab, find the desired template and click **Enable**:




*Enabling a review template*

## Template Parameters

| Parameter | Description |
|---|---|
| **Name** | The name that will identify the template in the Collaborator UI. |

| | |
|---|---|
| **Description** | An arbitrary description of the template. |
| **Review reject reasons** | Specifies whether to display a list of reject reasons 347 when rejectig reviews that use this template. |
| **<Checklist_ Name>: Use in review** | Specifies which checklists 272 should be available to reviews that use this template. |
| **<Checklist_ Name>: Mandatory** | Specifies whether all items must be checked before the review can be approved. |
| **<Checklist_ Name>: Default** | Right after you create a review, the Review Screen displays only one checklist. Review participants append other checklists later, if needed. This setting specifies the checklist to be visible by default. If no checklist is selected as default, the Review Screen will show the first created checklist. |
| **Review custom fields** | Specifies which review custom fields 256 should be available to reviews that use this template. |
| **Participant custom fields** | Specifies which participant custom fields 256 should be available to reviews that use this template. |
| **Defect custom fields** | Specifies which defect custom fields 256 should be available to reviews that use this template. |
| **Roles** | Specifies which of predefined role configurations 279 to use for this template. |
| **Automatic links: <Configuration_Name>** | Specifies which of predefined automatic link configurations 286 will be enabled for this template. |
| **Remote system links: Enabled** | Specifies whether to display the Remote System Links section of the Review Summary page for reviews that use the template. |

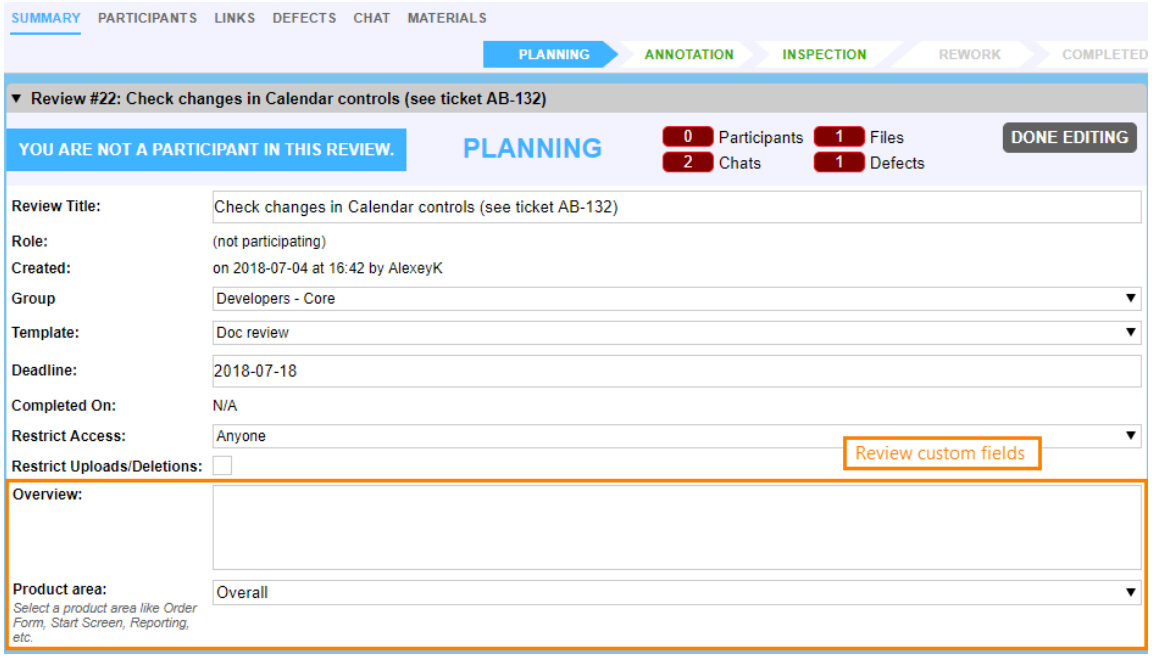
| | |
|---|---|
| **<Configuration_Name>: Automatically add remote system links** | Specifies which of predefined issue-tracking configurations[886] will be enabled for this template. This will converts ticket identifiers into hyperlinks to the specified remote system and will append the items to the Remote System Links section of the Review Summary page. |
| **<Configuration_Name>: Automatically create new work item for external defects** | Enables the ability to create new work items in the selected issue-tracking system[442] directly from Collaborator. |
| **<Configuration_Name>: Default project** | Specifies the project to be pre-selected in the dialog that suggests creating a ticket/work item. |
| **Enable comments/ defects during rework phase** | If enabled, other participants could create comments and defects even when the review is in the rework phase.<br><br>If disabled, only review creators, authors and Collaborator administrators could create comments and defects during the rework phase. |
| **Enable comments/ defects after reviews deadline** | If enabled, other participants could create comments and defects even when the review has reached its deadline.<br><br>If disabled, only review creators, authors and Collaborator administrators could create comments and defects when the review has reached its deadline. |

## 3.3.12 Custom Fields

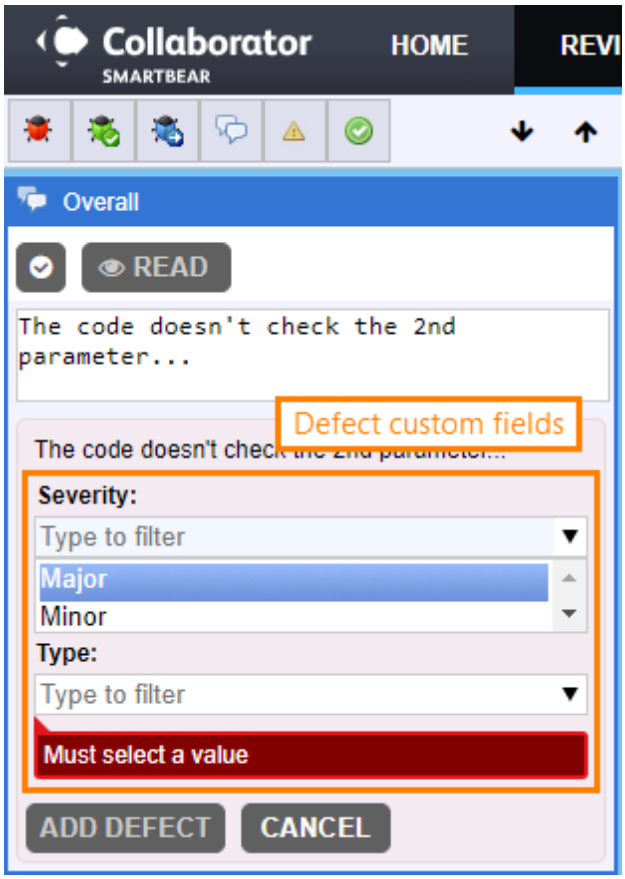
### About

Custom fields allow adding arbitrary information fields to reviews. They help to fit Collaborator reviews into your workflow. Currently Collaborator supports custom fields for the following areas: reviews, defects, checklists and participants.

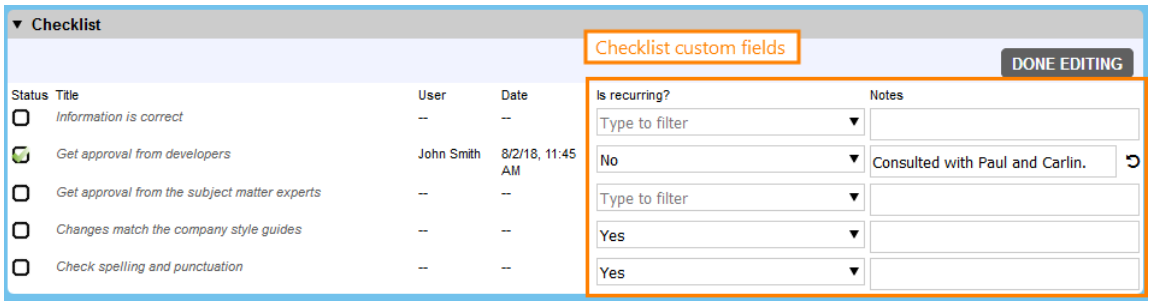
***Review custom fields*** allow to specify and view any additional information for the general section of the review.



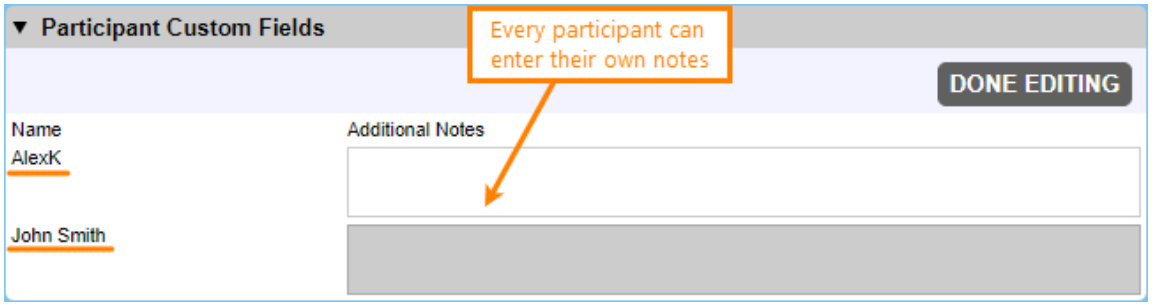

***Defect custom fields*** allow to specify and view any arbitrary information for defects.




***Checklist custom fields*** allow to specify and view any arbitrary information for checklist items.




***Participant custom fields*** allow each review participant to specify and view any arbitrary information about the review:




Although these fields show up in different places in the user interface, the administrative interface for configuring these fields is identical. Administrators create and edit custom fields of all these types in the **Review Templates > Custom Fields** settings. See below for details.

Custom field values are included in the search[458] and in reports[467].

## Requirements

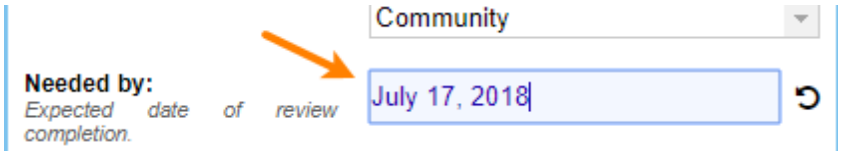
- Custom fields are not supported in Collaborator Community. Collaborator Team offers limited support for custom fields, and Collaborator Enterprise fully supports them. For a complete list of differences between Collaborator editions, see Collaborator Editions.
- To create and edit checklists, you need administrator permissions in Collaborator.

## Custom field types

Collaborator supports the following types of the custom fields:

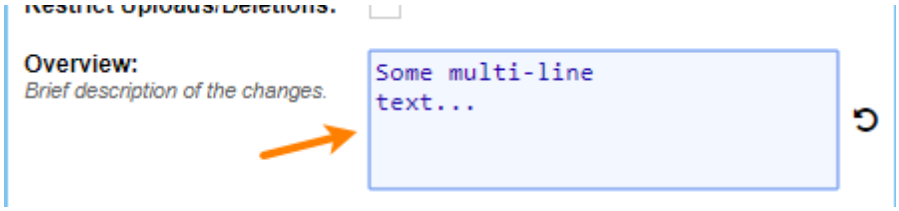
### String (single-line)

These are text fields displayed as usual text boxes. Can store up to 255 characters. Supports various validation options. Can be used for creating date/time edit boxes and other controls.




## String (multi-line)

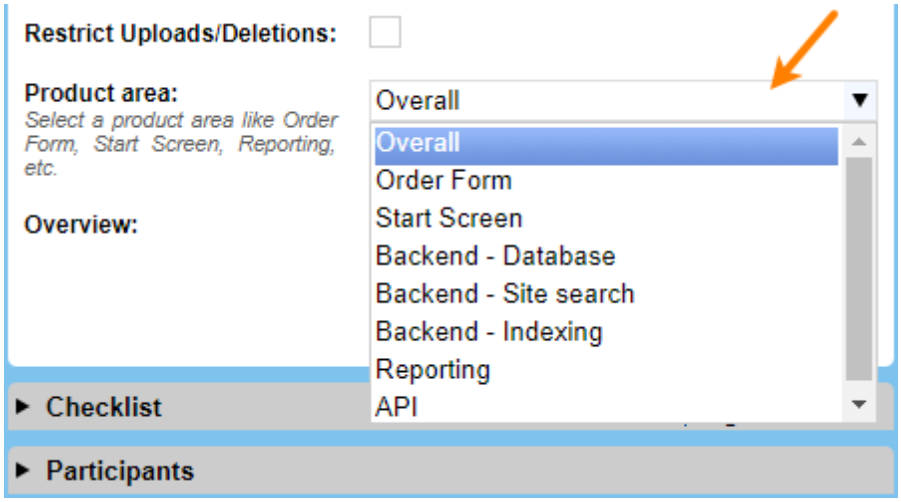
Fields that support long texts. Displayed as multi-line edit boxes with vertical and horizontal scrollbars (if needed).




The maximum allowed text size depends on the database server, where Collaborator is installed. The lower bound is 8192 characters. It can be more and reach up to several thousands of characters.

## Drop-down list

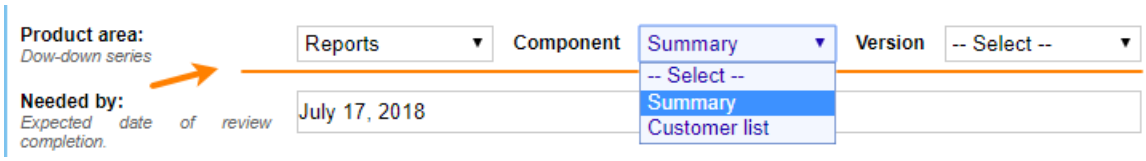
Fields that let users select one item from a pre-defined list of items. Displayed as combo boxes:




These fields support the search-as-you-type functionality: when users are typing, Collaborator shows only those items in the drop-down list that contain the entered text.

## Drop-down series

Fields that let users select items from several pre-defined lists of items, where the item selected in first list determined which items will be availabe in the second list and so on. Displayed as a sequence of combo boxes:




To create a field of this type:

1. In the **Title** property, you specify the names of combo boxes separated by vertical pipe characters ("|"):

```
Product Area|Component|Version
```

2. In the **Selectable Items** property, you specify items of these combo boxes. Each line in the Selectable Items list contains the values separated by vertical pipe characters ("|"). The number of these values should match the number of values in the Title property:

```
Reports|Summary|0.0.1
Reports|Summary|0.0.2
Reports|Customer list|Alice var
Reports|Customer list|John var
Login screen|Controls|0.0.1a
Login screen|Controls|BS
```

When you select an item in the first combo box, Collaborator automatically filters the items of the second combo box keeping only those that are match the selection. When you select a value in the second combo box, the third combo boxes will have only items that match the selection, and so on.

When creating reports, the field values you specify in filters should include pipe characters, for example, "Login screen|Controls|0.0.1a".

## Multi-select list

These fields are displayed as list boxes that support multiple choices:


Use `Ctrl+click` or `Shift+click` for multi-selection.

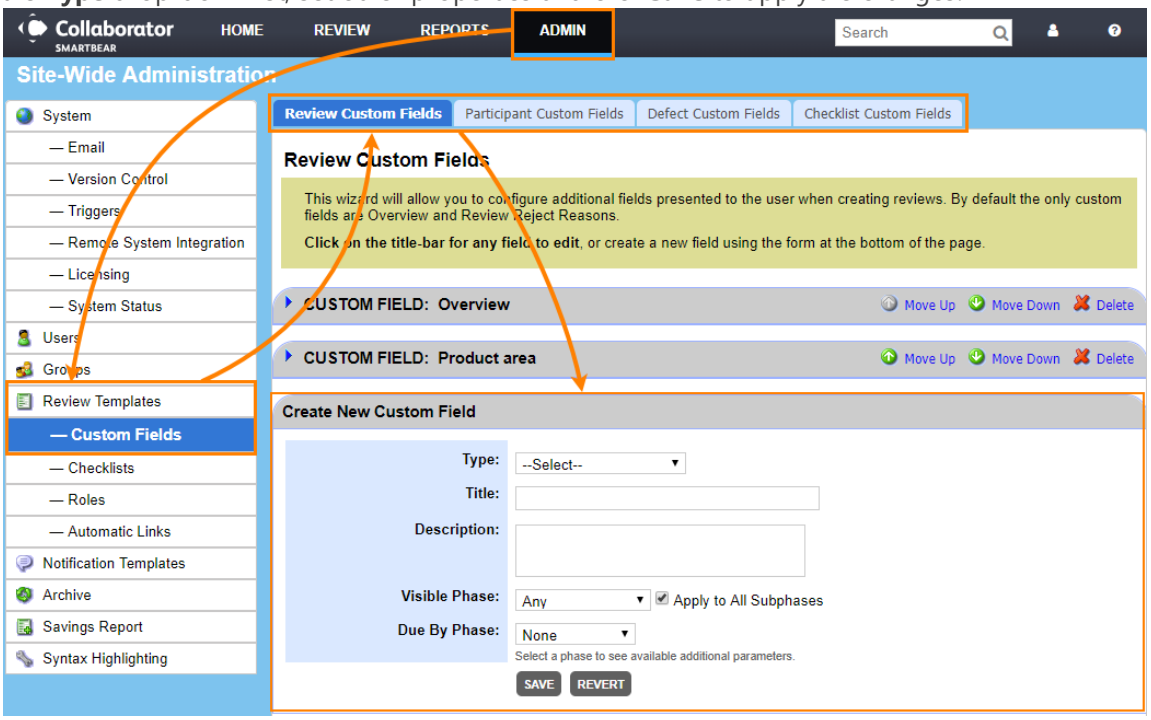
## Create a custom field

1. Log in to Collaborator as an administrator.

2. Go to **ADMIN > Review Templates > Custom Fields**. This page displays existing custom fields.

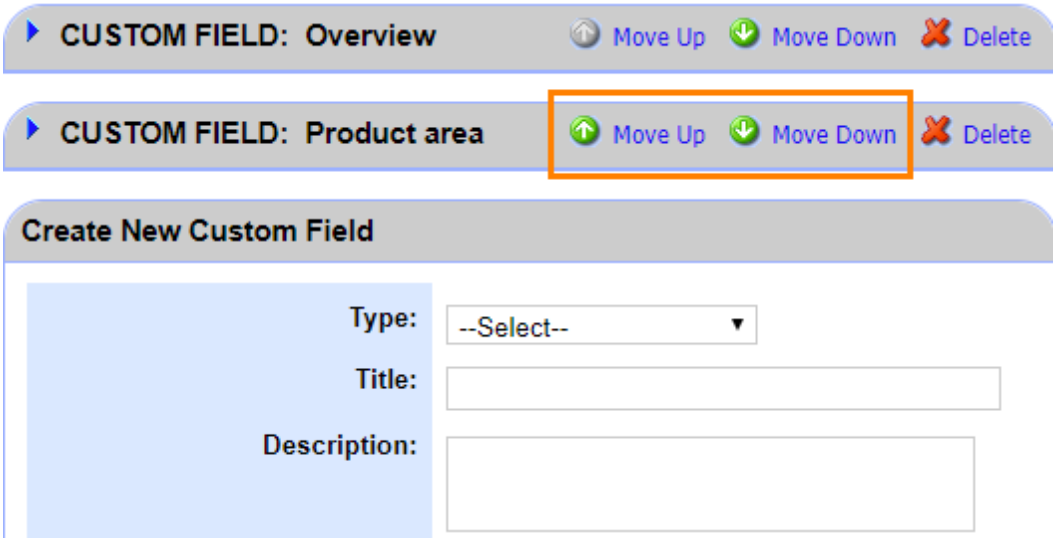
3. Switch to the tab that matches the type of the field you are going to create.

4. Find the Create new Custom Field section at the bottom of the tab, select the field type from the **Type** drop-down list, set other properties and click **Save** to apply the changes:




Important: The field type you set when creating a custom field cannot changed later.

5. Use the **Move Up** and **Move Down** buttons to position the new field among other custom fields:



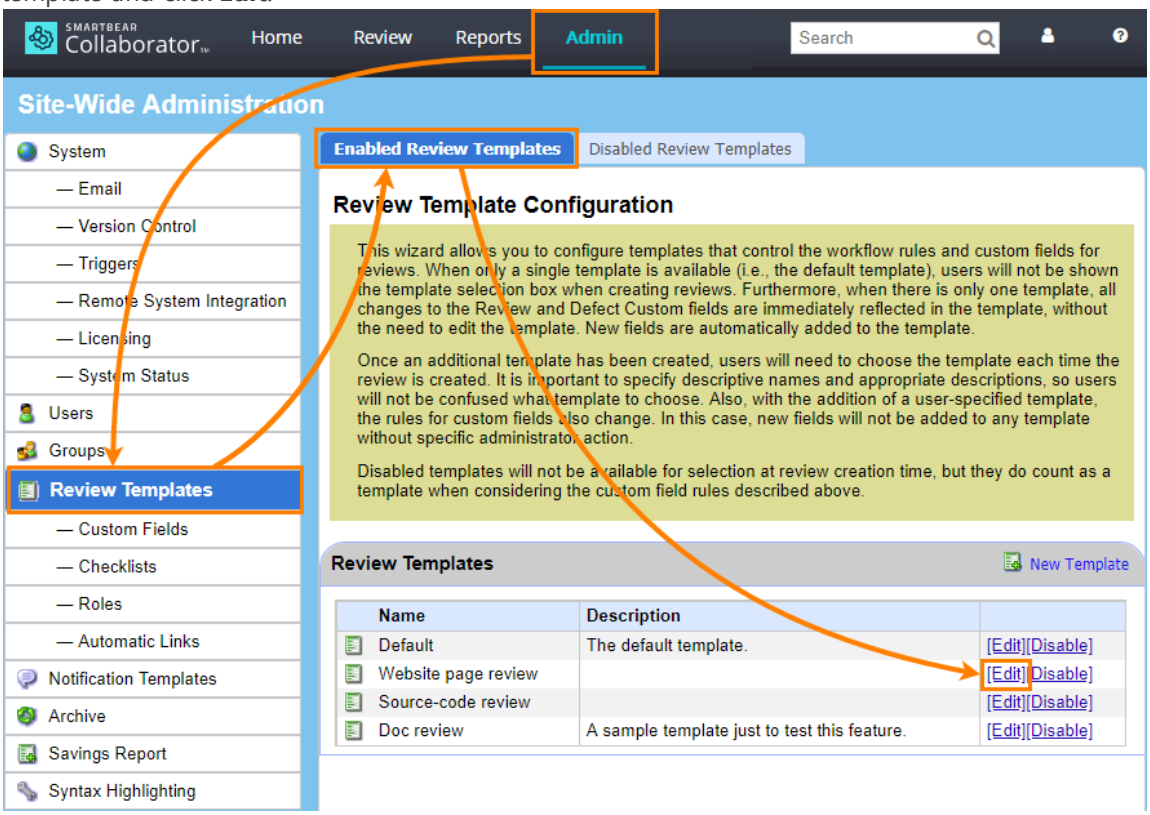

6. After you changed the parameters, you need to enable the custom fields in a review template (for review, defect and participant custom fields) or in a checklist (for checklist custom fields). See below.

## Enable custom fields in review templates

To use review, defect or participant custom fields in reviews, you need to add them to review templates you use. After doing this, custom fields will be available in new reviews based on these templates.

To add a field to a template:

1. Go to **ADMIN > Review Templates**. On the **Enabled Review Templates** tab, find the desired template and click **Edit**:




2. Scroll the template settings screen down to the **Review Custom Fields**, **Participant Custom Fields** or **Defect Custom Fields** section, and then select the desired custom fields to be available in templates:


3. Click **Save** to apply the changes, or click **Revert** to cancel them.

After you modified a template, Collaborator does not update all the reviews based on this template. It keeps existing reviews unchanged, and marks the template as disabled in them. It will apply the changes to new reviews that are the template.

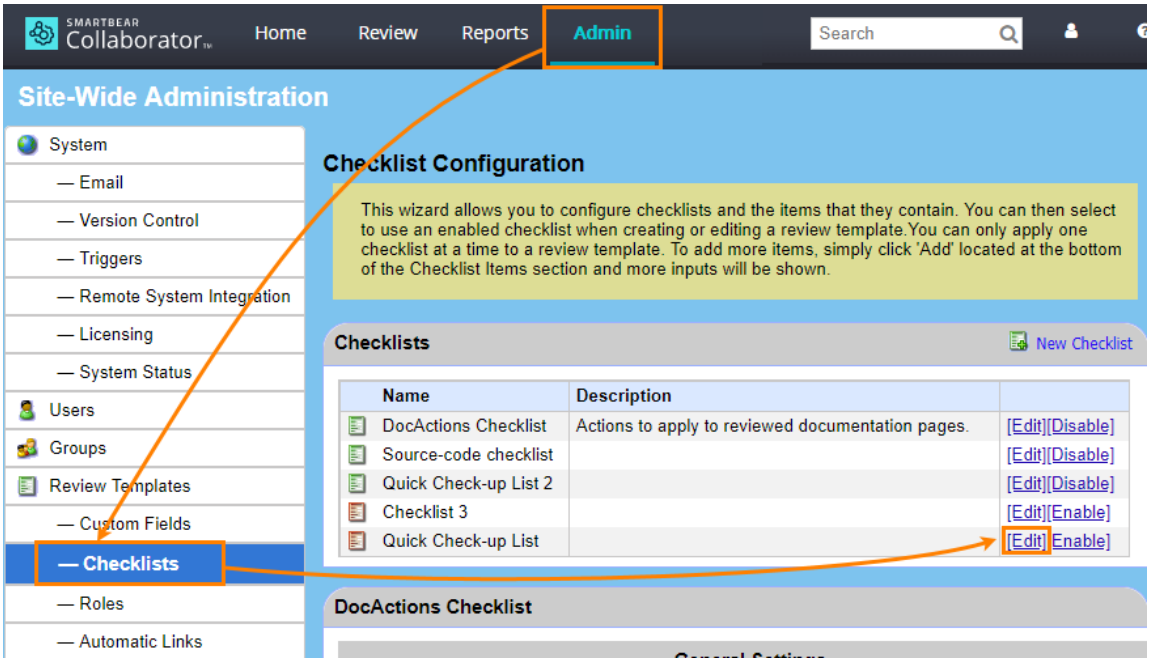
To use custom fields in existing reviews, you need to re-apply the template to these reviews. See Review Templates 246.

## Enable custom fields in checklists

To use checklist custom fields in reviews, you need to add them to the checklists you use. After doing this, custom fields will be available in new reviews that use the desired checklists.

To add a field to a checklist:

4. Go to **ADMIN > Review Templates > Checklists**. Find the desired checklist, if the checklist is active, then disable it, and then click **Edit**:

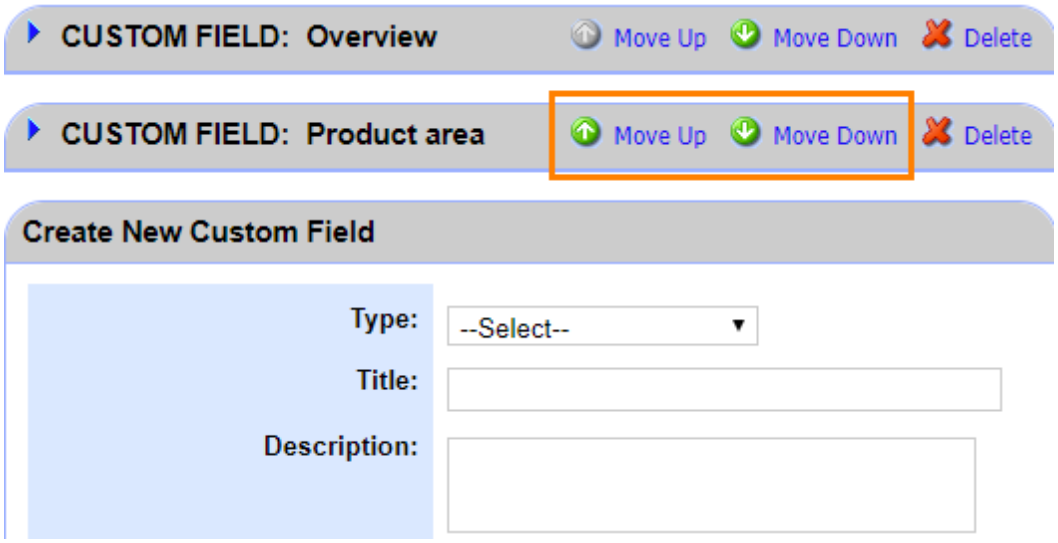



5. Scroll the checklist settings screen down to the **Checklist Custom Fields** section, and then select the desired custom fields to be available in the checklist:


6. Click **Save** to apply the changes, or click **Revert** to cancel them.

## Change the order of custom fields

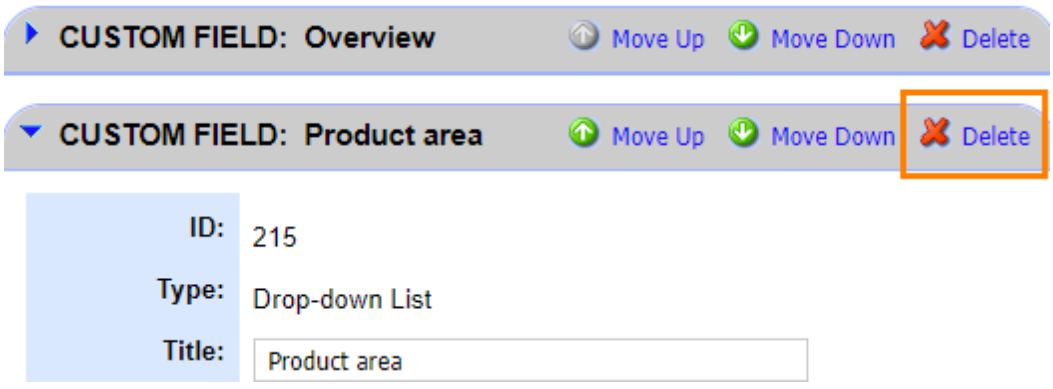
In the Custom Fields page, switch to the tab that matches the type of your custom field, and use the **Move Up** and **Move Down** buttons of the field to change its position:




There is no need to update review templates after this. The changes will be applied automatically.

## Delete a custom field

In the Custom Fields page, switch to the tab that matches the type of your custom field, and click **Delete** for the field:




There is no need to update review templates. Collaborator will remove the field from them automatically. Also, it will remove the field from all the reviews that use that field. The field values in these reviews will be lost.

# Custom field properties

## Common

| Property | Description |
|---|---|
| **Type** | The type of the custom field. See above 258. |
| **Title** | The name of the custom field, as it will be displayed to users.<br><br>Notes:<br><br>• To avoid user confusion, select descriptive names. Avoid using the names that are similar to some other existing field names.<br><br>• For custom fields of the drop-down series 258 type the Title property contains substrings separates by vertical pipe characters ("\|").<br><br>• If you host Collaborator on an Oracle server, note that custom field names which differ only by double-quote characters (") may break Oracle reporting views. See also Known Issues 979. |
| **Description** | Text of the tip that is shown below the field name in the UI. |
| **Visible Phase** | Specifies review phases 17, on what the field will be visible for users. To make the field visible from some phase, select that phase from the drop-down list and then select the **Apply to All Subphases** check box.<br><br>**Tip:** Never Visible means the field will not be displayed on any phase. These fields are useful for storing values provided by scripts and triggers.<br><br>For information on using participant custom fields on the Planning, Inspection and Completed phases, see below. |
| **Apply to All Subphases** | See above. |
| **Due By Phase** | Specifies the review phase on which the field must be populated. The possible values include:<br><br>**Value** — **Description**<br>*None* — The field is not required on any phase and may remain blank. |

| | |
|---|---|
| | <table><tr><td>*Planning*</td><td>The field must be populated on the Planning phase.</td></tr><tr><td>*Annotating*</td><td>The field must be filled-in on the Annotating phase. Since that phase is optional, the field may remain blank, if a review has never been in that phase.</td></tr><tr><td>*Rework*</td><td>The field must be filled-in on the Rework phase. Since that phase is optional, the field may remain blank if a review has never been in that phase.</td></tr><tr><td>*Inspection*</td><td>The field must be populated on the Inspection phase, no matter whether a review goes then to the Rework or to the Completed phase.</td></tr><tr><td>*Completed*</td><td>The field must be filled-in on the Inspection phase only if a review goes to the Completed phase. If a review goes to the Rework phase, the field may remain blank.</td></tr><tr><td>*Any*</td><td>The field must be populated when a review is moving from any phase to any other phase.</td></tr></table><br>To avoid situations when a required field becomes invisible, make sure that —<br>• Visible Phase is the same phase as Due by Phase,<br>— or —<br>• Visible Phase is an earlier phase relative to Due by Phase, and Apply to All Subphases is selected.<br>If Due by Phase is Completed, Visible Phase must be some earlier phase.<br>Required custom fields of the participant type are only required for participants that are expected to change the review on some phase: for example, on the Planning phase these are authors, on the Annotating phase – all participants, on the Inspection phase – reviewers, and so on. |
| **Allowed to Modify** | Available for the participant custom fields only. Defines who can modify the field value: the participant who is supposed to do this, any participant, moderator and so on. Default is Assigned participant. |

Other properties are specific to the field type (see below)

### Single-line text

| Property | Description |
|---|---|
| **Minimum Length** | Minimal allowed length of the text in the field.<br><br>To let users have empty values in a field, set this property to 0. |
| **Maximum Length** | Maximum number of characters a field can contain.<br><br>The maximum allowed value here is 255. The default value is also 255. |
| **Validator** | A Java-style regular expression that will be used to validate the contents of a field. The field value should match the specified regular expression. |

### Multi-line text

| Property | Description |
|---|---|
| **Minimum Length** | Minimal allowed length of the text in the field.<br><br>To let users have empty values in a field, set this property to 0. |
| **Maximum Length** | Maximum allowed number of characters in the field.<br><br>This value depends on the database server, where Collaborator is installed. In many cases, the lower possible bound is 8192, but it can be higher and can reach thousands of characters. |

### Drop-down List

| Property | Description |
|---|---|
| **Selectable Items** | The list of items to be displayed in the drop-down list. |
| **Display Ordering** | The order of items in the list: ascending, descending, or none (which means the items will be displayed in the order of appearance in the Selectable Items list). |
| **Default Value** | The value that the field will have by default. Should match one of the items in the Selectable Items list. |

Drop-down Series

| Property | Description |
|---|---|
| **Selectable Items** | The list of items of the of the drop-down lists.<br><br>Each item in the list includes substrings separated by vertical pipe characters ("\|"). The substrings correspond to items of different combo boxes in the "drop-down series" field. For example:<br>`DesktopClient\|New controls\|ver. A`<br>`DesktopClient\|New controls\|ver. B`<br>`WebClient\|Controls\|0.3a`<br>`WebClient\|Controls\|0.4`<br>`WebClient\|Wizards\|Create`<br><br>Items in the list must have the same number of substrings, and this number should match the number of substring in the Title property. |
| **Display Ordering** | The order of items in the list: ascending, descending, or none (which means the items will be displayed in the order of their appearance in Selectable Items). |

### Multi-select List

| Property | Description |
|---|---|
| **Selectable Items** | The list of items to be displayed in the drop-down list. |
| **Display Ordering** | The order of items in the list: ascending, descending, or none (which means the items will be displayed in the order of appearance in the Selectable Items list). |

## Notes

### Predefined custom fields

Some custom fields are available out-of-box:

- Overview (review custom field)
- Severity (defect custom field)
- Type (defect custom field)

Collaborator does not impose any limitations on these fields. You can edit them properties as if they were created by you.

## Custom fields with date and time values

Collaborator does not offer special fields for date and time values. To store these values, you can use custom fields of the String (single-line) type and use the following regular expressions to validate user input:

- For dates, you can use a regular expression like this:

```
(?:19|20)\d\d-(?:0[1-9]|1[012])-(?:0[1-9]|[12][0-9]|3[01])
```

It lets users enter a data in the 20th or 21st century, and yes, it accepts 31 days each month.

- For time values, you can use the following regular expression:

```
(?:[01][0-9]|2[0-3]):[0-5][0-9]
```

It validates time values against a 24-hour format.

You can combine these two regular expressions to accept both date and time in the same field.

Set the field description to inform users about the expected format, so that users they do not have to parse regular expressions to know what to enter.

## Changing custom field values

Custom fields are used in reviews. Changing field properties might cause reviews to contain invalid data. To prevent this, Collaborator follows these rules:

- Collaborator does not include new custom fields in reviews automatically. First, you need to add them to review templates. This operation disables the existing review template and creates a new template in place of it. So, the changes will be visible in new reviews that are based on the updated template.
  If you want the changes to be applied to existing reviews, you will have to reapply the new template to them. Existing values of custom fields will be retained upon changing review template if the custom field exists in both templates. The new fields will contain default values, or will show a dash or N/A to indicate that the value was not given.

- If you change properties of a field that is already present in a template, then —

  - If the change you made does not affect field values (for example, you renamed a field or changed its description), then users will see the changes next time they open the review in the review screen.

  - If the changes made existing values invalid (for instance, you decreased the maximum length, changed the validating regular expression or removed some items from a drop-down list), the existing field values will remain unchanged. However, users will have to specify new valid values when they are editing a review.

- Custom fields that you delete in ADMIN settings will be removed from the UI automatically. Now update of review templates is needed.

## Custom fields in external reporting systems

To make custom fields available for external reporting systems like Excel, Access, Crystal Reports, or Business Objects, Collaborator offers two view objects: reviewcustom and defectcustom.

The reviewcustom view does not includes custom fields of the Multi-select list type.

If a review does not use some custom fields (these fields are disabled in the corresponding review template), then these fields have the NULL value in reports for these reviews.

We recommend to refer to custom fields in reports by their names rather than by their order.

## Why are my custom fields absent from reviews?

New custom fields are not added to reviews automatically. You need to add them to review templates you are using.

Note that this will disable the review template and will create a new template in place of it. Your new custom fields will be visible in new reviews based on the updated template, and will not be visible in existing reviews. If you want the see the new field in an existing review, you will have to reapply the new template to that review.

## Using participant custom fields

- Participant custom fields cannot be visible only in the Planning phase. Otherwise, participants will not be able to specify field values.

- If a required participant custom field is visible only on the Inspection phase, users cannot "finish" this phase without specifying their values for the fields.

- Users are allowed to edit participant custom fields when the review is in the Completed phase without reopening the review. This gives them a possibility to answer post-review survey questions.

- Follow these links to get examples of reporting on participant custom fields:

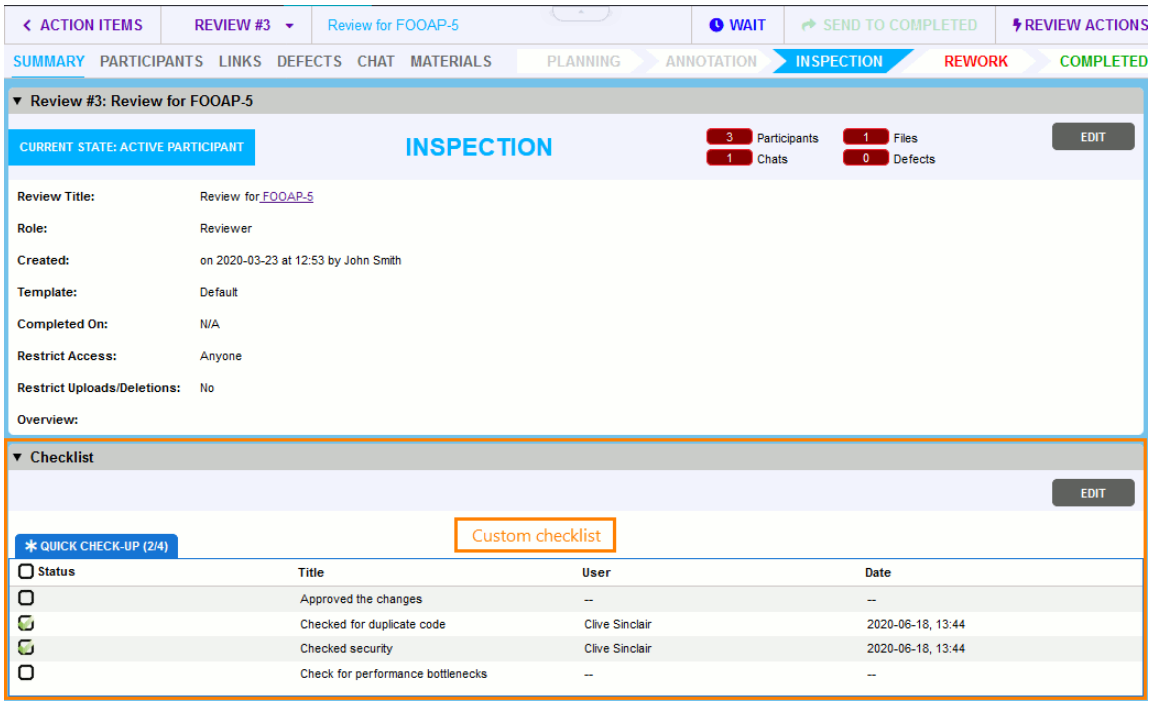
  - Example SQL: Participant Custom Fields 917

  - Example XPath and XSL 923

## 3.3.13 Checklists

### About checklists

Checklists are user-defined "to-do" lists that you can apply to reviews. You can use them to remind review participants to perform certain actions before the review can be completed:




To specify which checklist should be used for this or that review, modify the review template 246. You can use different checklists for different review templates.

You can create multiple checklists. To specify the checklists to use in this or that review, modify the templates 246 of these reviews. You can use different checklists for different review templates. One template can use one or multiple checklists. See Review Screen - Checklists 350.

Collaborator logs all user activities on the checklist. When a review participant select a check box in the list, Collaborator logs the participant's user name and the timestamp of the change. Also, it adds a message to the Chat section of the review. If a review participant clears some check box in the list, Collaborator also adds a message about this to the Chat section.

For auditing purposes, you can include checklist status changes into the Review Details Report 473. Make sure that **Checklist History** is set to *Display Checklist History* for this.

### Requirements

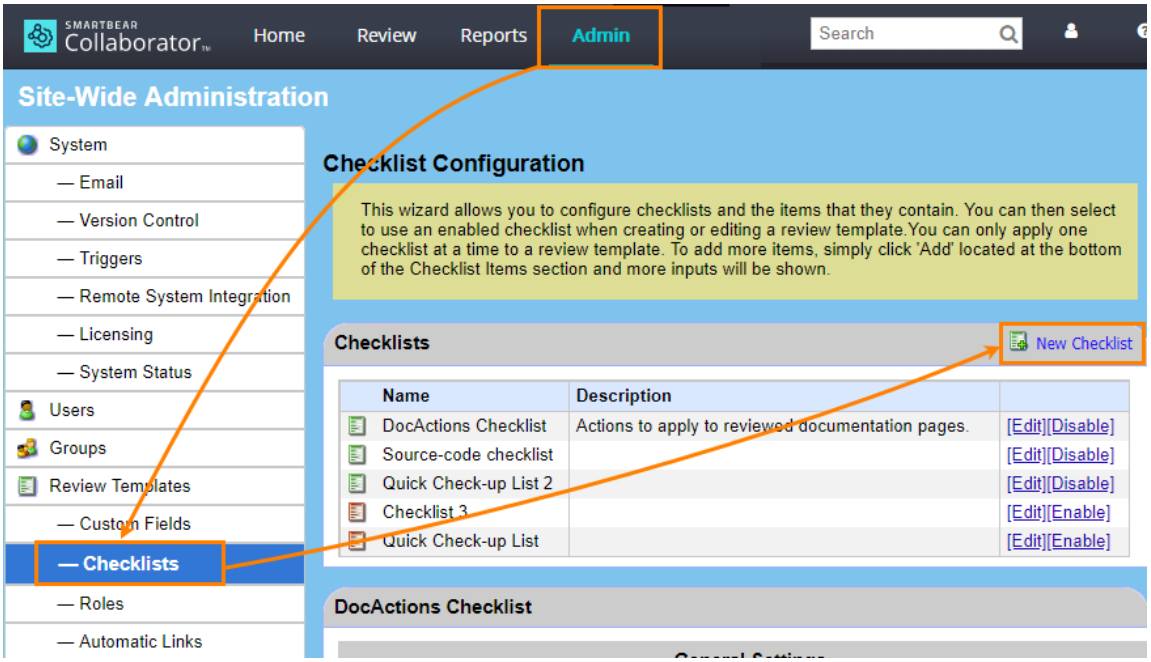
- Checklists are available in Collaborator Team and Collaborator Enterprise. For a complete list of differences between Collaborator editions, see Collaborator Editions 3.

- To create and edit checklists, you need administrator permissions in Collaborator.

# Create checklist

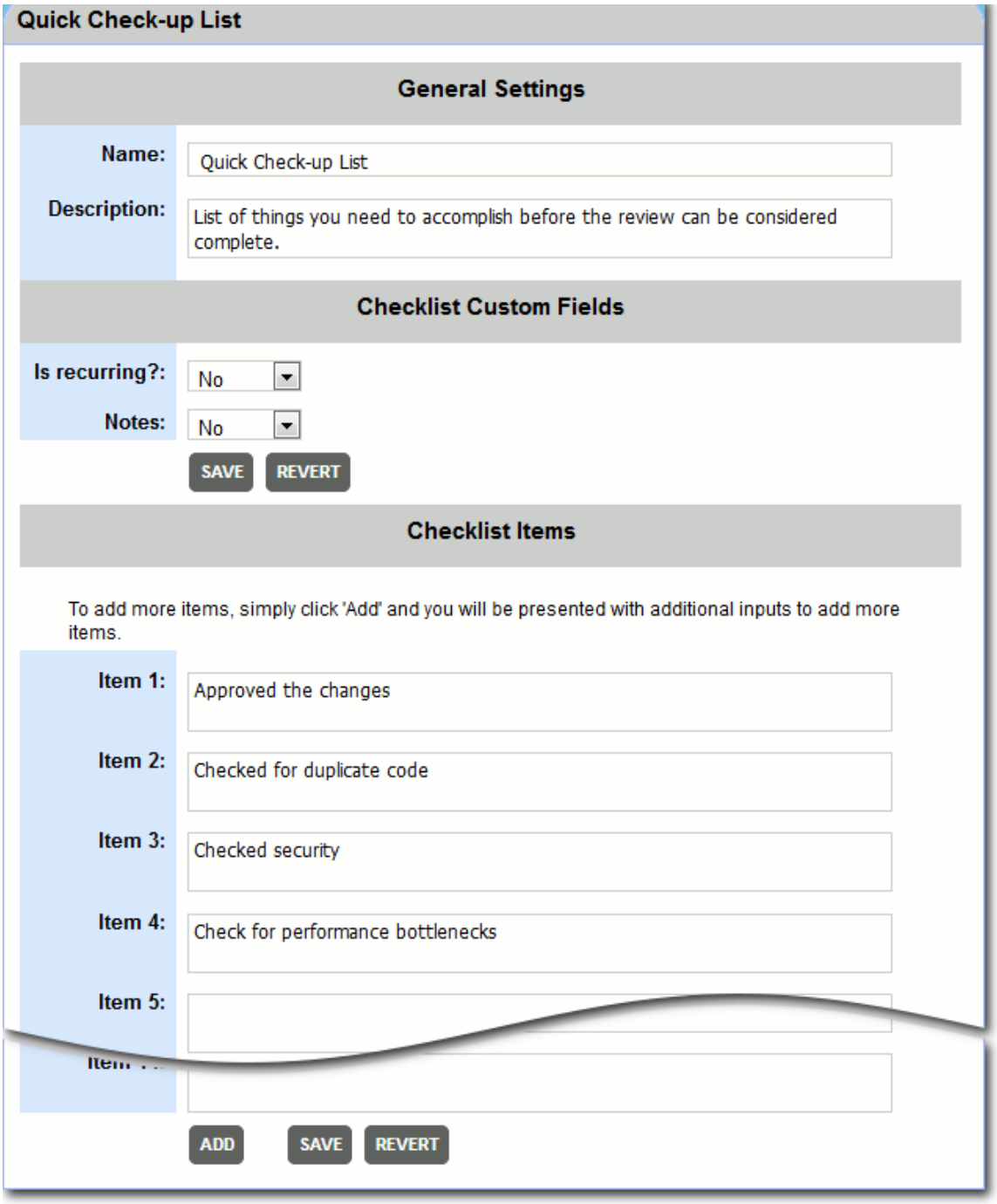
## 1. Create a new checklist

1. Log in to Collaborator as administrator.
2. On the main toolbar, click **ADMIN** and then go to the **Review Templates** > **Checklists** settings.
3. Click **New Checklist**:

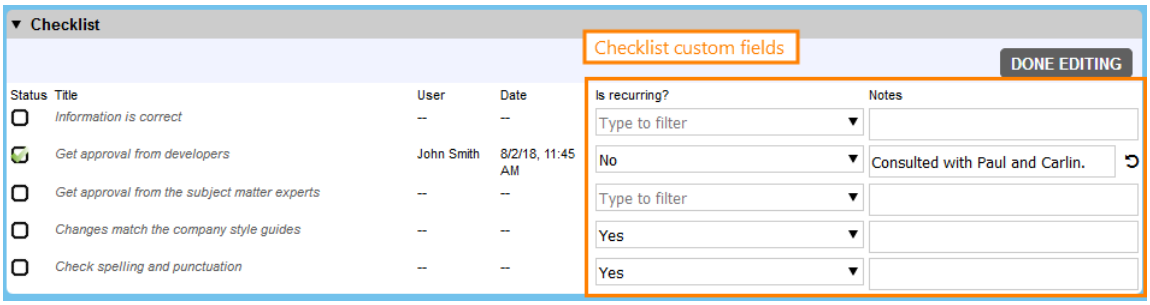



4. On the subsequent screen, specify the checklist name and arbitrary description. They will visible in the Collaborator settings.
   Checklist names must be unique.

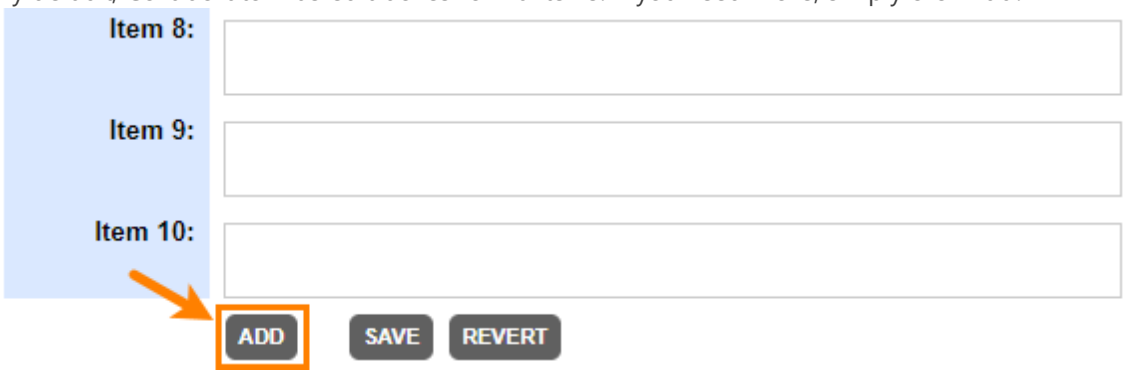
5. If you have created any checklist custom fields[258], specify which of them should be displayed for this checklist:




**Note:** Custom fields appear as additional columns in a checklist, for example:




6. Enter the list items in the **Item nn** edit boxes.
   Checklist items must be unique within a checklist.
   By default, Collaborator has edit boxes for 10 items. If you need more, simply click **Add**:



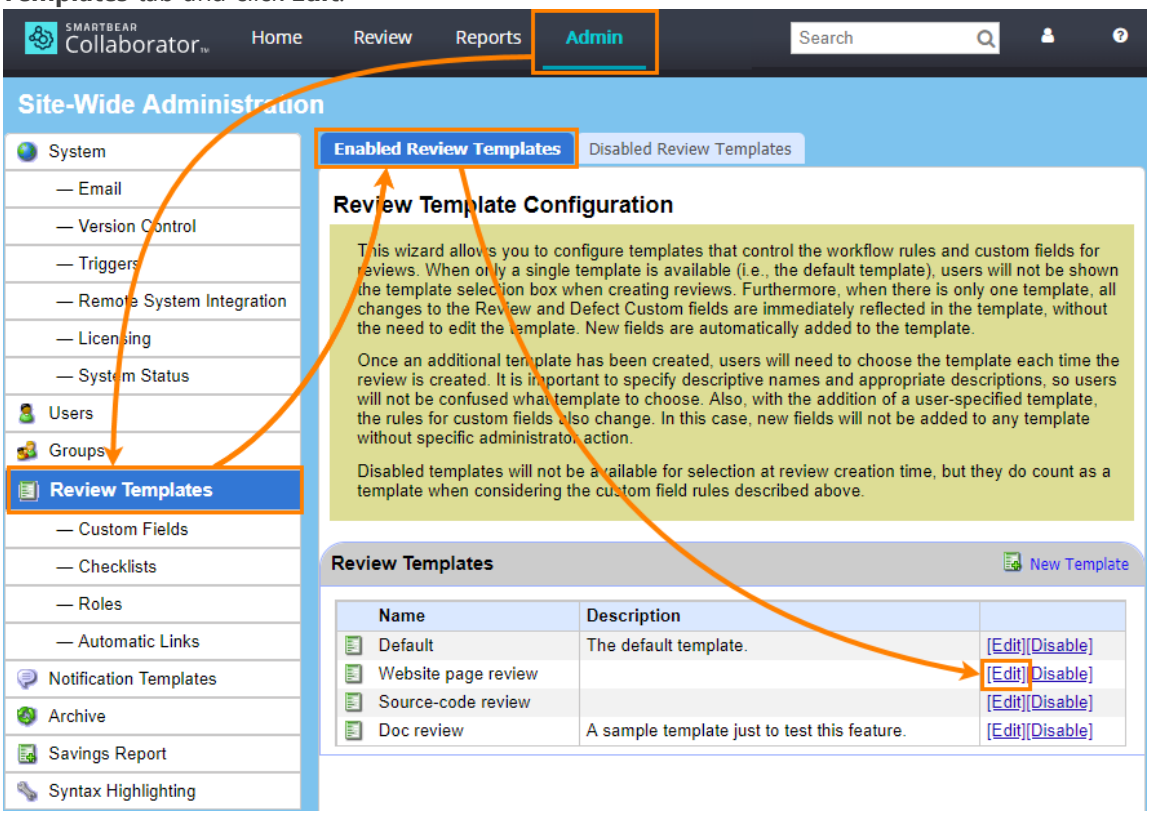

   If you don't need some item, simply clear its edit box. Collaborator will not display empty items on the review screen.

7. Click **Save** to apply the changes.

## 2. Add the checklist to a review template

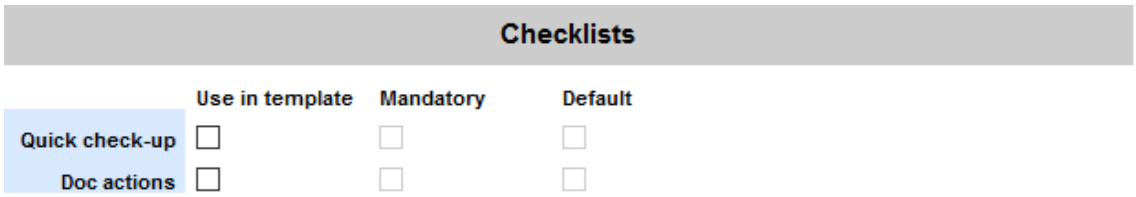
After you have created the checklist, apply the checklist to a template:

1. Go to **ADMIN > Review Templates**. Find the desired template on the **Enabled Review Templates** tab and click **Edit**:




2. In the template settings screen, find the desired checklist and specify its properties:

| Parameter | Description |
|---|---|
| **<Checklist_ Name>: Use in review** | Specifies which checklists should be available to reviews that use this template.<br><br>**Note:** The checklist will be available in new reviews. It is not added to existing reviews based on the template. |
| **<Checklist_ Name>: Mandatory** | Specifies whether all items must be checked before the review can be approved. |
| **<Checklist_ Name>: Default** | Right after you create a review, the Review Screen displays only one checklist. Review participants append other checklists later, if needed. This setting specifies the checklist to be visible by default. If no checklist is selected as default, the Review Screen will show the first created checklist. |



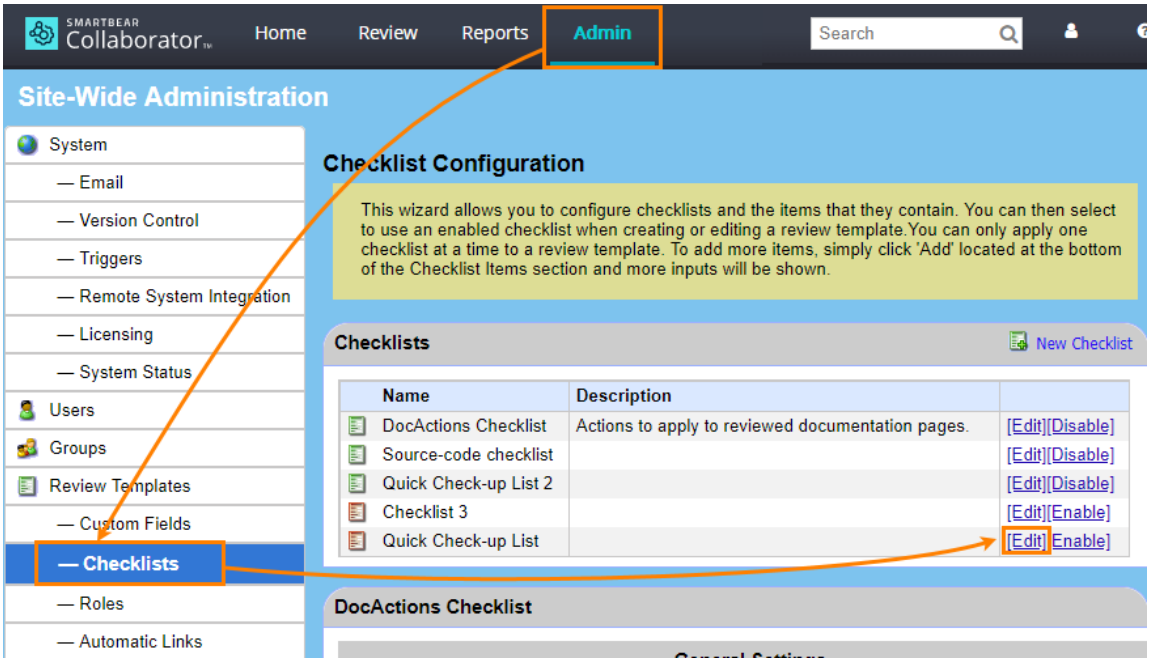

3. Click **Save** to apply the changes, or click **Revert** to cancel them.

## Edit checklist

1. Log in to Collaborator as administrator.

2. On the main toolbar, click **ADMIN** and then go to the **Review Templates** > **Checklists** settings.

3. Find the needed checklist in the list and click **Edit**:




4. On the subsequent screen, change the checklist parameters, and modify the text of list items. **To delete an item**, clear its edit box. Collaborator will not display empty edit boxes on the review screen.

5. Click **Save** to apply the changes, or click **Revert** to cancel them.

If you have reviews that use your checklist, then you are allowed to change the items only: you can edit their text, delete items or append new items to the list. If the checklist is not used in any review, you can also change the checklist name, description and custom fields. To use a checklist for existing reviews, re-apply the template 246 to them.

## Disable and enable checklists

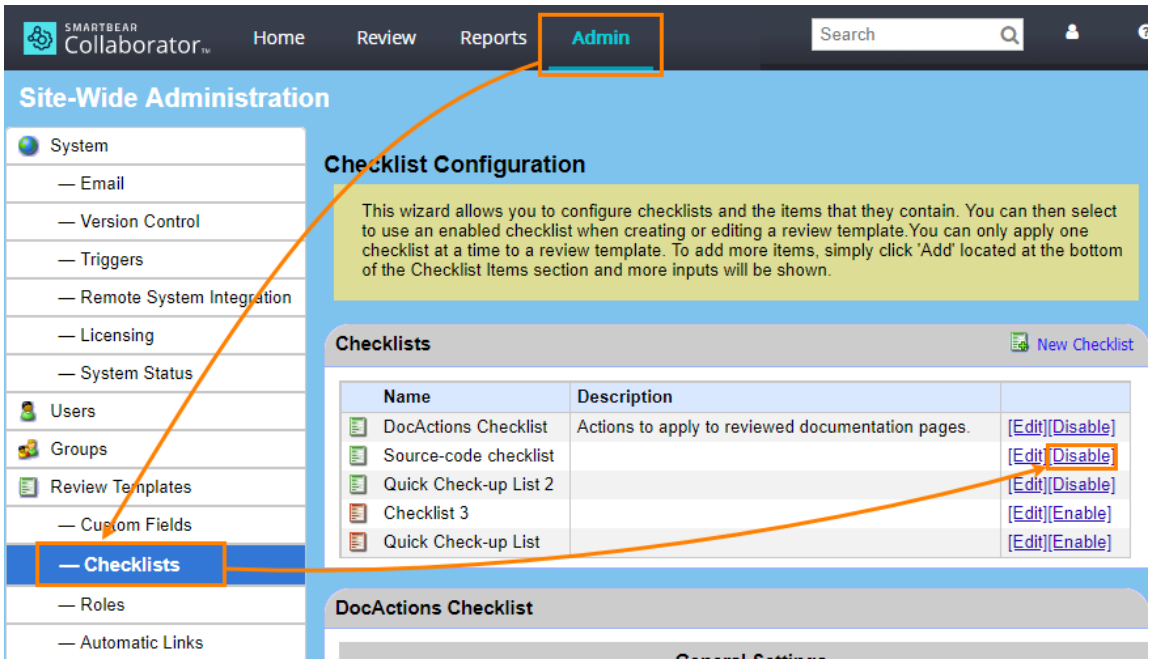
You cannot delete checklists in Collaborator. To make some checklist unavailable, you can disable it. Disabled checklists are not displayed in reviews.

To disable a checklist:

1.Log in to Collaborator as administrator.

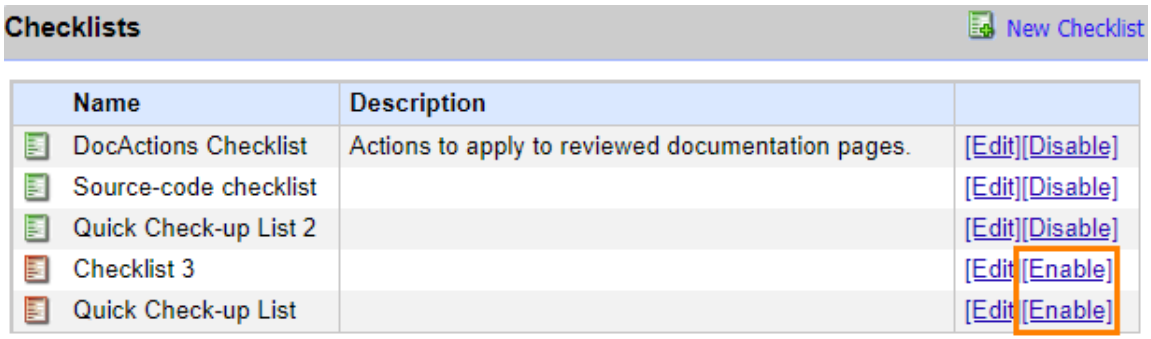
2.Go to **ADMIN > Review Templates > Checklists**.

3.Find the needed checklist in the list and click **Disable**:




**Note:** If you disable a checklist after it has been applied to some reviews, Collaborator will not update these existing reviews. The changes will be applied to future reviews. To use a checklist for existing reviews, re-apply the template to them.

After you disabled a checklist, Collaborator will change the link text to “Enable”. To enable a checklist, simply click this **Enable** link:




## 3.3.14 Roles

A "Role" in Collaborator indicates the way in which a person participates in a review.

***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

### Roles

Collaborator supports four "roles" for review participants. The following roles are configured by default but administrators can configure completely different templates.

**Author**

Authors are the people responsible for the changes or documents under review. Usually they are responding to comments and questions made by reviewers and observers. In the "Fix Defects" phase the review will be "in their court" and on their Action Items 333 list.

**Reviewer**

Reviewers are responsible for inspection, creating defects, and so on. With multiple reviewers, they will hopefully come to a consensus about each questionable item, but this is not required by the system. Reviewers typically create defects 436, but again this is not required.

**Observer**

Observers are involved and make comments but they are not "vital" to the review. If all other participants mark the review "complete," the review goes to the next phase regardless of observer state. Observer roles are usually used when you want to bring in someone who has special expertise on an issue.

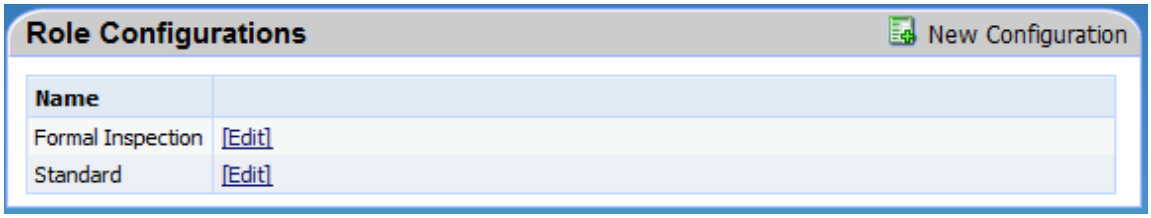
**Moderator** (optional, not enabled by default)

The Moderator maintains the pace and tenor of the review. This is an optional role that does not exist in the default installation. This is used for more formal review workflows where one person leads and controls the review.

Different styles of review require different roles with different terminology and rules for what each rule is allowed to see and do in a review. Formal reviews might have four roles with strict rules, informal reviews might have just an author and reviewer, and a "self-check" review might just require an author with optional external reviewer.

Users interact with roles when they are creating a new review[338] or editing the list of participants[352] in a review.

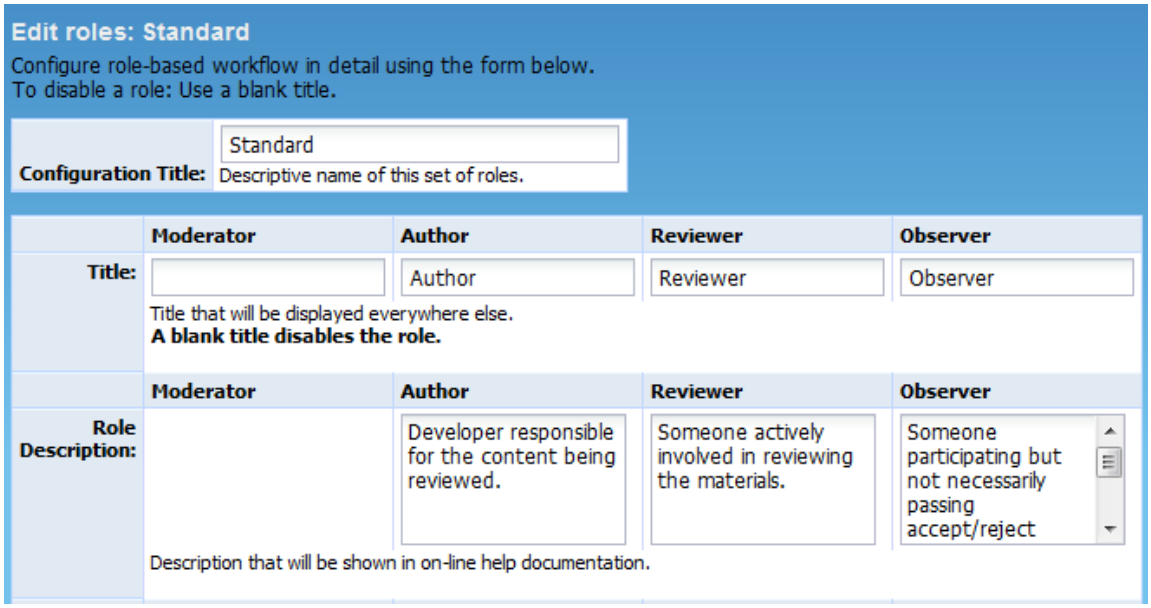
The Role Configuration screen lets you set up any number of sets of roles. Each set is given a name and corresponds to some concept of a review.




Any number of role configurations can be specified. The names here are never shown to an end user; they are used only when selecting a role configuration as part of a review workflow[246].

In each role configuration you can have between 1 and 4 roles.

A variety of options are available when editing an existing role configuration or creating a new one:



**Configuration Title**

The title of this configuration. This is strictly an administrative thing -- users of the system will never see this title. They see review workflows 246 instead.

**Title**

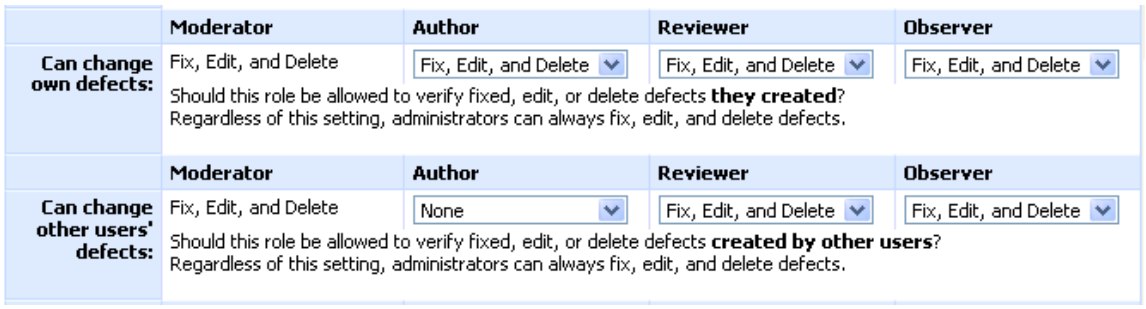
The name of the role as it will appear in drop-down lists and on-line help text. The user will use this title as the primary mechanism for referring to the role.

Use a blank title to indicate that this role should be completely disabled.

The "Author" role *must* always be in the second column, although it can be renamed. It *must* always exist. The concept of the "author" is a special one in the system because the author always has special responsibilities. For example, when a review is in "Rework" phase the author's Action Items 333 list will say "Rework defects found" whereas other participants will see "(No Action Required) Waiting for author to rework defects".

**Description**

A description of the role that will be prominently displayed to the user in all client user interfaces where roles are chosen, most prominently in the Participants 338 section of the Create Review wizard.




**Can change own defects**

If "None", this role will be allowed to create defects, but will not be allowed to mark their own defects fixed, edit their own defects, or delete their own defects.

If set to "Edit", this role will be able to create defects and edit their own defects, but will not be allowed to mark their own defects fixed or delete their own defects.

If set to "Fix and Edit", this role will be able to create defects, mark their own defects fixed, and edit their own defects, but will not be allowed to delete their own defects.

If set to "Fix, Edit, and Delete", this role will be able to create defects, mark their own defects fixed, edit their own defects, and delete their own defects

Administrators can always change or delete any defect, regardless of this setting.

In all but the most informal of reviews, authors are typically not allowed to mark defects fixed, but all other roles are allowed.

This setting also applies to the ability to externalize[442] a defect.

**Can change other users' defects**

If "None", this role will not be allowed to mark other users' defects fixed, edit other users' defects, or delete other users' defects.

If set to "Fix", this role will be able to mark other users' defects fixed, but will not be able to edit or delete other users' defects.

If set to "Edit", this role will be able to edit other users' defects, but will not be allowed to mark other users' defects fixed, or delete other users' defects.

If set to "Fix and Edit", this role will be able to mark other users' defects fixed and edit other users' defects, but will not be allowed to delete other users' defects.

If set to "Fix, Edit, and Delete", this role will be to mark other users' defects fixed, edit other users' defects, and delete other users' defects

Administrators can always change or delete any defect, regardless of this setting.

To enable this setting, the Restrict Access to Fix Defect[187] setting must be set to "No".

This setting also applies to the ability to externalize[442] a defect.

| | Moderator | Author | Reviewer | Observer |
|---|---|---|---|---|
| **Marks reviews "finished":** | No | No | Yes | Yes |

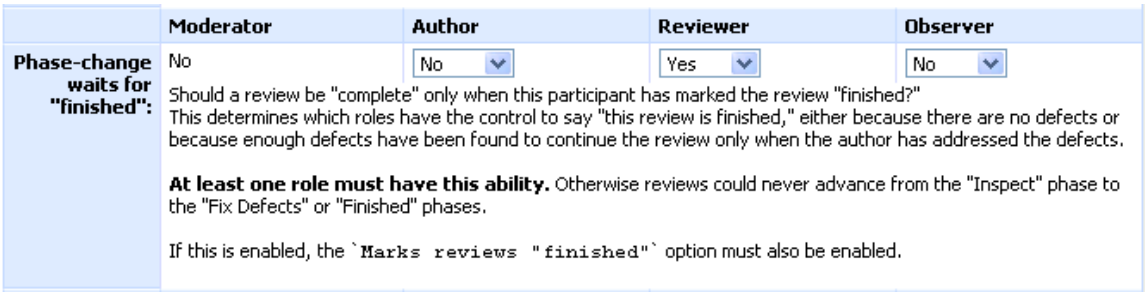
Should this role be required to indicate when the review is "finished"?
Participants who have marked a review "finished" show up to other participants with "Approved" in the State column of the Participants section on the row corresponding to their name.
This should be enabled for roles who have an obligation to do something in the review, e.g. open defects or make comments.

**Marks reviews "finished"**

If true, users with this role will be presented with a button in the Next Steps screen[347] that allows the user to say to the group: "I am finished looking at this review". The user will be brought back into the review if another user makes a non-trivial comment.

If false, the user is told "You must wait for other users".

Typically everyone except the author has this feature enabled. The author is usually not in control of when the review finished so his "I am finished" decision is not interesting.

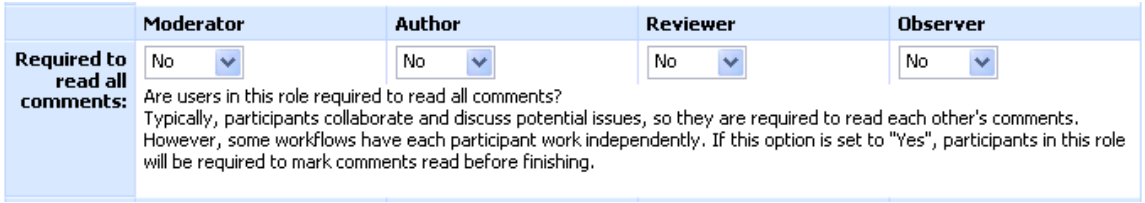



**Phase-change waits for "finished"**

Specifies if a review phase change requires that participants with this role complete reviewing. This applies to the “Inspection -> Rework” and “Inspection -> Completed” changes. If this parameter is Yes and some of participants with this role have not yet approved the review, Collaborator will not change the phase.

Typically, you enable this parameter for *Reviewers* and disable for *Observers*.

This parameter has effect only if **Marks reviews "finished"** is also enabled.



**Required to read all comments**

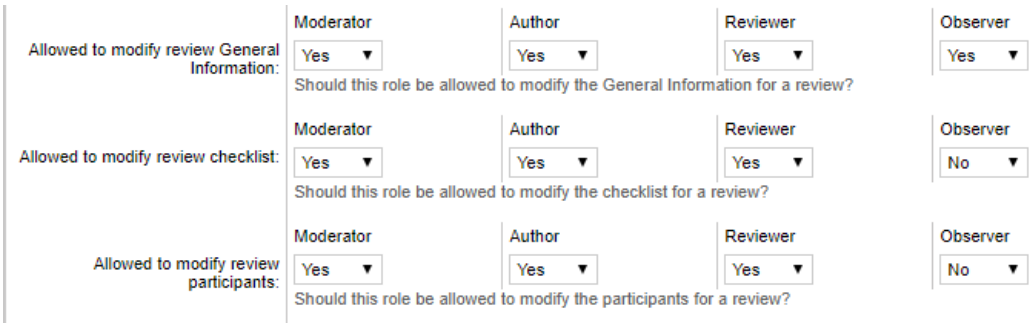
Please note that "Marks reviews finished" and "Phase-change waits for finished" must both be set to "yes" in order for this setting to be evaluated for the selected roles. Changes here will have no effect if one or both of those settings are set to "no" for the specified role.

If yes, participants in this role will be required to mark comments read before finishing the review.

Typically participants collaborate and discuss potential issues, so they are required to read each other's comments. However, some workflows have each participant work independently.

Setting this configuration to "no" is usually applied to observer-type roles because they are frequently called in for something specific and do not need to "hang around" and see what other users have to say. This is also helpful in workflows where the reviewers independently submit comments to the author, who is then in charge of all changes without having to communicate back to reviewers.

Note that this setting does not prevent anyone from coming back to the review and continuing to actively participate at any time.




**Allowed to modify review General Information**

Should this role be allowed to modify the review title, template, custom fields and other data of the General Information[348] section?

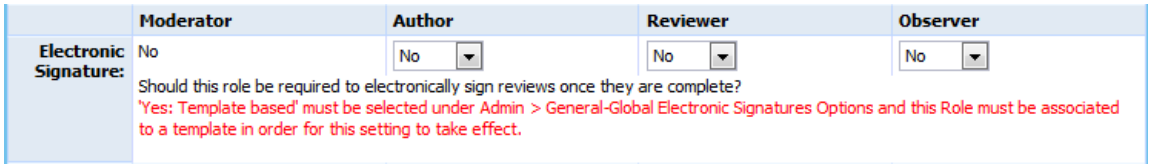
Typically there is no harm in allowing this, but you might want to lock down exactly who has the last word on the review custom fields and title.

**Allowed to modify review checklist**

Should this role be allowed to modify the checklist for a review[272]?

**Allowed to modify review participants**

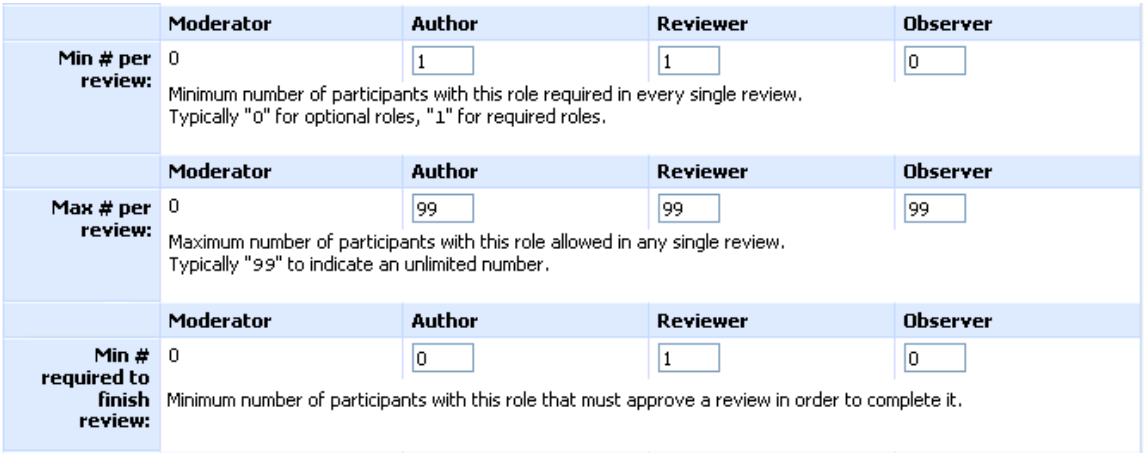
Should this role be allowed to modify the participants for a review 352?




**Electronic Signature**

Should this role be required to either sign off on or decline to sign off on a completed review?

This setting takes effect when template based electronic signatures 193 are enabled in the General Settings 180 screen.




**Min # per review**

The minimum number of users with this role required in the review. Can be zero to indicate "This role completely optional". Most often used with reviewers or moderators to require at least one, or with a "very careful" review where multiple reviewers are required.

**Max # per review**

The maximum number of users with this role required in the review. The maximum supported number per role is 999. Usually there is no reason to limit the number of users who can participate as a role in a review, except for highly formal reviews where a strict process is required.

**Min # required to finish review**

The minimum number of users with this role required to approve the review in order to complete it.

This value should be less than or equal to a value specified in "Min # per review" for the respective role.

If the user role allows to mark the review as finished and this value is 0, then all users with this role must approve the review to "close" it. If the user role is not obligated to approve reviews and this value is 0, then all users with this role are not required to "close" a review.

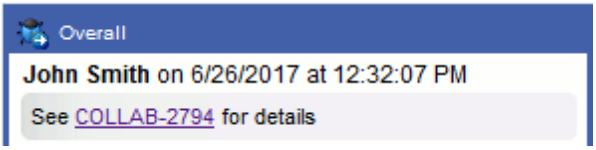
## 3.3.15 Automatic Links

Collaborator creates links in text that is typed by users as review comments, custom fields, and so on. Several types of links are created automatically:

- URL's create links to that URL
- Names of files in the review create links that open that file in the Diff Viewer
- Line numbers (for example, 'line 345') in a file comment create links to that line in the Diff Viewer

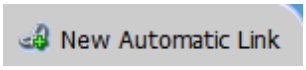
***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

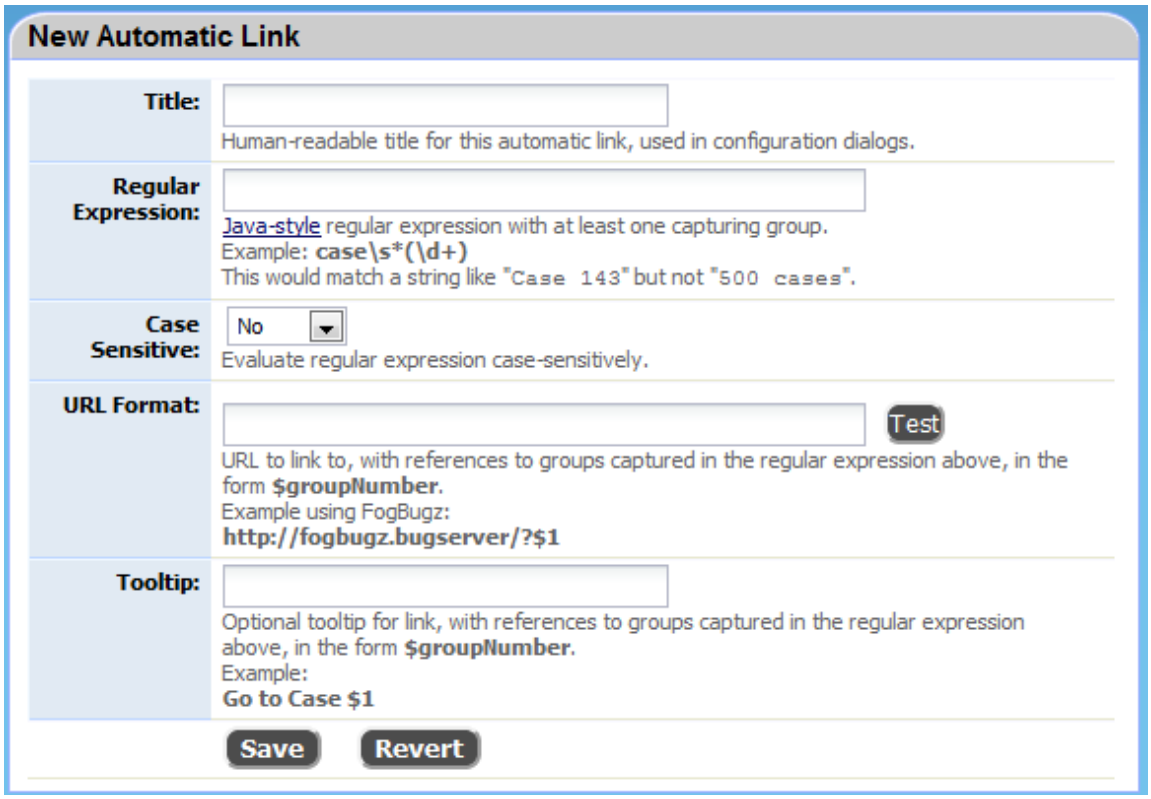
You can define your own Automatic Links to integrate Collaborator with external systems. Each Automatic Link consists of a regular expression pattern, URL format, and optionally a tooltip. When a match is found, Collaborator creates a hyperlink around the matching text. The URL of the hyperlink and the tooltip are constructed by replacing references to groups with the text selected by the corresponding capturing groups in the regular expression.




Examples of external systems you might want to link to include issue/bug trackers, build tools, or documentation library systems.

### Creating New Automatic Links

To create a new Automatic Link, click the *New Automatic Link* button. 




**Title** Human-readable title for this automatic link, only used in the Automatic Link Configuration page.

**Regular Expression** A Java-style regular expression that identifies a reference to the external tool in text. This regular expression will be used to find matches in user-entered text in review titles, custom fields, comments, and defects.

You must include at least one one grouping expression (parenthesis). This is combined with the URL Format to link to construct the hyperlink to the external tool.

For example: `case\s*(\d+)`

This would match a string like `Case 1423` but not `500 cases`. This is the standard way to refer to a bug in FogBugz.

**Case Sensitive** Evaluate regular expression case-sensitively. Usually this is set to "no".

**URL Format** URL to link to, with references to groups captured in the regular expression above, in the form *$groupNumber*.

Example using FogBugz:

| | |
|---|---|
| | http://fogbugz.bugserver/?$1 |
| **Tooltip** | Optional tooltip for link, with references to groups captured in the regular expression above, in the form *$groupNumber*.<br><br>Example:<br><br>Go to Case $1 |

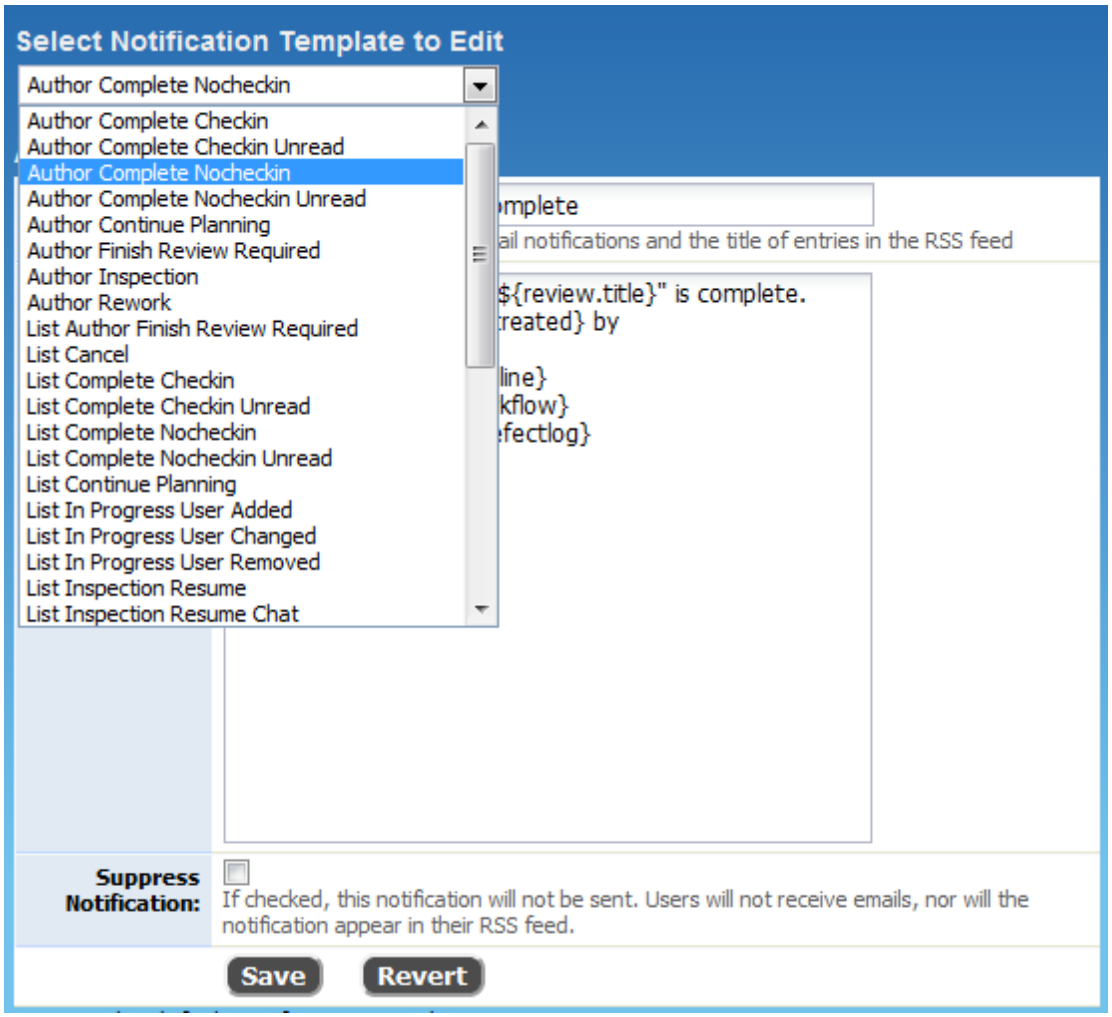
## 3.3.16 Notification Templates

Using settings on the **Notification Templates** page, you can customize the content of notification e-mails that Collaborator sends to review participants (see Notifications on Review Changes 329).

***Note:*** *This feature is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page 3.*

### About Notification Types

At the top of the page, there is a drop-down list of available notifications:




You can find detailed description of events below 290 in this topic.

## To modify a notification template

1. Select the desired notification from the drop-down list.
2. Specify the e-mail subject in the **Title Template** box.
3. Enter the desired e-mail text into the **Message Template** box.
4. Click **Save** to apply the changes.

**Notes:**

1. The text in the "Title Template" and "Message Template" boxes can include **Collaborator variables** (like `${review.id}` or `${review.creator.name}`). These variables let you insert user- and review-specific information into notifications. For complete list of available values, see Variables 161.

2. Any links to Collaborator server from the notifications, use server's URL as specified by the External URL 180 setting.

### To enable or disable a notification

1. Select the desired notification from the drop-down list.
2. Select the **Suppress Notification** check box to disable the notification. Clear this check box to enable the notification.

### To return to default settings

- Simply click **Restore Default Templates** on the page.

## How to Determine Notification Recipients Quickly

To indicate notification recipients, Collaborator puts the following keywords at the beginning of notification name:

| Recipient | Description |
|---|---|
| **Author** | The author of the review, for which the event is generated. The author is also called "review creator". |
| **User** | All participants of the review (except for the author). |
| **List** | All participants of the review (except for the author) **and also** the e-mail address (or mailing list) specified by the Notification List 199 setting of Collaborator. Using these notifications are convenient for notifying multiple users, including those users who are not mentioned in the Participants list of a review. |
| **Reviewer** | All users with a Reviewer role on the review. |
| **Poke** | The user, who is poked (that is, notified explicitly). |

## Event Descriptions

The drop-down list at the top of the setting page 288 displays notifications in alphabetical order. Below is a list of notifications organized by their type.

## Review Phase Notifications

Below is a description of notifications that Collaborator sends when a review phase changes:



- **Annotating**

  The following notifications indicate that the review is in the Annotating phase:

| Notification | Sent To | Remarks |
|---|---|---|
| **List Annotating** | All review participants and recipients of the Notification List e-mails | Just a notification that the review is in the Annotating phase. |
| **User Annotating** | All review participants | A notification that the review is in the Annotating phase. The default message template also describes the recipient's role in the review. |

- **Canceled**

  The following notifications indicate that the review has been canceled:

| Notification | Sent To |
|---|---|
| **List Cancel** | All review participants and recipients of the Notification List e-mails |
| **User Cancel** | All review participants |

The default notification message mentions the user, who canceled the review.

- **Completed**

  The following notifications indicate that all the reviewers have marked the review as completed, and that the author needs to mark it as completed as well in order for the review to be finished:

| Notification | Sent To |
|---|---|
| **Author Finish Review Required** | Review author |
| **List Author Finish Review Required** | All review participants and recipients of the Notification List e-mails |

**Note:** Configurations on the Admin | Roles 279 page have the Marks reviews "finished" 283 setting. It specifies, to which roles the above-mentioned notifications are applicable. If this setting does not specify that some role can mark reviews as "finished", Collaborator does not send notifications to the participants that have this role.

The following notifications indicate that the review has been completed:

| Notification | Sent To | Remarks |
|---|---|---|
| **Author Complete Checkin** | Review author | Indicates that the review has been completed, and the author can commit the changes to the source control. |
| **Author Complete Checkin Unread** | Review author | Indicates that the review has been completed, but has unread comments. The author can commit the changes to the source control, but should read the comments before doing this. |
| **Author Complete Nocheckin** | Review author | Indicates that the review has been completed. |
| **Author Complete Nocheckin Unread** | Review author | Indicates that the review has been completed and that there are unread comments in it. |
| **List Complete Checkin** | All review participants and recipients of the Notification List e-mails | Indicates that the review has been completed, and the author can commit the changes to the source control. |

| Notification | Sent To | Remarks |
|---|---|---|
| **List Complete Checkin Unread** | All review participants and recipients of the Notification List e-mails | Indicates that the review has been completed, but has unread comments. The author can commit the changes to the source control, but should read the comments before doing this. |
| **List Complete Nocheckin** | All review participants and recipients of the Notification List e-mails | Indicates that the review has been completed. |
| **List Complete Nocheckin Unread** | All review participants and recipients of the Notification List e-mails | Indicates that the review has been completed and that there are unread comments in it. |
| **User Complete** | All review participants | A notification that the review has been completed. |
| **User Complete Unread** | All review participants | A notification that the review has been completed and that there are some unread comments in it. |

**Note:** the list above contains the Checkin and Nocheckin notifications that are very similar to each other. The notification message, which Collaborator sends, depends on the "*Create a Commit Action Item for Completed Review*" setting (you can find it on the Admin | General 180 setting page). If the setting's value is `Create`, then Collaborator sends the "Checkin" notifications. Otherwise, it sends the "Nocheckin" notifications.

- **Inspection**

  The following notifications indicate that the review is in the Inspection phase.

| Notification | Sent To | Remarks |
|---|---|---|
| **Author Inspection** | Review author | Indicates that the review has got to the Inspection phase and that the author can respond to comments. |

| Notification | Sent To | Remarks |
|---|---|---|
| **List Inspection Resume** | All review participants and recipients of the Notification List e-mails | Indicates that the review returned back to the Inspection phase and that participants can review the author's proposed fixes. |
| **List Inspection Resume Chat** | All review participants and recipients of the Notification List e-mails | Indicates that the review returned back to the Inspection phase and that it has comments that participants need to read. |
| **List Inspection Start** | All review participants and recipients of the Notification List e-mails | Indicates that the review has got to the Inspection phase for the first time. |
| **User Inspection Resume** | All review participants | Indicates that the review returned back to the Inspection phase and that participants can review the author's proposed fixes. |
| **User Inspection Resume Chat** | All review participants | Indicates that the review returned back to the Inspection phase and that it has comments that participants need to read. |
| **User Inspection Start** | All review participants | Indicates that the review has got to the Inspection phase for the first time. |

- **Planning**

  The following notifications indicate that the review returned back to the Planning phase. The author needs to finish planning in order for inspection to be able to continue.

| Notification | Sent To |
|---|---|
| **Author Continue Planning** | Review author |
| **List Continue Planning** | All review participants and recipients of the Notification List e-mails |

- **Rejected**

The following notifications indicate that the review has been rejected. The default notification message says what user rejected the review.

| Notification | Sent To |
|---|---|
| **List Reject** | All review participants and recipients of the Notification List e-mails |
| **User Reject** | All review participants |

- **Rework**

  The following notifications indicate that the review is in the Rework phase. The author needs to rework found defects:

| Notification | Sent To |
|---|---|
| **Author Rework** | Review author |
| **List Rework** | All review participants and recipients of the Notification List e-mails |
| **User Rework** | All review participants |

- **Signing**

  Collaborator sends these notifications to indicate that reviewers need to sign the review[193].

| Notification | Sent To | Remarks |
|---|---|---|
| **Author Review Signed** | Review author | Indicates that the author has signed the review. |
| **Author Review Signed Checkin** | Review author | Indicates that the author has signed the review. |
| **Note:** The two notifications above are similar to each other. The notification message, which Collaborator sends, depends on the "*Create a Commit Action Item for Completed Review*" setting (you can find it on the Admin \| General[180] setting page). If the setting's value is `Create`, then Collaborator sends the "Checkin" notification. Otherwise, it sends the other notification. | | |
| **List Complete Waiting for Signature** | All review participants and recipients of the Notification List e-mails | Indicates that the review needs to be signed for approval. |

| Notification | Sent To | Remarks |
|---|---|---|
| **List Signature Declined** | All review participants and recipients of the Notification List e-mails | Indicates that some participant has declined to sign the review. |
| **User Complete Signature** | All review participants | The review has been completed and needs to be signed by participants. |

- **Stalled**

  The following notifications indicate that the review is stalled and needs some activity:

| Notification | Sent To | Remarks |
|---|---|---|
| **List Review Stalled Author Not Reworking** | All review participants and recipients of the Notification List e-mails | Indicates that the review is stalled because the author needs to fix some defects. |
| **List Review Stalled Reviewer Not Finished** | All review participants and recipients of the Notification List e-mails | Indicates that the review is stalled because a reviewer needs to approve it. |
| **Review Stalled Author Finish Annotating** | Review author | Indicates that the review is stalled in the Annotating phase. It is waiting for the author to begin the Inspection phase. |
| **Review Stalled Author Not Reworking** | Review author | Indicates that the review is stalled in the Reworking phase, and that the author should fix defects. |
| **Review Stalled Reviewer Not Finished** | Participants with the Reviewer role | Indicates that the review is stalled in the Reworking phase. The default message specifies the reviewer, who needs to approve the review in order for it to be completed. |

| Notification | Sent To | Remarks |
|---|---|---|
| **User Waiting Failed** | All review participants | Indicates that the review is stalled because some review participants have already approved the review and others are waiting for each other. |

- **Approaching Deadline**

  The following notifications indicate that the review is approaching its deadline and needs some activity:

| Notification | Sent To | Remarks |
|---|---|---|
| **List Review Approaching Deadline Author Not Reworking** | All review participants and recipients of the Notification List e-mails | Indicates that the review is approaching its deadline and the author needs to fix some defects. |
| **List Review Approaching Deadline Reviewer Not Finished** | All review participants and recipients of the Notification List e-mails | Indicates that the review is approaching its deadline because a reviewer needs to approve it. |
| **Review Deadline Change** | All review participants and recipients of the Notification List e-mails | Informs that the review's deadline has been changed. |
| **Review Approaching Deadline Author Finish Annotating** | Review author | Indicates that the review is approaching its deadline but is still in the Annotating phase. The author needs to move it to the Inspection phase. |
| **Review Approaching Deadline Author Not Reworking** | Review author | Indicates that the review is approaching deadline in the Reworking phase, and that the author should fix defects. |

| Notification | Sent To | Remarks |
|---|---|---|
| **Review Approaching Deadline Reviewer Not Finished** | Participants with the Reviewer role | Indicates that the review is approaching deadline in the Reworking phase. The default message specifies the reviewer, who needs to approve the review in order for it to be completed. |

## Changes in File and Participant Lists

Collaborator sends the following notifications when the review's properties change:

| Notification | Sent To | Remarks |
|---|---|---|
| **List New Files** | All review participants and recipients of the Notification List e-mails | Indicates that new files have been added to the review. |
| **List In Progress User Added** | All review participants and recipients of the Notification List e-mails | Indicates that one or more users have been added to the review. The default notification message specified the users' roles in the review. |
| **List In Progress User Changed** | All review participants and recipients of the Notification List e-mails | Indicates that the role of some review participant has changed. |
| **List in Progress User Removed** | All review participants and recipients of the Notification List e-mails | Indicates that one or more users have been deleted from the review's Participants list. |
| **User New Files** | All review participants | Indicates that new files have been added to the review. |

## Changes in Comments

When a review is in the Inspection phase and there are unread comments, Collaborator sends the *List Inspection Resume Chat* and *User Inspection Resume Chat* notifications. See their descriptions above 293.

## Miscellaneous

| Notification | Sent To | Remarks |
|---|---|---|
| **Poke** | A review participant that you poked in the Collaborator UI | Just a notification that the review needs the user's attention. |
| **List Calendar Invite** | All review participants and recipients of the Notification List e-mails | A notification in iCalendar format to schedule a formal meeting on review. |

## See Also

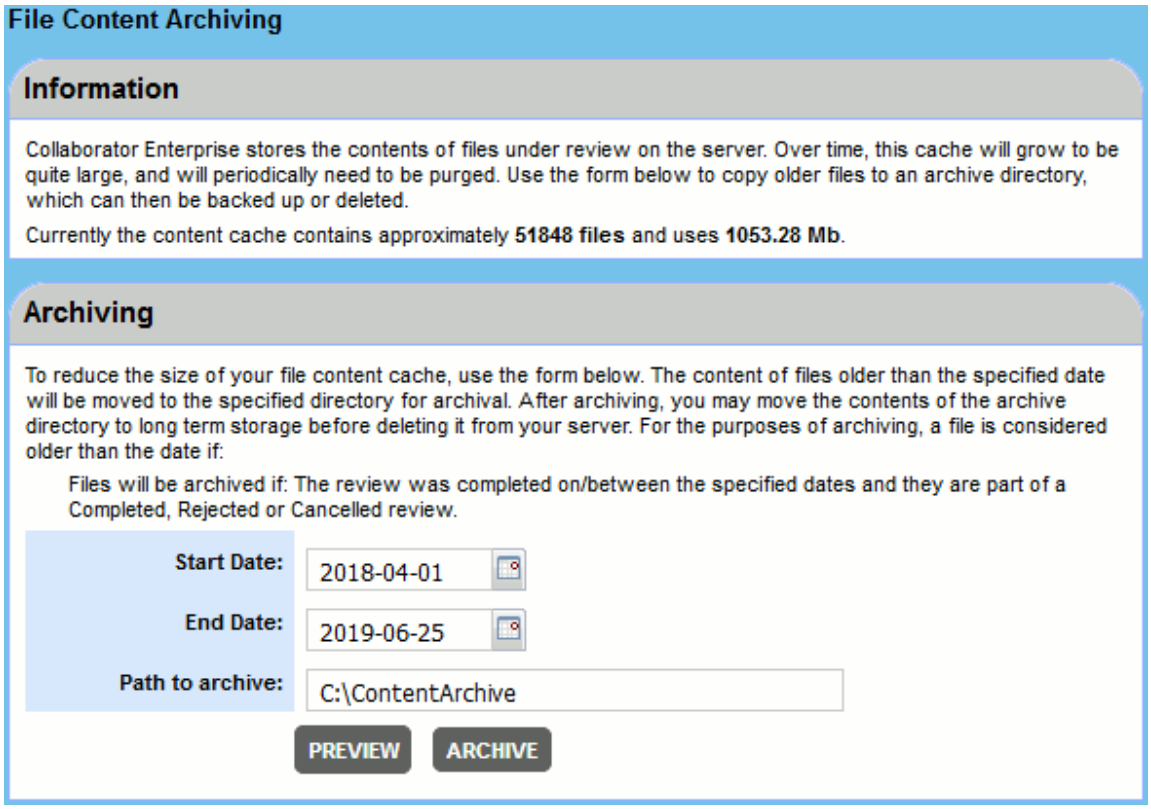
Notifications on Review Changes 329

Variables 161

### 3.3.17 Archiving Content Cache Data

Collaborator stores the contents of files under review on the server. Over time, this cache will grow to be quite large, and will periodically need to be purged. When deleting a review 346, Collaborator deletes files that were uploaded for this review from the content storage as well (unless they are used in some other reviews).

The **Archive** section of the Admin interface allows you to check the current status of the content cache and to archive (or delete) files that have not been in use for a long time.




The **Information** panel provides estimates of the total number of files and their collective size. Keep in mind that these are approximate estimates, since a complete scan of content cache folder could take plenty time. To determine a precise size of content cache and the total number of files in it, use standard file system tools.

The **Archiving** panel allows to archive old files. A file is considered "old" if:

- A file belongs to a review that was completed within the specified date period and the review phase is completed, rejected or cancelled.

-- OR --

- The file has never been part of a review (or was a part of a deleted review) and was uploaded within the specified date period.

Files that belong to in-progress reviews (in planning, annotating, inspection or rework phases) are NOT archived.

When a file belongs both to in-progress and to closed reviews, it will be copied to created archive, but remain in content cache.

## Archiving Files

> **Tip:** Archiving could be a long lasting operation (depending on amount of files), therefore we recommend that you use this feature at night or during another time when Collaborator database is idle.

To archive old files:

1. Specify the desired date period using the **Start Date** and **End Date** fields.
2. Specify where to store the archive in the **Path to archive** field.
3. Press **Preview** to determine how many files and how much data size will be moved off by the archive.
4. Press **Archive** to create the content cache archive.

Once the files are archived, they can be preserved in long-term storage or deleted entirely.

On attempts to open any of the archived file, users would see a "Content for is not available - it was probably archived by your administrator." message.

## Restoring Archived Files

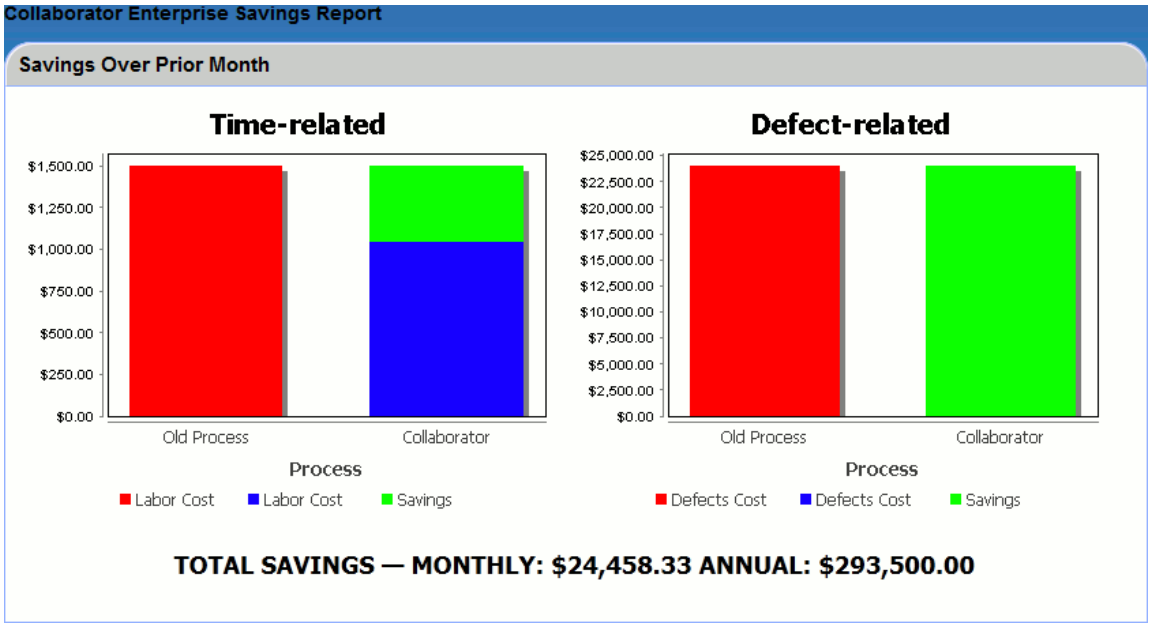
To restore files after they have been archived, simply copy them back into the live directory.

The directory structure must be preserved.

The copy operation can be done while the server is running; users will not be disrupted.

# 3.3.18 Savings Report

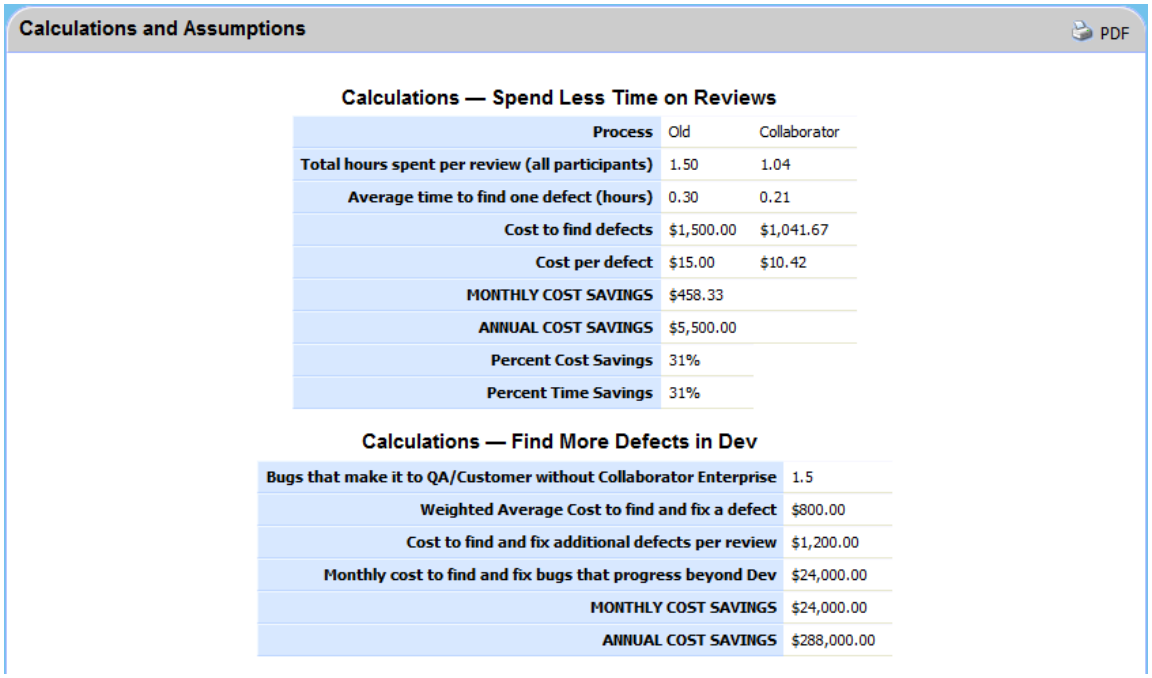
This page illustrates the benefits of a code review with Collaborator. It shows how much time and money Collaborator is likely to save your team.



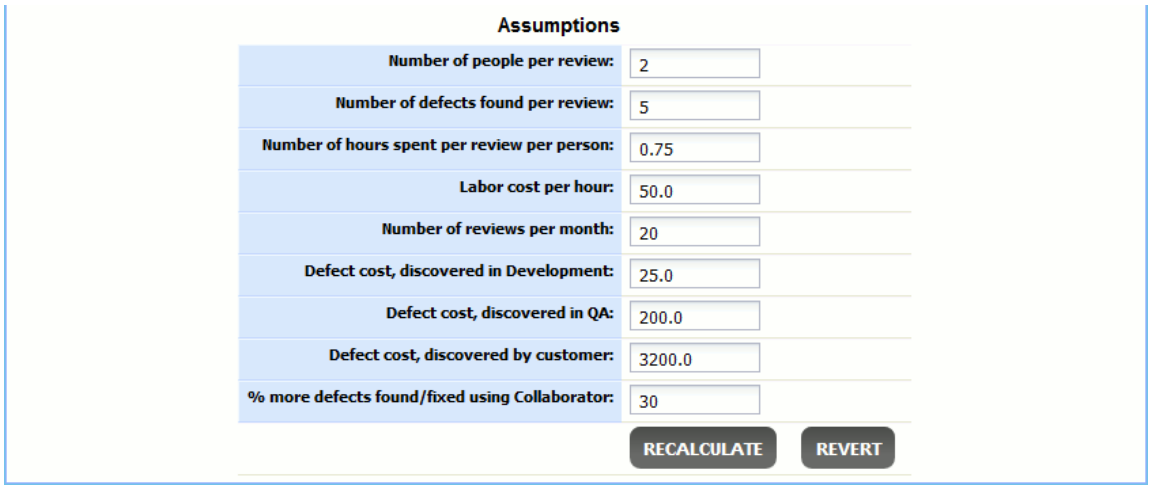

We find the following two metrics clear and easy to track.

- **Labor Cost** - Developer time saved by using Collaborator to do reviews.
- **Defects Cost** - The cost to find and fix bugs discovered in QA or after product release with the cost to find and fix bugs found by Developers during code review.

To export the Savings Report in a printable format, press PDF button in the upper-right corner.



These data is calculated automatically. The default numbers come from industry-standard sources, including some of our own customer data.




To get a personalized calculation, please specify the following initial data that better reflect your code review process:

**Number of people per review**

Number of people that usually take part in one review. You can get Minimal 285 and Minimal required 285 number of participants from Role configuration 279 settings.

**Number of defects found per review**

Average number of defects found in a single review.

**Number of hours spent per review per person**

How much time one person spends on a single review.

Studies show that developers find a bug every 10-15 min with Collaborator. That is, 4.8 bugs in an hour.

**Labor cost per hour**

Average per hour salary of your developers.

**Number of reviews per month**

Average number of reviews in a month.

Benchmark: Smart Bear's team of 5 developers does 75 reviews/month. For a team of 25, that review frequency would be 375 reviews per month.

**Defect cost, discovered in Development**

The cost of a defect found during development.

| | |
|---|---|
| | Code review typically costs 30% that of QA because bugs are found faster and locations/methods of fixes are already identified. |
| **Defect cost, discovered in QA** | The cost of a defect found during quality assurance. |
| **Defect cost, discovered by customer** | The cost of a defect found in production. |
| **% more defects found/fixed using Collaborator** | How many defects were found by code reviews with Collaborator.<br><br>We have chosen 30% as a default but feel free to change it. If your current code review process is not finding many bugs, then you should probably increase this value. |

When done, press **Recalculate**.

The **Fun Facts** section may help you in filling in the assumptions data as it displays the code review statistics within the previous 30 days.

## 3.3.19 Syntax Highlighting

When users open any text-based file in the Diff Viewer 377 and the "Syntax Coloring 372" option is enabled, Collaborator attempts to determine a computer language of this file and apply an appropriate syntax highlighting to it.

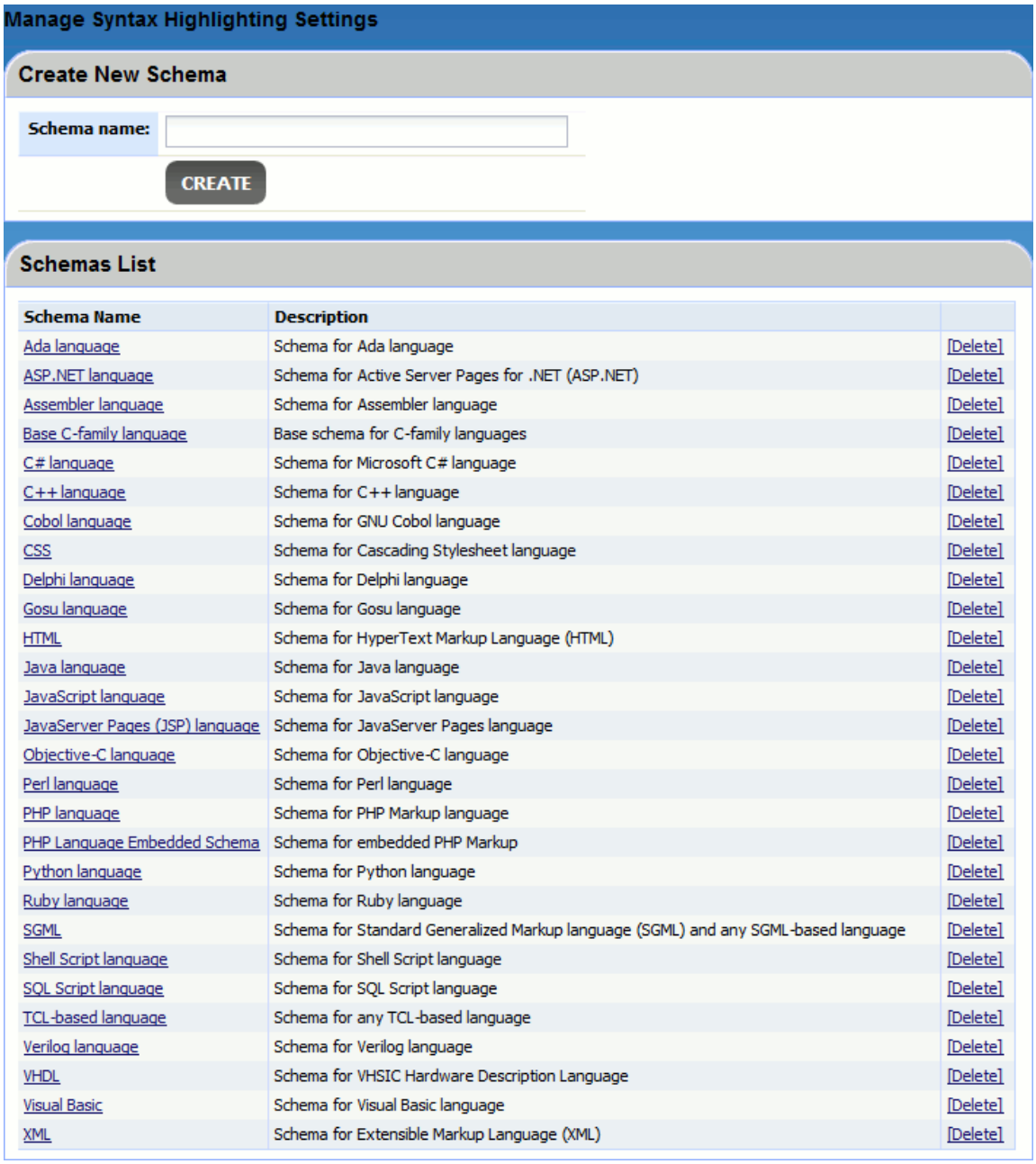
Collaborator has built-in support of syntax highlighting for most popular computer languages: C++, C#, Java, Ruby, Perl, ASP.Net, Python, SQL, HTML, XML and many others. Additionally, Collaborator administrators can create custom syntax highlighting schemas to add syntax highlighting for any other computer language.

The Syntax Highlighting page of the Admin screen allows you add, manage and delete syntax highlighting schemas for various computer languages.

***Syntax highlighting schema*** denotes what fragments of text should be highlighted as keywords, strings, constants, comments and so forth. It also specifies a list of file extensions to which the scheme will be applied. All syntax highlighting schemas are completely configurable, that is, you can specify new file extensions, add or delete keywords, modify patterns for strings, comments and so forth.

**Important.** In order to see highlighting schema updates within existing reviews, end-users may need to clear their browser's cache or upload new revisions of the review material.

The Schemas List provides a list of all available schemas, both predefined (default) and custom-created.




| Schema Name | Description | |
|---|---|---|
| Ada language | Schema for Ada language | [Delete] |
| ASP.NET language | Schema for Active Server Pages for .NET (ASP.NET) | [Delete] |
| Assembler language | Schema for Assembler language | [Delete] |
| Base C-family language | Base schema for C-family languages | [Delete] |
| C# language | Schema for Microsoft C# language | [Delete] |
| C++ language | Schema for C++ language | [Delete] |
| Cobol language | Schema for GNU Cobol language | [Delete] |
| CSS | Schema for Cascading Stylesheet language | [Delete] |
| Delphi language | Schema for Delphi language | [Delete] |
| Gosu language | Schema for Gosu language | [Delete] |
| HTML | Schema for HyperText Markup Language (HTML) | [Delete] |
| Java language | Schema for Java language | [Delete] |
| JavaScript language | Schema for JavaScript language | [Delete] |
| JavaServer Pages (JSP) language | Schema for JavaServer Pages language | [Delete] |
| Objective-C language | Schema for Objective-C language | [Delete] |
| Perl language | Schema for Perl language | [Delete] |
| PHP language | Schema for PHP Markup language | [Delete] |
| PHP Language Embedded Schema | Schema for embedded PHP Markup | [Delete] |
| Python language | Schema for Python language | [Delete] |
| Ruby language | Schema for Ruby language | [Delete] |
| SGML | Schema for Standard Generalized Markup language (SGML) and any SGML-based language | [Delete] |
| Shell Script language | Schema for Shell Script language | [Delete] |
| SQL Script language | Schema for SQL Script language | [Delete] |
| TCL-based language | Schema for any TCL-based language | [Delete] |
| Verilog language | Schema for Verilog language | [Delete] |
| VHDL | Schema for VHSIC Hardware Description Language | [Delete] |
| Visual Basic | Schema for Visual Basic language | [Delete] |
| XML | Schema for Extensible Markup Language (XML) | [Delete] |

## Creating a syntax highlighting schema

To create a syntax highlighting schema for a new computer language, scroll to the **Create New Schema** section, specify name for a new schema and press **Create**.

This will create a blank schema which you can configure as you like.

## Deleting a syntax highlighting schema

To delete a syntax highlighting schema, locate the desired schema in the list and press **Delete** link next to it.

## Resetting predefined syntax highlighting schemas

To reset predefined syntax highlighting schemas to their initial state, scroll to the **Restore default highlighting schemas** section and press **Restore Defaults**.

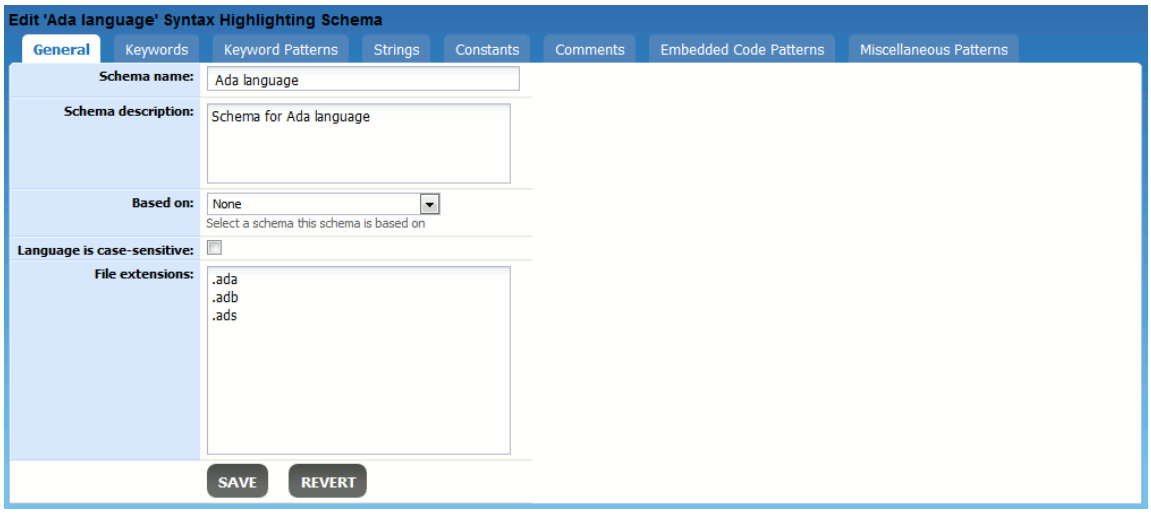
This action affects only predefined schemas, it does not alter custom schemas.

## Modifying a syntax highlighting schema

Click the name of the desired syntax highlighting schema in the Schemas List. This will display schema editor.

The editor has several tabs to configure different aspects of syntax highlighting. All the tabs will be described below.

On the **General** tab you can specify general set of scheme parameters.




| | |
|---|---|
| **Schema name** | Defines a name of the syntax highlighting schema. |
| **Schema description** | Defines a detailed description for the syntax highlighting schema. |
| **Based on** | One computer language could be based on another computer language. For example, C++ and C# languages are based on C language and HTML is based on SGML. |

Syntax highlighting schemas may inherit a number parameters from their parent schemas, so that you will not need to redefine the same parameters twice.

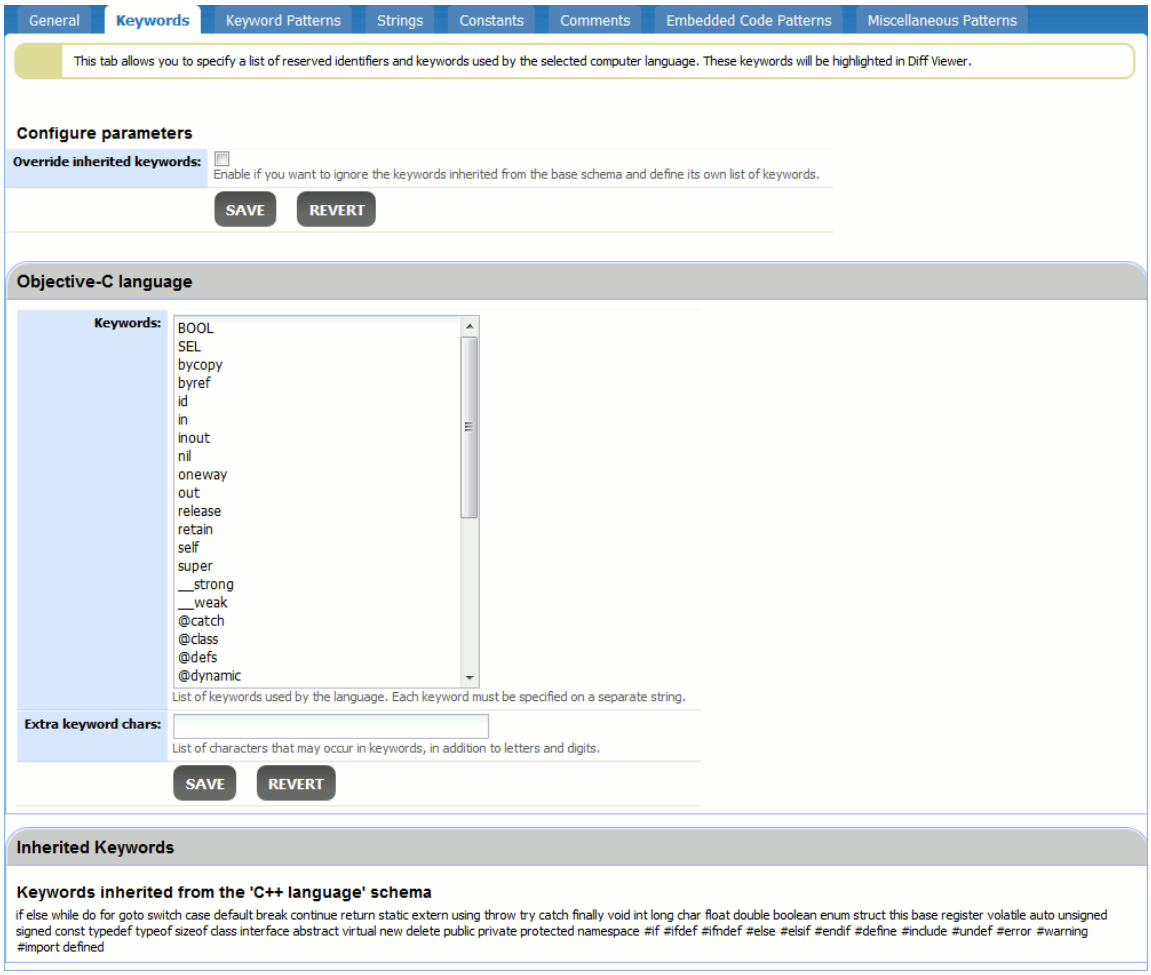
Select "None" if the desired computer language is not based on any other language. Otherwise select a schema of the parent language in the drop-down list.

**Language is case-sensitive** — Specifies whether the computer language is case-sensitive.

**File extensions** — Defines a list of extensions which may have source code files of the selected computer language. Specify each extension separated by semicolon, comma or space characters, or specify them on a separate line. Prefix each with a dot character.

Collaborator checks this list of file extensions to determine a syntax highlighting schema to apply for a file displayed in the Diff Viewer 377.

On the **Keywords** tab you can specify a list of reserved identifiers and keywords used by the selected computer language.




**Override inherited keywords**

This option is displayed only if current schema is based on some other schema.

Indicates whether you want to ignore the keywords inherited from parent schemas and define its own list of keywords. The list of keywords inherited from the parent schemas is displayed in the bottom of the tab.

**Keywords**

List of keywords used by the language. Specify each keyword separated by semicolon, comma or space characters, or specify them on a separate line.

**Extra keyword chars**

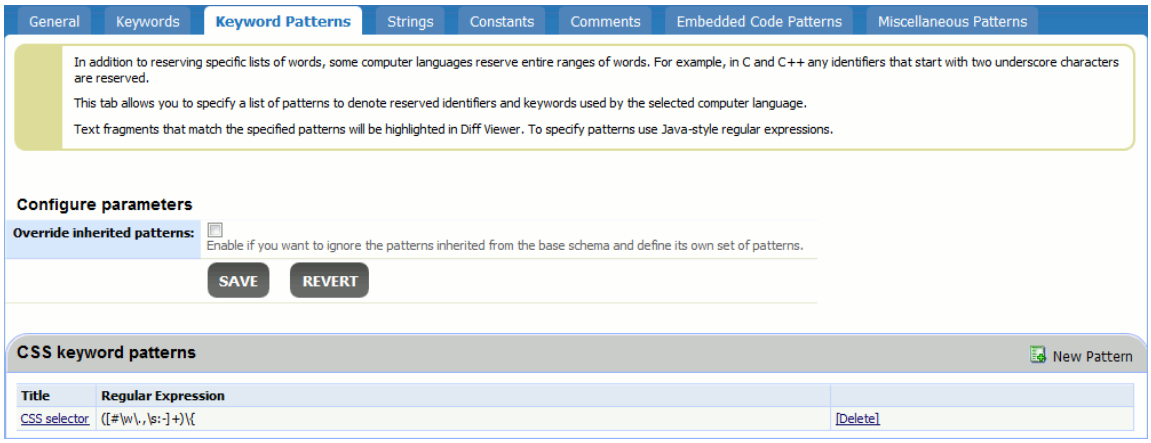
Some computer languages may have keywords that consist of 2 or more words/identifiers separated by some special characters. For example, Cobol language has keywords like ALPHANUMERIC-EDITED, SEGMENT-LIMIT and so forth.

Use this setting to define a list of characters that may occur in keywords of this language, in addition to letters and digits. Specify each character separated by semicolon, comma or space character.

In addition to reserving specific lists of words, some computer languages reserve entire ranges of words or notation formats. For example, HTML tags or CSS selectors have a specific notation format. Another example is C and C++ languages where any identifiers that start with two underscore characters are reserved.

To apply syntax highlighting to a specific fragments of source code, Collaborator uses regular expression patterns. Each syntax highlighting schema allows specifying multiple patterns to highlight keywords, strings, constants, comments, embedded code and other important fragments of source code.

On the **Keyword Patterns** tab you can specify a list of patterns to denote reserved identifiers and keywords used by the selected computer language.

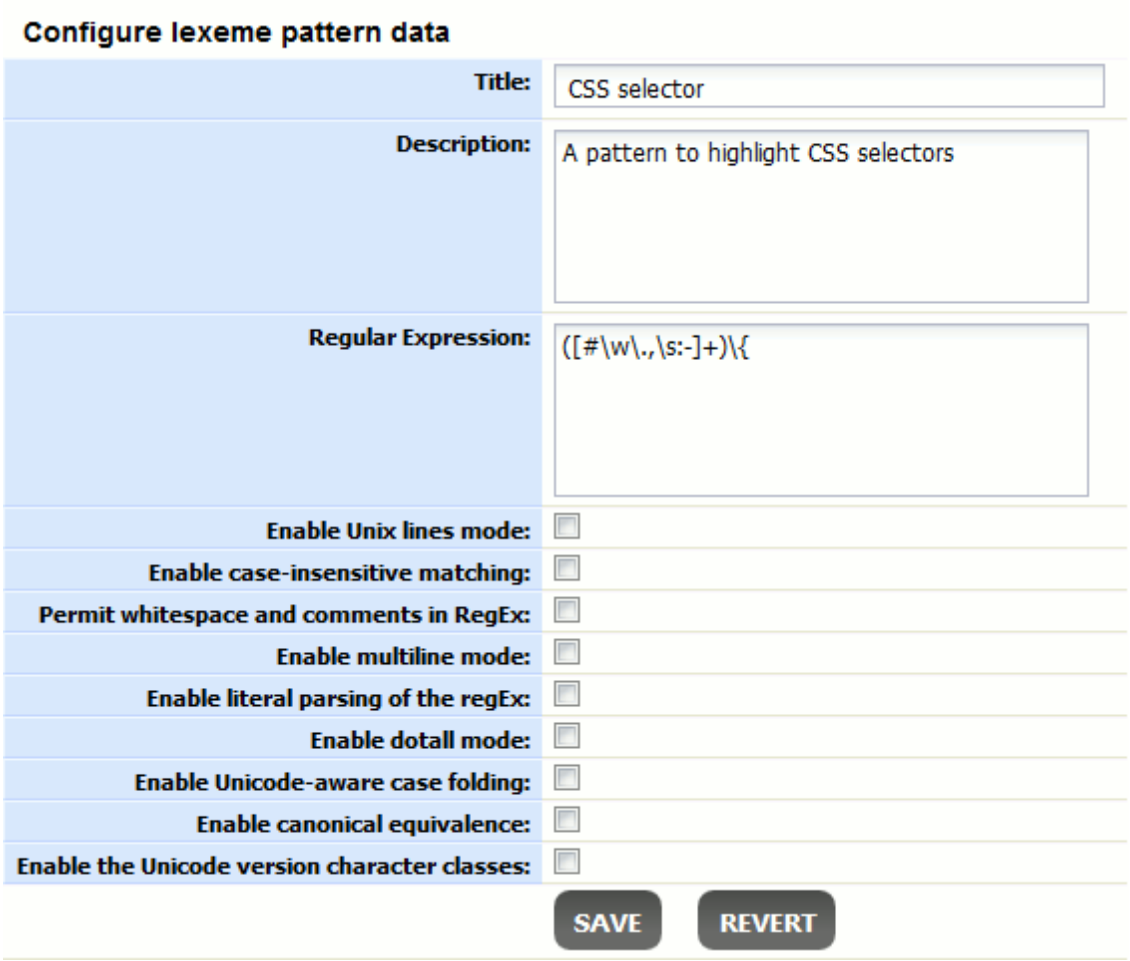



**Override inherited patterns**

This option is displayed only if current schema is based on some other schema.

Indicates whether you want to ignore the patterns inherited from parent schemas and define its own set of patterns. The list of patterns inherited from the parent schemas is displayed in the bottom of the tab.

To edit an existing pattern, click its name in the list. To create a new pattern, click **New Pattern**. Any of these actions will display a "Configure lexeme pattern data" form:




| | |
|---|---|
| **Title** | Defines a name of the lexeme pattern. |
| **Description** | Defines a detailed description for the lexeme pattern. |
| **Regular Expression** | A Java-style regular expression that identifies the desired fragment of text. |
| **Enable Unix lines mode** | When this flag is specified then only the '\n' line terminator will be recognized in the behavior of ., ^, and $. |
| **Enable case-insensitive matching** | When this flag is specified then two characters will match even if they are in a different case.<br><br>By default, case-insensitive matching assumes that only characters in the US-ASCII char-set are being matched. To enable Unicode-aware case-insensitive matching, enable both this and the "Enable Unicode-aware case folding" flags. |

| | |
|---|---|
| **Permit whitespace and comments in RegEx** | When this flag is specified then white-spaces will be ignored, and embedded comments starting with # will be ignored until the end of a line. |
| **Enable multiline mode** | In multiline mode the expressions ^ and $ match just after or just before, respectively, a line terminator or the end of the input sequence. By default these expressions only match at the beginning and the end of the entire input sequence. |
| **Enable literal parsing of the regEx** | Treat pattern string as a sequence of literal characters. In this case, metacharacters or escape sequences will not have any special meaning. |
| **Enable dotall mode** | In dotall mode, the expression . matches any character, including a line terminator. By default this expression does not match line terminators. |
| **Enable Unicode-aware case folding** | When this flag is specified then case-insensitive matching (if enabled by the "Enable case-insensitive matching" flag) will be performed in a manner consistent with the Unicode Standard. By default, case-insensitive matching assumes that only characters in the US-ASCII charset are being matched. |
| **Enable canonical equivalence** | When this flag is specified then two characters will be considered to match if, and only if, their full canonical decompositions match. The expression "a\u030A", for example, will match the string "\u00E5" when this flag is specified. By default, matching does not take canonical equivalence into account. |
| **Enable the Unicode version character classes** | Enables the Unicode version of Predefined character classes and POSIX character classes. |

Once specified the pattern data, click **Save**. Add as many patterns as you need.

Other tabs of syntax highlighting schema editor have the same functionality, but deal with different syntactical elements:

On the **Strings** tab you can specify a list of patterns to denote string literals within the selected computer language.

On the **Constants** tab you can specify a list of patterns to denote constants.

On the **Comments** tab you can specify a list of patterns to denote comments.

Some computer languages may include fragments of code on another computer language. Typical examples are client-side and server-side scripts. Based on HTML, they include fragments of embedded code on JavaScript, PHP, ASP.Net, Ruby and other languages.

Collaborator may apply syntax highlighting to embedded code as well. To do this, you need to define a list of patterns that will match the fragments of embedded code in the **Embedded Code Patterns** tab and select which syntax highlighting schema must be applied to these fragments in the **Embedded code schema** drop-down list.

On the **Miscellaneous Patterns** tab you can specify a list of any other syntactical elements to be highlighted.

# 4 Web Client

This chapter describes the web user interface of Collaborator. The Web Client allows you to perform all types of peer review tasks: create reviews as an author, participate in someone else's reviews as a reviewer or observer, inspect the review materials and so forth.

> In order to use the Collaborator's Web Client, your administrator must first install and configure 56 the server part of Collaborator.

The topics of this section contain detailed information on working with web client.

## In This Section

- Web Client - Overview 313
  Provides general information about Collaborator's Web Client.
- Account Management 314
  Contains information on your Collaborator's user account settings, logging in and logging out operations and subscribing to notifications.
- Performing Reviews 332
  Describes creating, performing, and managing peer reviews.
- Searching & Reporting 458
  Describes how to search for reviews and how to create reports.
- Troubleshooting 483
  Describes how to gather a support logs in case of troubles.

## Related Topics of Interest

- Collaborator Server 56
  Describes the server component of Collaborator.

- Desktop Clients 484
  Describes GUI Client, Command-Line Client, plugins for Eclipse, Microsoft Visual Studio and other clients for desktops.

# 4.1 Web Interface Client - Overview

The Collaborator's Web Client allows you to perform all types of peer review tasks: create reviews as an author, participate in someone else's reviews as a reviewer or observer, inspect the review materials and so forth.

## Requirements

In order to use the Collaborator's Web Client, your administrator must first install and configure 56 the server part of Collaborator.

The Web Client supports most actual versions of the following browsers:

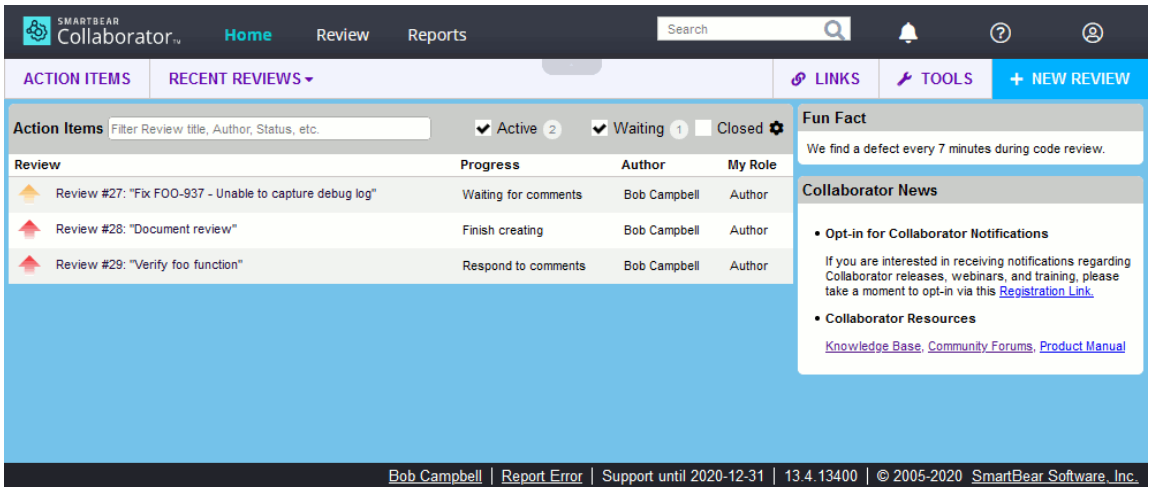
- Microsoft Edge
- Mozilla Firefox
- Apple Safari
- Google Chrome
- Opera Software Opera

Other browsers and versions *might* work to varying degrees.

## Opening Web Client

To open the Collaborator's Web Client just start a web browser and navigate to the URL of the Collaborator Server. If you do not know the URL, ask your Collaborator administrator.

If you have not been authenticated, you will be redirected the Log-in Screen 315. Otherwise, the Home Page will be displayed.




*Collaborator Web Client Home Page*

From the Home Page you can:

- Check your Action Items 333 to see if you have any assigned reviews.
- View detailed information about the review in the Review Summary Screen 345.
- Create new reviews 336.
- Search 458 for specific reviews or generate various reports 467.
- Manage your user preferences 317 and notifications 329.
- Read news, release announcements, webinar invitations and other information from the SmartBear.

## 4.2 Account Management

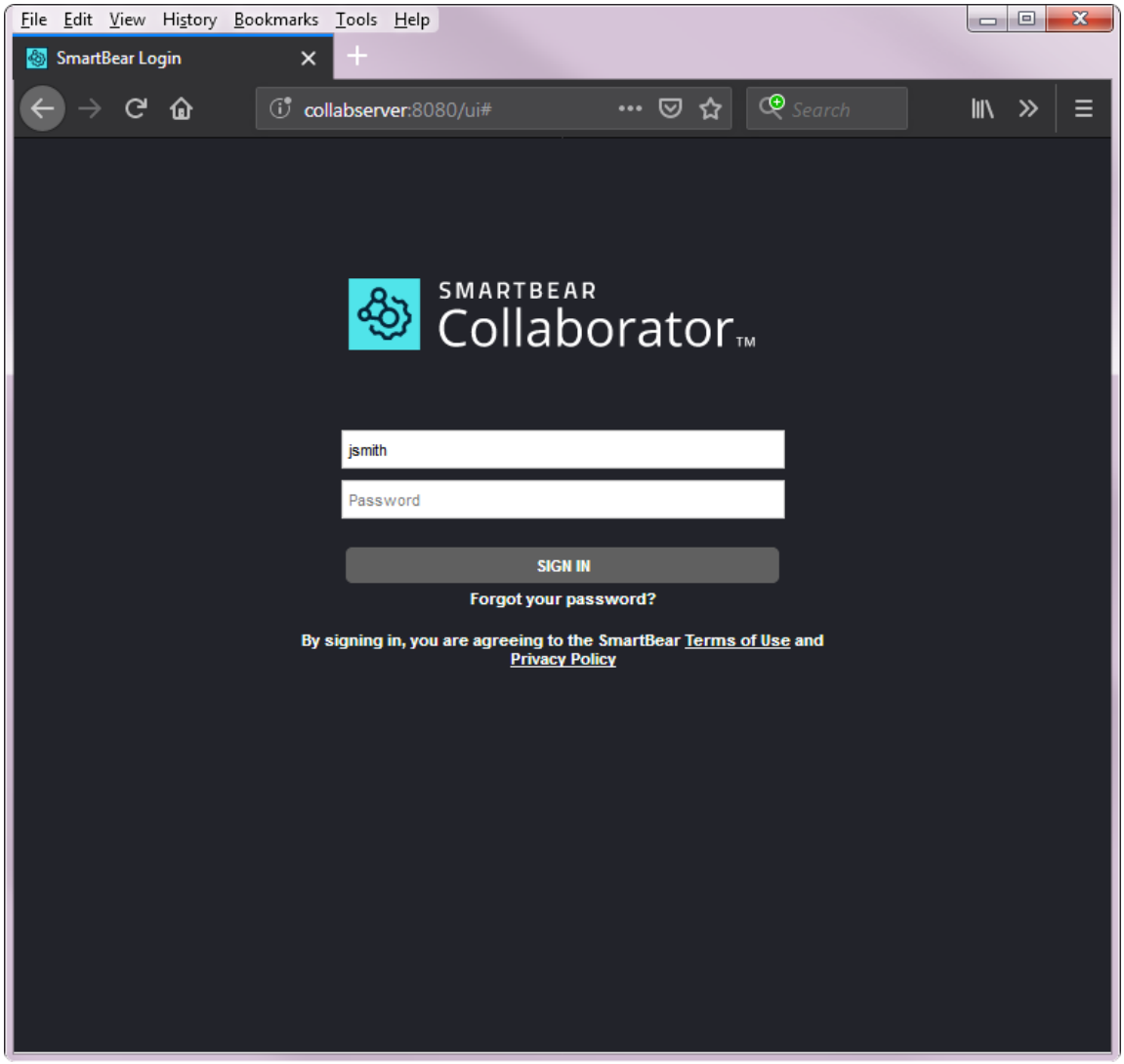
Topics of this section contains information on your Collaborator's user account settings, logging in and logging out operations and subscribing to notifications.

### In This Section

- Logging In and Logging Out 315
  Describes how to log in to and log out from Collaborator.
- User Preferences 317
  Provides information on user account settings.
- Notifications on Review Changes 329
  Contains information on notifications that Collaborator sends to inform review participants on review changes.

### 4.2.1 Log in / Log out

When you first visit the Collaborator web interface, most likely you will be redirected to the login screen. The URL to this screen will depend on your system administrator. It often includes a special port number.

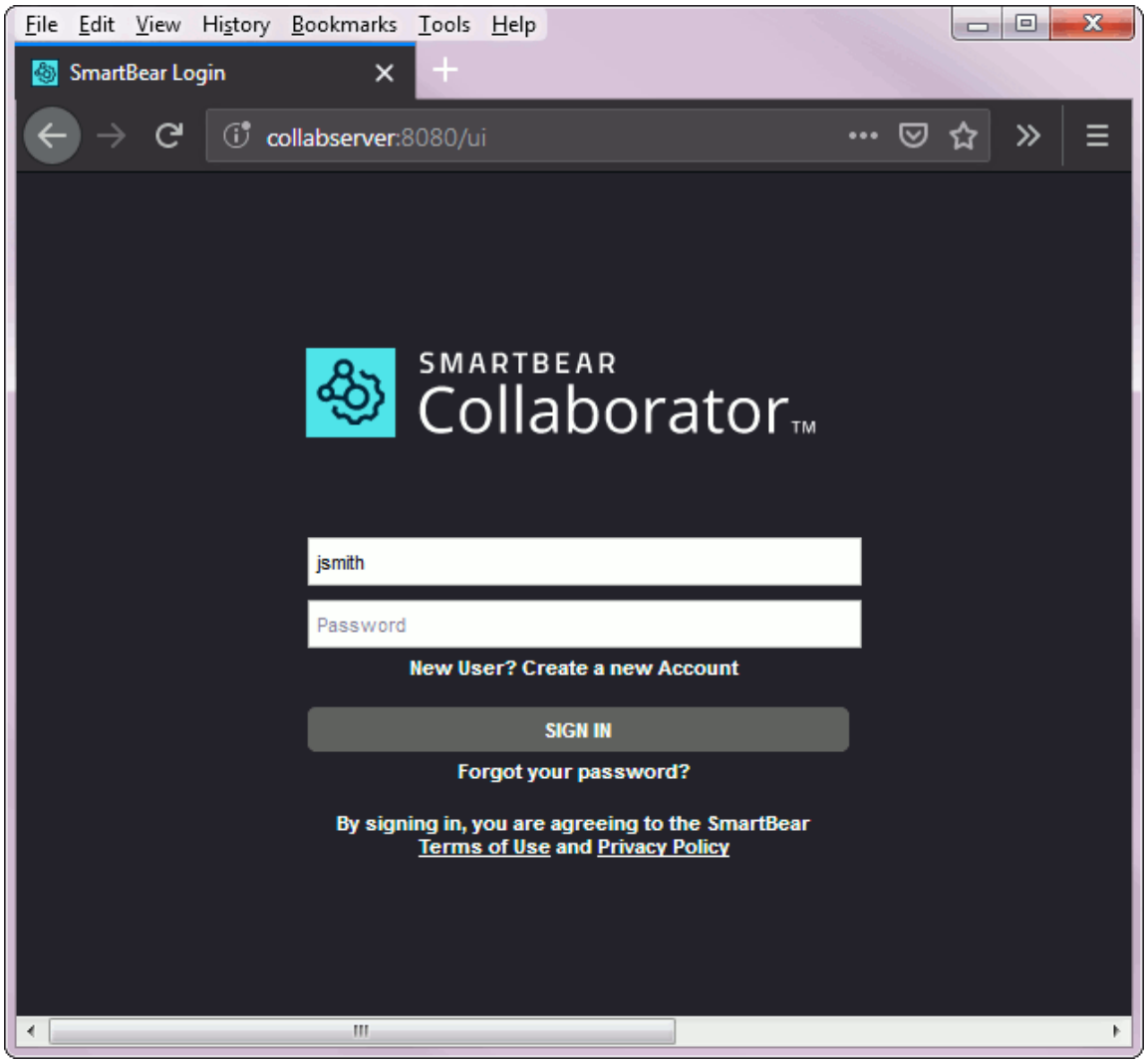



If your Collaborator server uses single sign-on authentication[130], then you will be redirected to single sign-on server's login screen instead.

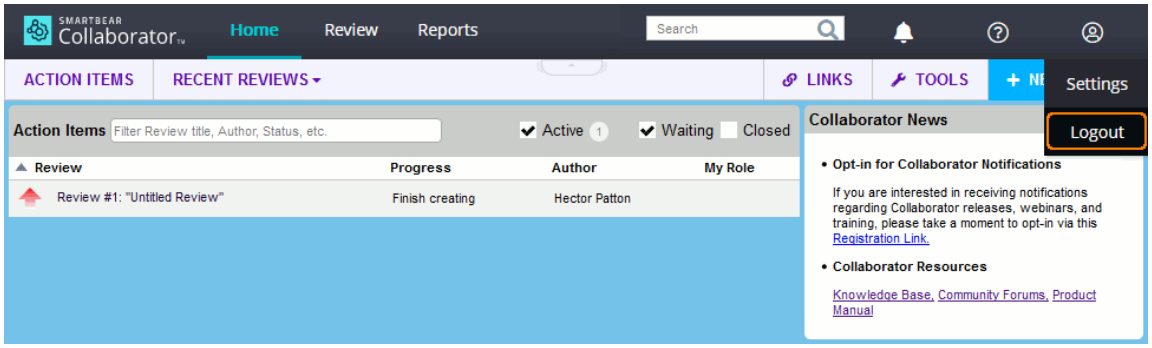
Typically, accounts in Collaborator will be identical to your version control system account.

If your account is not set up, you will need to contact an administrator to set one up[220]. If you use LDAP or ActiveDirectory[119] for authentication in your company, you can use that username/password with Collaborator and it will automatically create your user account.

Some system administrators will have enabled[180] a "Create an account" form on the front page:



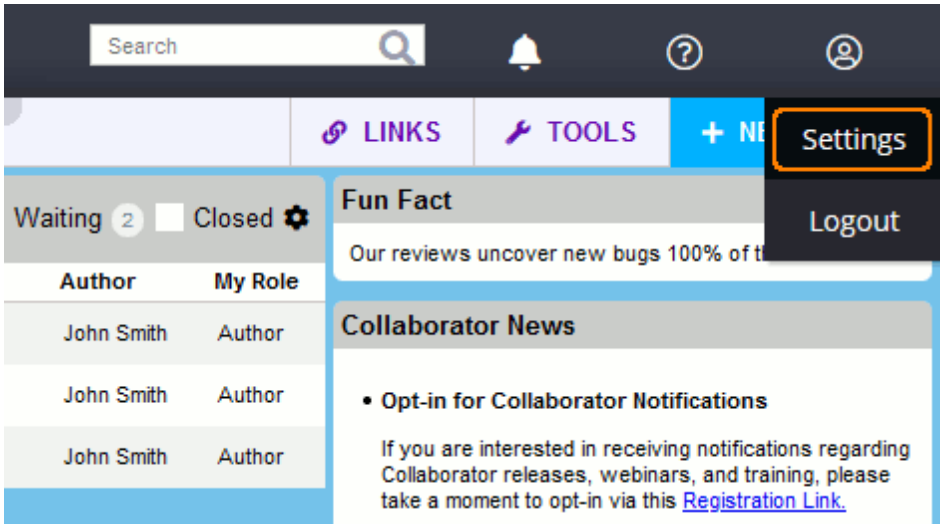

Once you are logged in, you are redirected to the Home Page. You can log out using the "Logout" button in the menu bar, but this is usually not necessary.



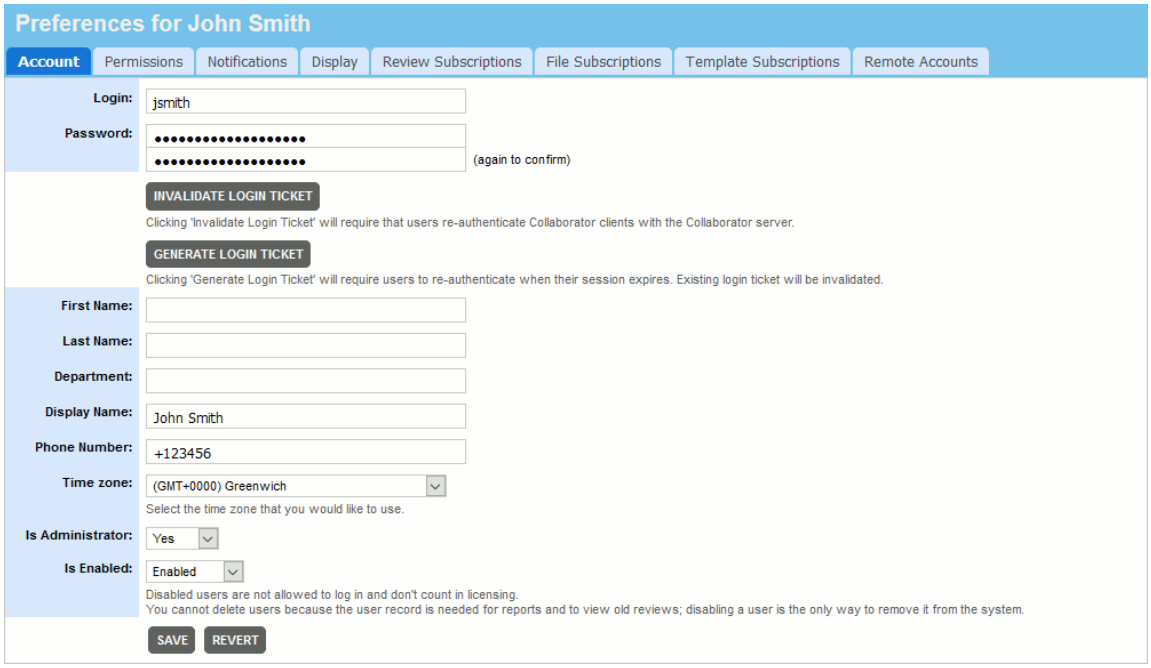

If you do not explicitly log out, the system will give your browser a cookie so the next time you go into a page you will not have to re-login first.

### 4.2.2 User Preferences

To edit your settings and preferences, use the **Settings** menubar command:




## Account Information




**Login** The token you use to log into 315 the system. Typically this will be identical to your version control login name.

Normally, you cannot edit your own login name. However, Collaborator administrators are able to edit anyone's login name, so contact an administrator if you need this done.

**Password** The password needed to log into the system. Can be blank for "no password," although in all cases the password will be displayed with many dots as shown in the screenshot above. This prevents the causal observer from knowing anything about your password, even its length.

When changing your password you have to type it in twice to confirm.

Instead of using password, you can use ***login ticket*** for authentication in Collaborator desktop clients. Login tickets are special alpha-numeric identifiers that act as user credentials for a limited time period. By default, tickets are valid for 30 days. Collaborator administrators can decrease or increase ticket's time-to-live 196.

To obtain a ticket, press **Generate Login Ticket** and copy the resulting alpha-numeric identifier. Now you can specify it instead of password in Collaborator desktop clients. Generating new login ticket, invalidates any existing login tickets.

To invalidate your existing login tickets immediately, press **Invalidate Login Ticket**. This will cause Collaborator desktop clients to re-authenticate.

**First Name** and **Last Name** Your first and last name, respectively.

If your server uses LDAP or AD authorization, these fields are read-only.

**Department** Name of your department, if any.

If your server uses LDAP or AD authorization, this field is read-only.

**Display Name** Your display name as seen by other users everywhere in the system. Up to 128 characters long. Typically, you can modify this setting yourself. However this can be disabled by Collaborator administrations, so that only administrators could modify all user display names 189.

If you leave this field blank your login name will be used for your name. This is undesirable however; most other users will not know you by your login name.

**Phone Number** The phone number other users can reach you at. This field is optional.

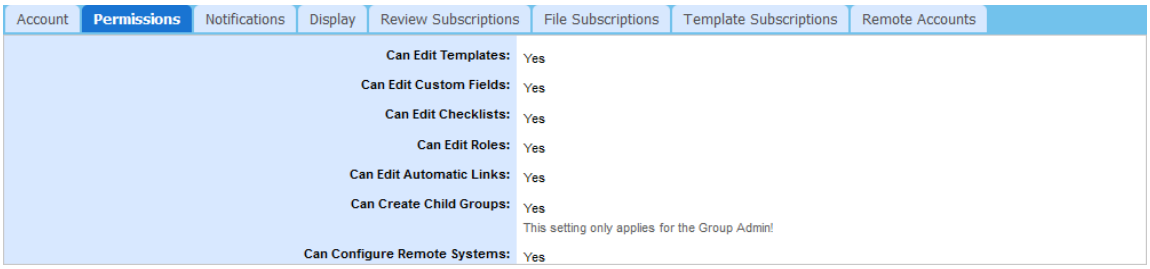
**Time Zone** Your time zone. This will be used to determine users's displayed date/time.

This is defaulted to the time zone selected on the Collaborator server.

**Is Administrator** Controls whether you are a system administrator[224]. Typically, you are not able to change this field, unless you are an administrator. Be careful: If you remove administrative access from yourself, there is no going back! (Unless another administrator adds you back.)

**Is Enabled** This field is used by system administrators to determine whether you are allowed[223] to log into the system. If you are looking at your preferences, you can log in, so you will always see this as "enabled".

## Permissions Information




**Can Edit Templates** Controls whether you have permissions[225] to manage templates[246]. Typically, you are not able to change this field, unless you are an administrator.

**Can Edit Custom Fields** Controls whether you have permissions[225] to manage custom fields[256]. Typically, you are not able to change this field, unless you are an administrator.

**Can Edit Checklists** Controls whether you have permissions[225] to manage checklists[272]. Typically, you are not able to change this field, unless you are an administrator.

**Can Edit Roles** Controls whether you have permissions[225] to manage roles[279]. Typically, you are not able to change this field, unless you are an administrator.

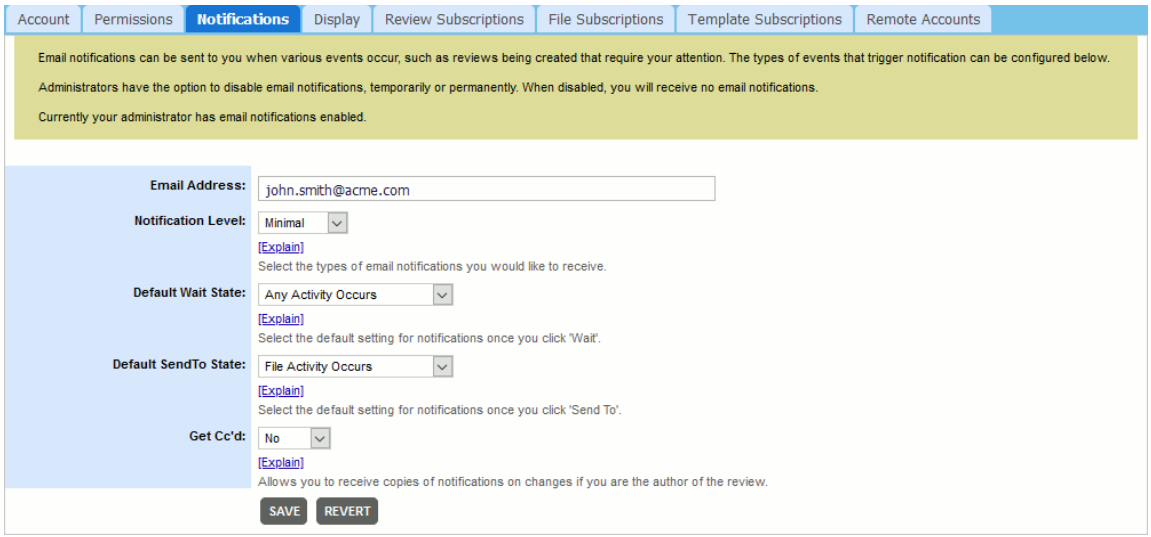
**Can Edit Automatic Links** Controls whether you have permissions[225] to manage automatic links[286]. Typically, you are not able to change this field, unless you are an administrator.

**Can Create Child Groups** Controls whether you have permissions[225] to create child groups. Actual for group administrators[231]. Typically, you are not able to change this field, unless you are an administrator.

**Can Configure Remote Systems**

Controls whether you have permissions 225 to manage remote system integrations 216. Effective when the Allow users to configure remote systems 216 global setting is enabled as well. Typically, you are not able to change this field, unless you are an administrator.

## Notification Preferences




**E-Mail Address**

The email address that you want all notifications delivered to, and also to display to other users in the system in case they want to contact you directly via email.

Although this field is technically not required, you really should fill it in, and if you do not, you will get a warning in your Action Items list 333.

**Notification Level**

This controls how many e-mails you want to get. See the [Explain] link for details on the options. Generally, these are the choices:

- Get all notifications, even if you were the cause of the notification event
- Get notification only of those events that you did not cause yourself
- Get no email notifications

**Default Wait State**

Specifies the default level of notifications when you perform the "Wait" action during reviews. Possible values are:

- Any Activity Occurs

- Activity by Author Occurs
- File Activity Occurs
- Someone Pokes Me

**Default Send To State**

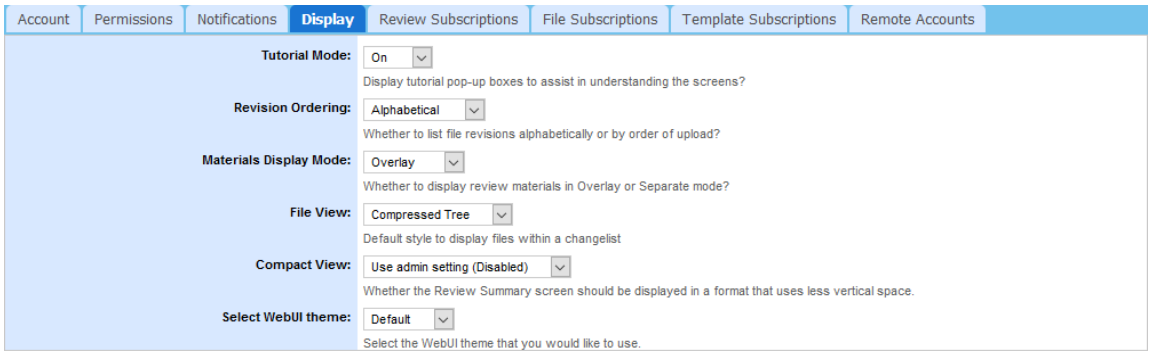
Specifies the default level of notifications when you perform the "Send To" action during reviews. Possible values are:

- Any Activity Occurs
- Activity by Author Occurs
- File Activity Occurs
- Someone Pokes Me

**Get Cc'ed**

This allows you to receive copies of notifications on changes if you are the author of the review.

## Display Preferences




**Tutorial Mode**

Controls whether those little yellow tutorial boxes are displayed throughout the system. By default, these boxes are displayed to help you understand the user interface.

**Revision Ordering**

Controls how file revisions are displayed, alphabetically or by order of upload.

**Materials Display Mode**

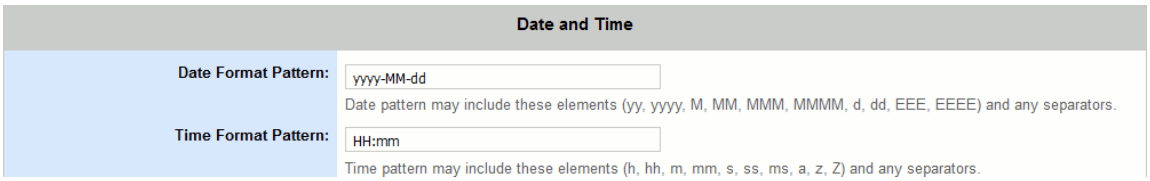
Allows the selection of the default display mode (Overlay or Separate) of materials on the Review Summary page.

**File View**

Controls which style should be used to display files within a changelist. Choices include Compressed Tree, Tree, or Flat.

**Compact View** When enabled, the Review Summary screen is presented in a format that uses less vertical space.

**Select WebUI theme** Specifies which of pre-defined Web client[312] themes you would like to use. This setting overrides the default WebUI theme[185] selected by administrator.




**Date Format Pattern** Specifies the preferred format of date values. The format pattern may include the following elements:

| Element | Description |
|---|---|
| yy | The last two digits of the year (that is, 2001 would be displayed as "01"). |
| yyyy | The full year (that is, 2001 would be displayed as "2001"). |
| M | The one- or two-digit month number. |
| MM | The two-digit month number. Single-digit values are preceded by a zero. |
| MMM | The three-character month abbreviation. |
| MMMM | The full month name. |
| d | The one- or two-digit day. |
| dd | The two-digit day. Single-digit day values are preceded by a zero. |
| EEE | The three-character weekday abbreviation. |

| | |
|---|---|
| EE<br>EE | The full weekday name. |

The format pattern includes the elements specified above separated by the desired symbols (:, /, spaces and so on). These symbols will be used as separators for the shown date/time values.

Below are some examples of date format patterns:

| Pattern | Displayed Value |
|---|---|
| yyyy-MM-dd | 2001-05-15 |
| d/M/yy | 15/5/01 |
| d MMMM yyyy, EEEE | 15 May 2001, Tuesday |

**Time Format Pattern**

Specifies the preferred format of time values. The format pattern may include the following elements:

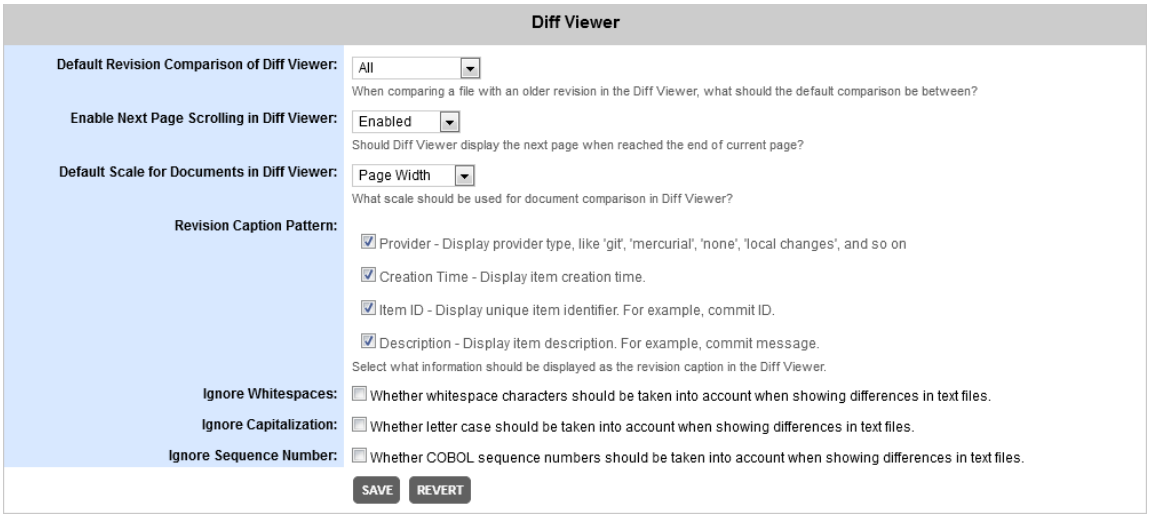
| Element | Description |
|---|---|
| h | The one- or two-digit hour. |
| hh | The two-digit hour. Single-digit hour values are preceded by a zero. |
| m | The one- or two-digit minute. |
| m<br>m | The two-digit minute. Single-digit minute values are preceded by a zero. |
| s | The one- or two digit second. |
| ss | The two digit second. Single-digit second values are preceded by a zero. |
| ms | The milliseconds. |
| a | The time would be displayed in the 12-hour format. |

| | |
|---|---|
| z | The alphanumeric time zone offset from UTC (that is, Pacific Standard Time would be displayed as "UTC-8"). |
| Z | The four-digit time zone offset from UTC (that is, Pacific Standard Time would be displayed as "-8000"). |

The format pattern includes the elements specified above separated by the desired symbols (:, /, spaces and so on). These symbols will be used as separators for the shown date/time values.

Below are some examples of time format patterns:

| Pattern | Displayed Value |
|---|---|
| hh:mm | 09:17 |
| h:m:ss.ms | 9:17:38.486 |
| h:m a z | 9:17 AM UTC-5 |




**Default Revision Comparison of Diff Viewer**

Allows the user to select which file revisions (if available) will be compared, by default, when the Diff Viewer is launched.

The available options are:

- All changes - Compare the current revision of a file against its *base revision* - that is, a state of file before any changes related to current review have been made. For pre-commit reviews, base revision is the revision that you checked out from the repository. For post-commit reviews, base revision is the revision that precedes your commit.

- First vs Last - Compare the most recent revision of a file against its first revision that was uploaded during this review.

- Branch only - Compare the current revision of a file against its base revision excluding changes merged from other branches.

- Last commit - Compare the current revision of a file and its previous revision.

- Accepted - Compare the current revision of a file against its latest accepted revision[433]. During the review, participants may accept some particular revisions of a file to denote that they agree with the changes. Uploading a further revision of that file clears the Accepted mark. If a participant has not accepted any revision yet, then compares the current revision of a file against its base revision.

- Commits - Compare the most recent revision against any arbitrary revision chosen in the **Select revision** drop-down.

> The "Branch only" mode may produce slightly different output depending on whether the review was created manually or via repository integration. In reviews created manually it is not always possible to filter-out merge changes, so sometimes they still could be displayed. Reviews created via integration have broader access to the repository and thus can exclude merge changes more thoroughly. The drawback of the latter approach is that the contents of individual merge commits will be displayed incorrectly.

**Enable Next Page Scrolling in Diff Viewer** — Specifies whether Diff Viewer should display the next page when reached the end of current page in document reviews.

**Default Scale for Documents in Diff Viewer**

Specifies the initial zoom level when reviewing word-processing documents, presentations, PDF documents, images and vector graphics. You can set the scale to various percentages, Page Width or Full Page.

When you change scale of current document in the Diff Viewer it will be cached so that all further pages of that document would be displayed in the same scale.

**Revision Caption Pattern**

Specifies what information should be displayed as the revision caption in the simple mode of revision selection of Diff Viewer. Any combination of the following is possible:

- Provider - Display provider type, like 'git', 'mercurial', 'none', 'local changes', and so on.
- Creation Time - Display item creation time.
- Item ID - Display unique item identifier. For example, commit ID.
- Description - Display item description. For example, commit message.

**Ignore Whitespaces**

Controls whether white-spaces are taken into account when showing differences.

This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports.

**Ignore Capitalization**

Controls whether capitalization is taken into account when showing differences.

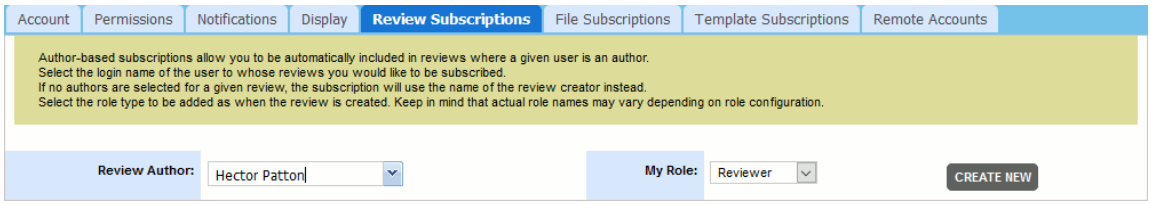
This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports.

**Ignore Sequence Number**

Controls whether COBOL sequence numbers are taken into account when showing differences.

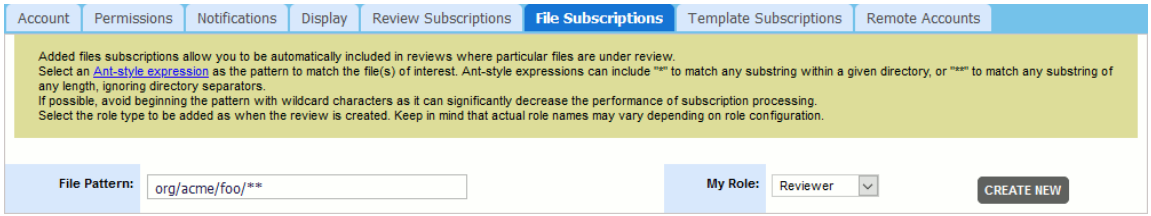
This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports.

## Review Subscriptions Preferences



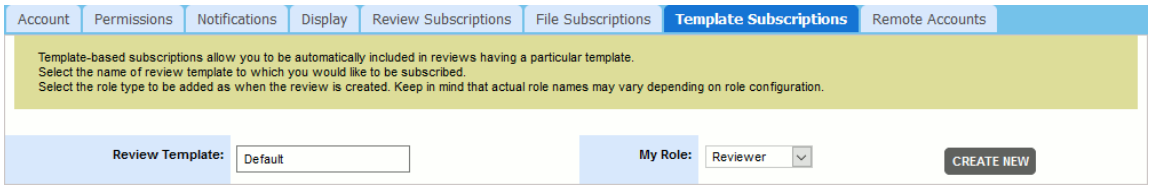

Review Subscriptions allow you to automatically subscribe and be included in reviews with a preferred author. Under the "review creator" field, select the login name of the user to whose reviews you would like to be subscribed. If no authors are selected for a given review, the subscription will use the name of the review creator instead - unless the review creator is flagged as a system admin. Choose your role type (**Reviewer**, **Observer**, or **Moderator)** in the review. Keep in mind that actual role names may vary depending on your role configuration[281].

## File Subscriptions Preferences



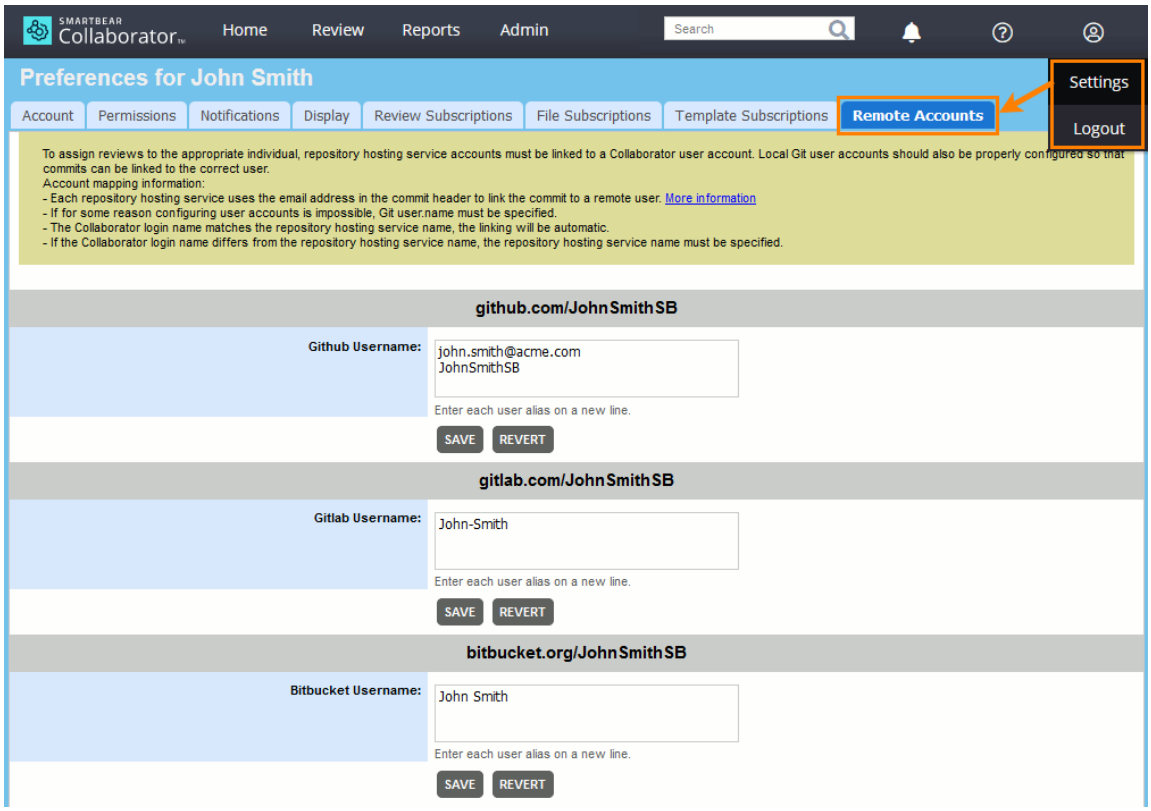

File Subscriptions allow you to automatically subscribe and be included in reviewers where a particular file is under a review. Choose your role type (**Reviewer**, **Observer**, or **Moderator)**. Select an Ant-style expression as the pattern to match the file(s) of interest. The expression will be matched to the file path and repository name; it will not look for matches in the complete repository path with URL and other server descriptions. Ant-style expressions can include "*" to match any substring within a given directory, or "**" to match any substring of any length, ignoring directory separators. If possible, avoid beginning the pattern with wildcard characters as it can significantly decrease the performance of subscription processing.

## Template Subscriptions Preferences




Template Subscriptions allow you to automatically subscribe and be included in reviews having a particular template. Select the name of a template you would like to subscribe to in the "Review Template" field and specify your desired role type (**Reviewer**, **Observer**, or **Moderator**).

## Remote Accounts Preferences




Remote Accounts allow you to specify correspondence between your user name or email address in Collaborator and your logins on various remote servers. These settings are used to match user names for Collaborator integrations with remote servers. See Configuring User Remote Accounts 885.

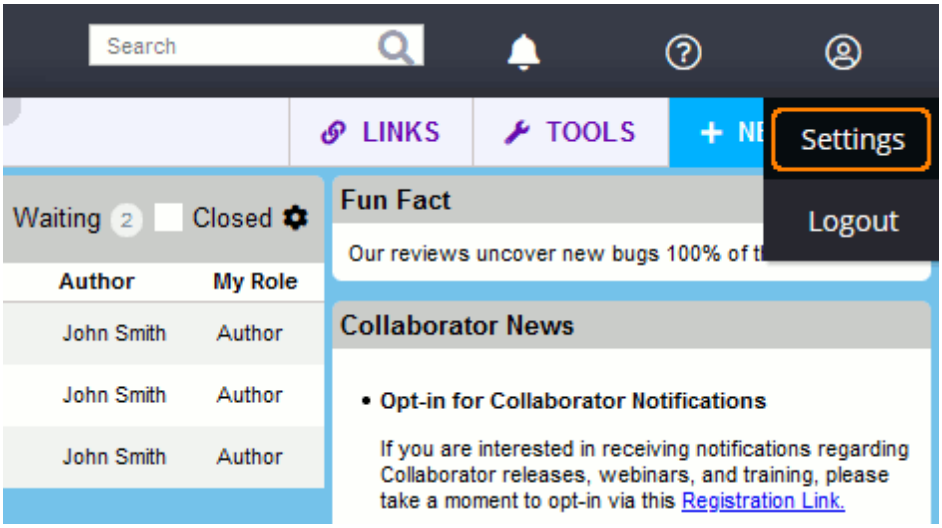
### 4.2.3 Notifications on Review Changes

You can subscribe to notifications about changes in reviews, in which you participate. Collaborator will notify you about changes in review statuses and activities of other participants. The notifications can come through email or RSS.

## Subscribing to Email Notifications

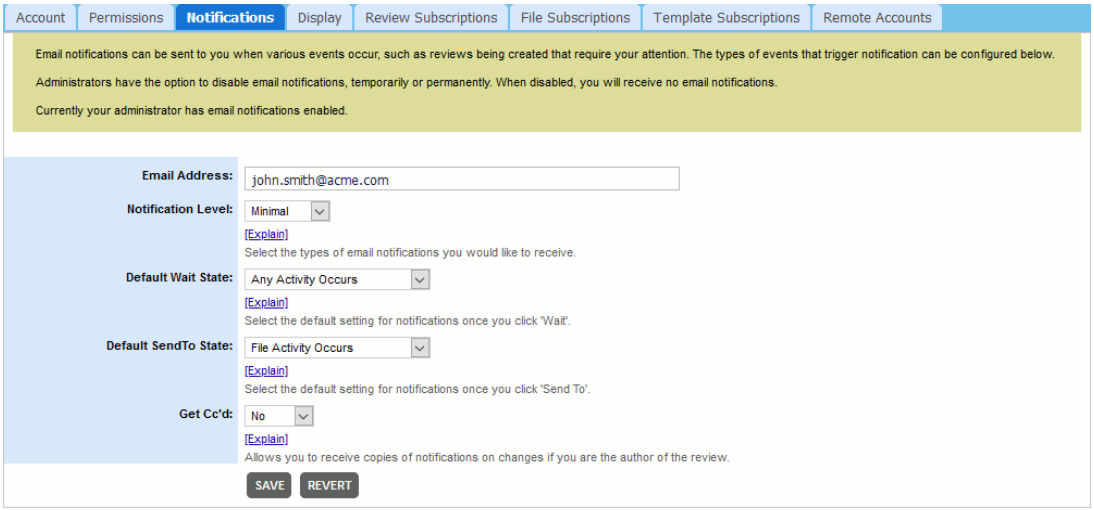
1. Log in to the Collaborator Web Client.

2. Click **Settings** on the home page.




   This will open the **Preferences** page for your account.

3. Switch to the **Notifications** tab:



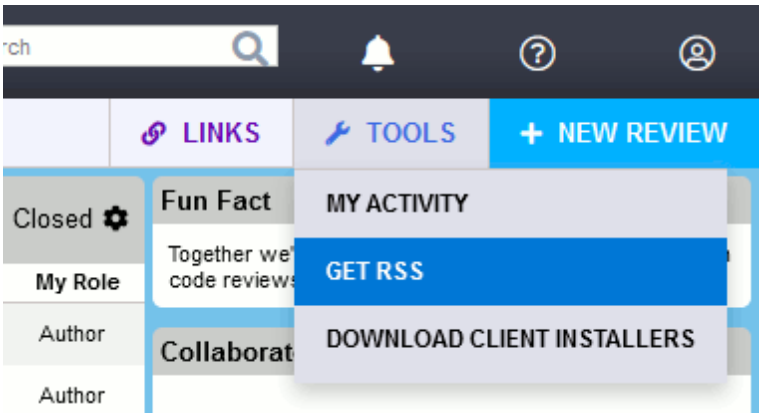

4. On the tab, check your email address and specify the desired notification level (see below 331).

5. Click **Save** to apply the changes.

## Subscribing to RSS Notifications

1. Log in to the Collaborator Web Client.

2. Click **Tools | Get RSS** in the home page menu:




## Who Can Receive Notifications

Collaborator can send notifications both to review author as well as to other participants. The user needs to be in an active "Wait[345]" state, which is an option at the very bottom of the Review Summary[345] screen under "Next Steps". If a user is waiting for an action, they will be informed via email once that action takes place.

If you do not participate in a review, but want to be notified about its status, you can do the following:

- Ask the review author or administrator to add you into the review (they can add you as an observer).
- Create an author-based subscription[327] which will automatically add you to the reviews created by the particular author.

By default, Collaborator notifies you about actions of other users on the review. To get all notifications, including ones for actions initiated by you, change your notification level setting to *All* (see below[331]).

## When Collaborator Sends Notifications

Collaborator sends notifications when an author, or a participant makes changes to a review, namely:

- When a review phase changes. Collaborator sends notifications when a review gets to the Annotating phase, or when it is Completed, Canceled or Rejected. Administrators can enable notifications for specific phases and disable them for others. (These notifications are sent both to active participants and to waiting participants.)
- When a review requires some action. For instance, Collaborator is clever enough to detect situations when a review stalls 202, and notifies the participants that the review requires some action in order for it to be completed.
- When a participant adds a comment to the review, or creates a defect. (These notifications are sent to participants who are waiting for review activity 345.)
- When participants add files to a review. (These notifications are sent to participants who are waiting for review activity 345.)
- When you add new participants to or remove them from a review.
- When review goes to Inspection phase. (These notifications are sent to active and waiting participants.)

The entire list includes several dozens of events. You can control your overall notification level only (see below 331). Administrators can enable or suppress specific notifications 288.

## When Collaborator Does Not Send Notifications

- Collaborator does not send notifications for reviews that are in the Planning phase.
- Collaborator does not send repeated notifications. If several events occur within the review, then only the notification of the first event is sent. After which, the notification recipients have their state set to "Active". They will not receive further notifications, except for phase change notifications.
- By default, Collaborator does not send notifications to the user who changed a review. (However, this can be controlled by the Notification Level 331 setting.)

See below 332 for information on why you can miss a notification.

## Setting the Desired Notification Level

1. Log in to the Collaborator Web Client (if you have not logged in yet).
2. Go to **Settings | Notifications**.
3. Choose the desired level from the **Notification Level** combo box. The following choices are possible:

| | |
|---|---|
| **None** | No notifications except for those that the administrator sends explicitly. |
| **Minimal** | Default. You will get notifications on other review participants' activities on a review. "Non-activity" notifications will not be sent. |
| **All** | You will receive all the available notifications. |

## Changing Notification Message Text

Regular users are not allowed to customize the text of notification messages. You need to have administrator permissions in order to customize the notification message text 288.

## Why Do I Fail to Get Notifications?

Below are some typical reasons for this:

- A review is in the Planning phase (Collaborator does not send notifications for such reviews).
- You are active participant of the review.
- Your role in the review is not one that would receive the notification in question.
- You chose the *None* notification level in your Collaborator account's settings. Check the settings and select either Minimal or All level.
- You are the user who modified the review and your notification level is *Minimal* (default). By default, Collaborator does not notify the user, who made some change, about this change. If you want to receive such notifications, change your notification level to *All*.
- The administrator could disable notifications of certain types or disable all the notifications. Please ask the administrator to check their settings.
- Your email client can have active filter or anti-spam rules that delete the notification emails or move them to a Junk folder. Please check the rules of your email client.

# 4.3 Performing Reviews

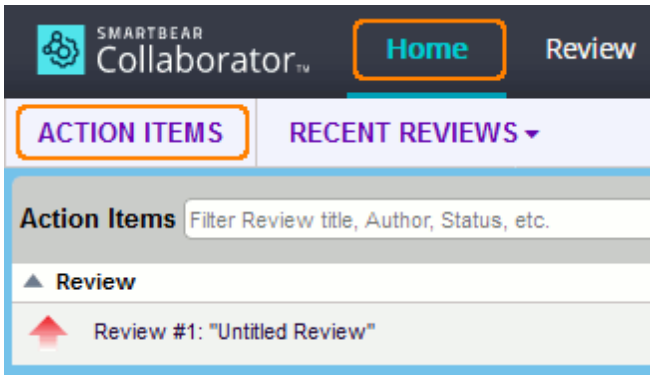
This chapter describes creating, performing, and managing peer reviews.

## In This Section

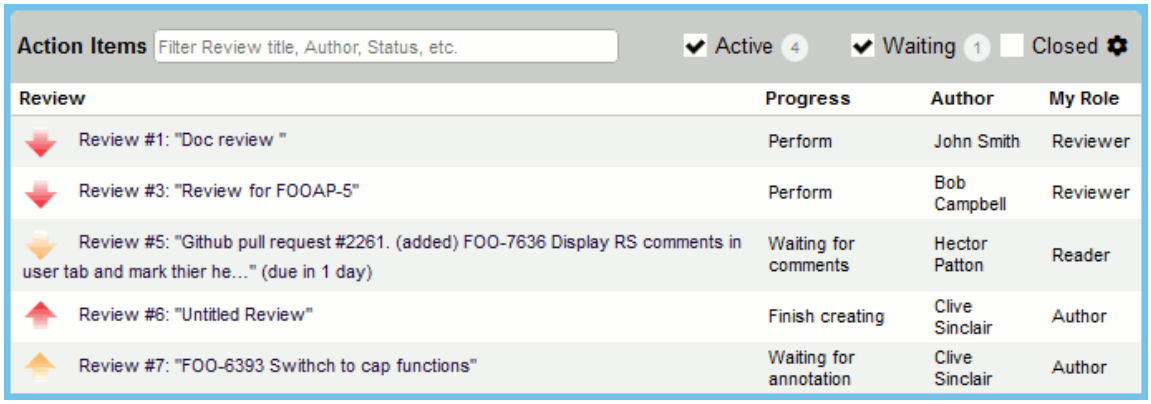
- Review Overview 17
  Describes a typical review and its phases.

## 4.3.1 Action Items

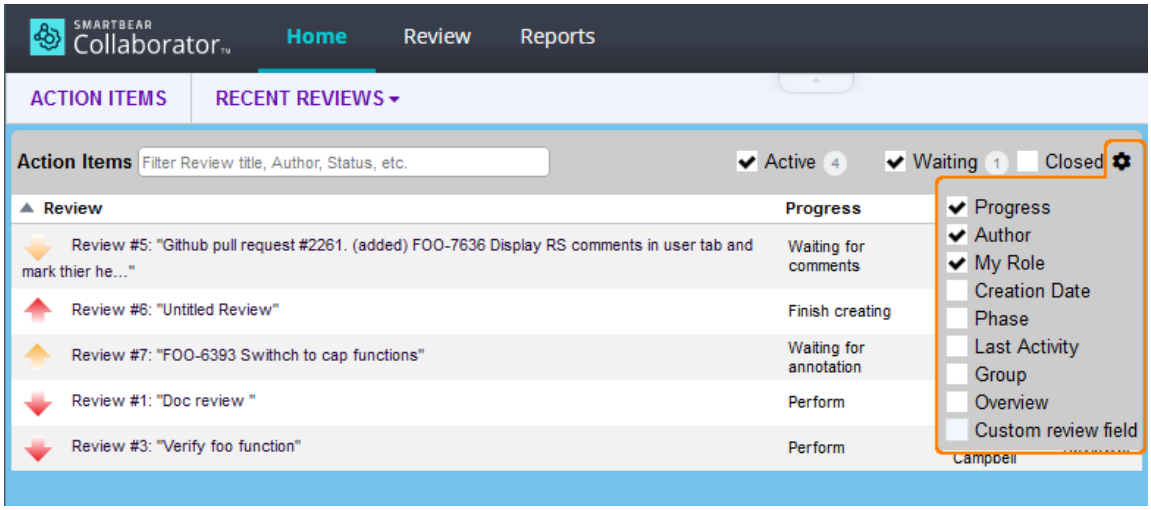
Action Items are relevant, current things that you are included in. You can view your list of action items by clicking either **Action Items** or **Home** toolbar buttons:



You can select which action items are shown, by selecting the check-boxes for "Active", "Waiting" and/or "Closed". Selecting "Active" displays all reviews that require your immediate action to proceed. Selecting "Waiting" displays all reviews awaiting for rework, comments, signature or other participants. Selecting "Closed" will list reviews that have recently been completed, cancelled or rejected.




You can configure what information will be displayed in the Action Items table. You can add or hide standard review fields as well as custom review fields. To do this, hover the ⚙ button and select the desired review fields in the drop-down list.




To filter some particular items, just specify the sought-for text in the filter box. This will display only the items that contain the specified text in the listed fields.

Additionally, you can sort action items by clicking on the column headers: "Review", "Progress", "Author", "My Role" and so on.

**Note:** Action Items cannot display more than 1000 reviews.

Your Action Item list refreshes automatically every five minutes. You can refresh this screen manually, as well.

When you have an Action Item directing you to commit files, you remove it by clicking the review link on the left. Collaborator uses these Action Items to track whether review materials have been submitted to version control. If Collaborator does not have enough information to determine that the materials are already versioned it will create a reminder for you when the review is completed.

For recently completed reviews the Action Items list displays the "Dismiss" link to confirm that the review has been finished and dismiss it from the list of your active reviews.

Action Items are also visible in other clients: in Tray Notifier 613, in Eclipse Plug-in 537, in Visual Studio Extension 578 and some others.

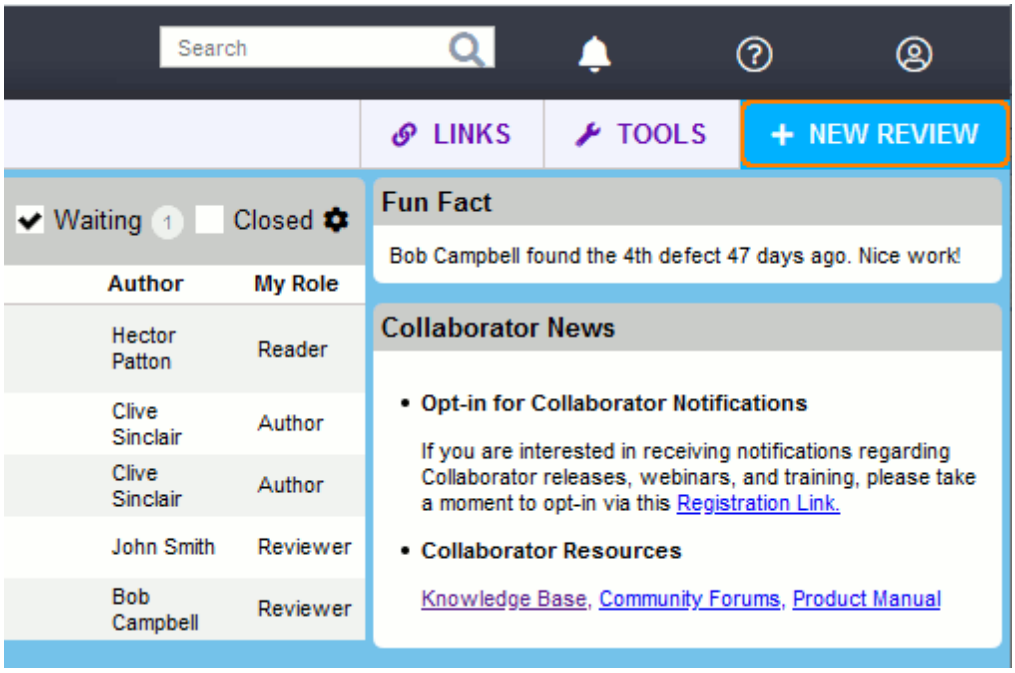
## Progress values for action items:

| | | |
|---|---|---|
| Annotate Review | Action | you have been invited by the author to annotate the review |
| Commit files | Action | you need to commit your files to source control |
| Finish Creating | Action | you need to finish creating the review |
| Fix Configuration | Action | not review related - you need to finish filling out your user configuration information |
| Perform | Action | you need to perform the review |
| Respond to Comments | Action | you need to respond to comments in the review |
| Review cancelled | Non-Action | the review has been cancelled |
| Review completed | Non-Action | the review has been completed |
| Review rejected | Non-Action | the review has been rejected |
| Rework defects | Action | you need to rework the defects in the review |
| Sign review | Action | you need to sign the review |
| Waiting for annotation | Non-Action | you are waiting for others to annotate the review |

| | | |
|---|---|---|
| Waiting for comments | Non-Action | you are waiting for others to make comments on the review |
| Waiting for defect rework | Non-Action | you are waiting for open defects in the review to be reworked |
| Waiting for other reviewers | Non-Action | you are waiting for others to complete the review |
| Waiting for signatures | Non-Action | you are waiting for others to sign the review |

### 4.3.2 Creating a Review

To create a new review, click the **New Review** button on the Home screen:




You will see a Create Review Screen where you can specify information about the newly created review.

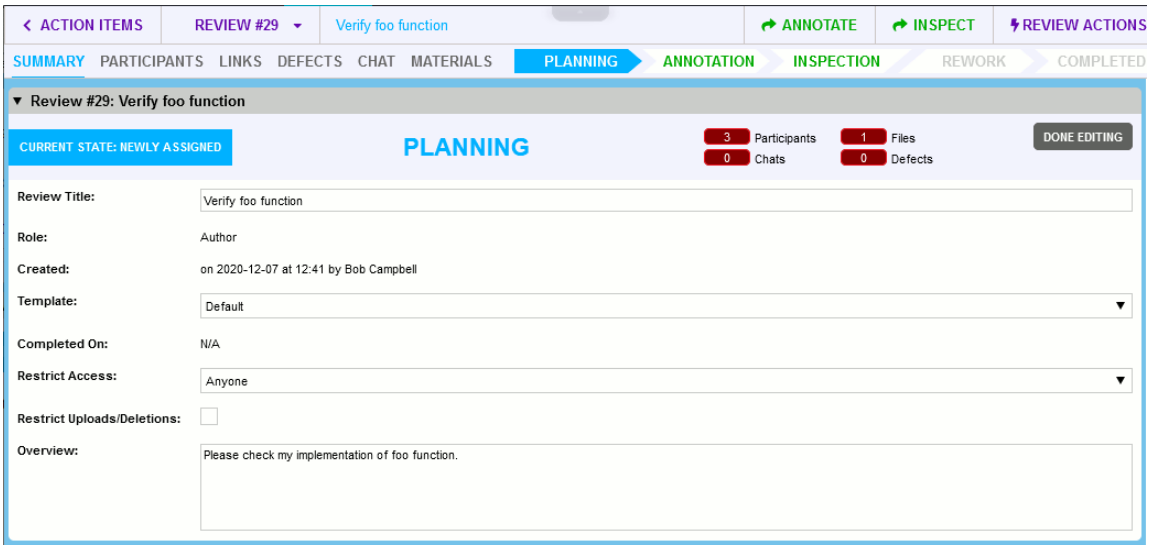
> **Tip:** Alternatively, you may create a new review by copying any of existing reviews. This approach allows to retain some data (participant list, custom fields values, review materials) from the original review. See Copying Previous Review 450 for details.

Creating a review involves the following steps:

- Specify General Information 337

- Assign Review Participants 338
- Add Review Materials 340
- Start the Review Procedure 344

## Specifying general information about a review




In the **General Information** section you can specify the following data about a new review:

- **Review title -** a brief description for the review. The title is used all over the place -- in Action Items 333, in notification emails, to web page titles, and so forth. Titles do not have to be unique in the system; reviews already have a unique ID that is assigned by the server.
- **Role** - your role in the current review. This field is read-only. Your and other participant's roles will be set through the Participants section described below.
- **Created** - a date when the review was created and a user who created it. This field is read-only and filled out automatically.
- **Group** - a group of users associated with the review. This field only appears when the Groups feature 226 is enabled and you belong to some user group. This field displays only the first 1000 groups and it excludes groups which have no members.
  Once you select a group, the participants list will be updated accordingly to show only those users belonging to the selected group. If you belong to only one user group, the group is set automatically.
  Changing the user group will cause you to lose any unsaved data applied to the review and may invalidate any saved data that is not applicable to the new group.

- **Template**- a template for the review. The template defines all the rules and workflow that governs the review. This includes custom fields for reviews and defects as well as the set of roles and rules for participants in the review. Your system administrator will have already configured 246 the templates you can pick from and probably has some guidelines as to which templates should be used under which circumstances.
  If there is only one configured template, this template will be applied automatically.

- **Deadline** - a date when the review should be completed. This field only appears if a system administrator has set the default review deadline 192 to a value other than 0. In that case, you will see the Deadline field with a default date that is calculated based on the value provided by the administrator. You can set the review deadline to any date that is on or after the current date. On the day before the deadline, participants will see a note in their action items that review is 'Due today'. This is intended to alert participants that if the review is not completed by midnight of the current day, it is overdue.

- **Completed on** - a date when the review was completed. This field is read-only and filled out automatically.

- **Restrict access** - allows you to limit those users who may access your review. By default, any user in the system can view and comment on any review at any time. However, there are circumstances where you may want to restrict viewing to those users specifically invited to the review, and this option enables you to do so. Please note that system administrators are able to view all reviews regardless of restricted access selections, and also have the ability to require 180 that all reviews be restricted. In such cases, you will not be given an option to restrict access.

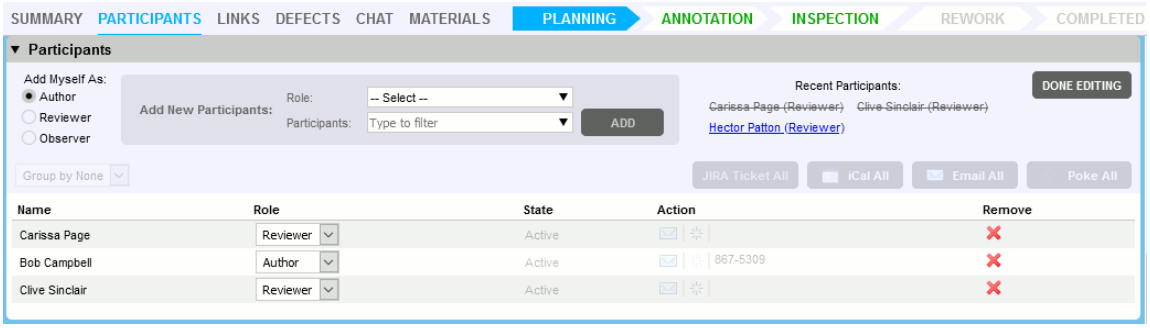
- **Restrict uploads/deletions -** allows you to limit who can upload files into the review and delete files from the review. If this setting is enabled, only the creator and the administrators will be allowed to upload or delete files.

- **Overview** - a detailed description of the review. This field is optional and does not have to be filled out if you would like to leave it empty.

- **<Custom Fields>** - any other review custom fields 256 that are enabled for current template.

## Assigning review participants

The next step in creating a review is to invite people to the review, and give each of them a role to play. In the Participants section you need to select your role on this review and add other participants.




The commonly-used roles are as follows:

- Author: Responsible for the files under review and probably responsible for fixing any defects that are found.
- Reviewer: Responsible for finding defects; is allowed to mark defects "fixed". The review is not finished until all reviewers approve it.
- Observer: Invited to the review but if all other participants are finished the review is complete. Useful when some users are not available just now and you do not want to wait for them to show up.
- Moderator: Controls the pace and flow of the review; coordinates other people and decides when the review changes phases.

Note, that the number of roles, their names and abilities are configured by your administrator[279]. Therefore the list of user roles on your server may be different from the list above.

To add yourself as a participant in the review, select a role in the **Add Myself As** area.

To add another user as a participant, select the desired role in the **Role** drop-down menu, then select one or more users in the **Participants** drop-down menu, and click Add. (The **Participants** drop-down menu displays only the first 1000 users.)


*Participants drop-down menu*

**Tips:** To filter the user list, start typing the name of the desired user. The list will display only those users that contain all typed characters.

To select multiple users, just click each desired user in the **Participants** drop-down list.

To select a range of users, click on the first user, hold down the `Shift` key and then click the last user.

To select all users (respecting current filter, if any), click **Select All** button.

To clear previous selection, click **Clear All** button.

The **Filter by** drop-down is used to choose a group whose members are then listed in the **Participants** drop-down menu. If there are no groups associated with the review, the **Filter by** drop-down is not displayed. This field displays only the first 1000 groups and it excludes groups which have no members.

User/role combinations you have used recently will also appear under **Recent Participants**. This makes it easy and fast to select common combinations.

To change a participant's role, select a new role from the drop-down menu in the **Role** column.

To remove a participant, click the **Remove from Review** button.

## Adding materials to a review

Review materials is what you are asking to review. Various types of materials can be uploaded into the review. This includes but is not limited to:

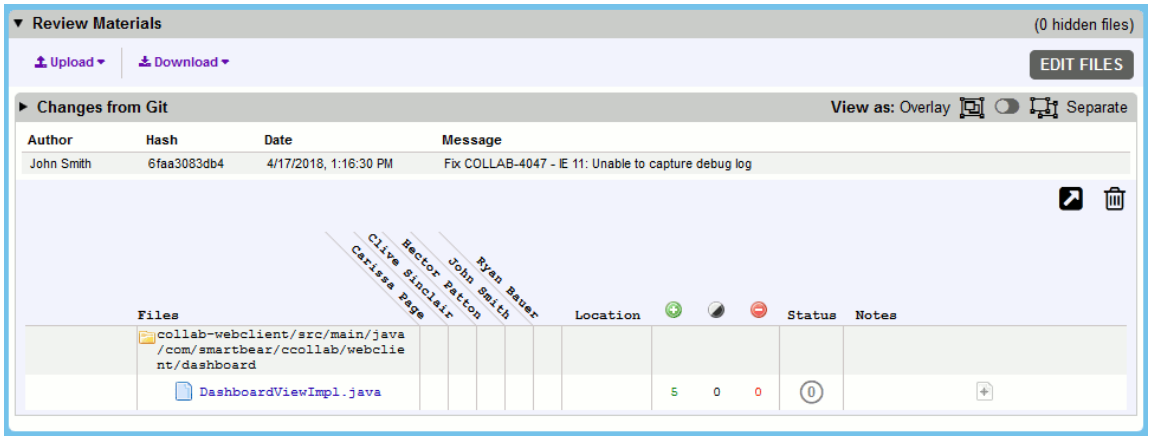
- Files controlled by a version control system: changes committed to a remote server (pushed commits), changes that are not yet committed to a server (un-pushed commits), local changes, differences between files, changelists, branches and so forth.

- Arbitrary files from your computer: text files 377, word processing documents 383, spreadsheets 389, presentations 395, PDF documents 414, Visio graphics 400, images 410 and so on.

Once you have attached anything to a review, it will be listed in the Review Materials section.

## Attaching review materials via desktop clients

The most common way to get files attached to a review is to use one of desktop clients: Command-Line Client 506, GUI Client 496, Eclipse Plug-in 525, Visual Studio Extension 566 or Plug-ins for Perforce Visual Client 786. All of these clients can integrate with version control system and/or easily upload local files or file-differences to the Collaborator server.




*Local changes from files controlled by Git have been uploaded into this review via client.*

## Attaching review materials via Web Client

It is also possible to upload content directly from the Web Client. The advantage of performing this action from a web browser is that you do not need to install and configure the client.

From the Web Client you can upload the following types of content:

- Changes committed to a pre-configured 204 version control system
- Arbitrary files from your computer

### Attaching changelists via Web Client

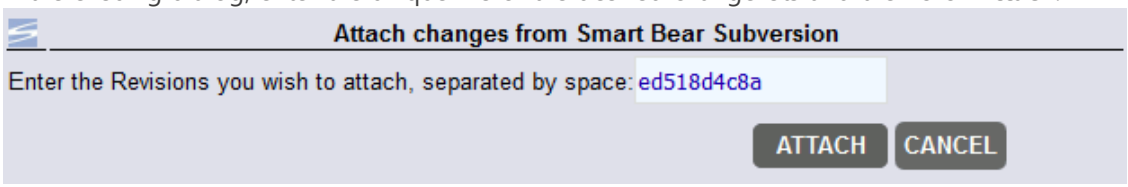
> In order to perform these steps, your Collaborator server must be connected to a version control system as described in Collaborator Administration - Version Control 204.

To add a changelist:

- Click the **Upload** button.

- Select the name of the connected version control system from the drop-down list:


- In the ensuing dialog, enter the unique IDs of the desired changelists and then click **Attach**.




You can attach any number of changelists to a single review. Also, you can attach changelists from different version control servers to a single review.

If there is an error retrieving that version or communicating with the server, you will see an error message.

**Attaching Live URL's**

You can attach live links to web sites. Attaching a live URL will display the live contents of the link during the review (rather than a local copy of the content). This feature is useful for accessing a document on a Wiki or document management system, particularly if you link to a specific version of that document.

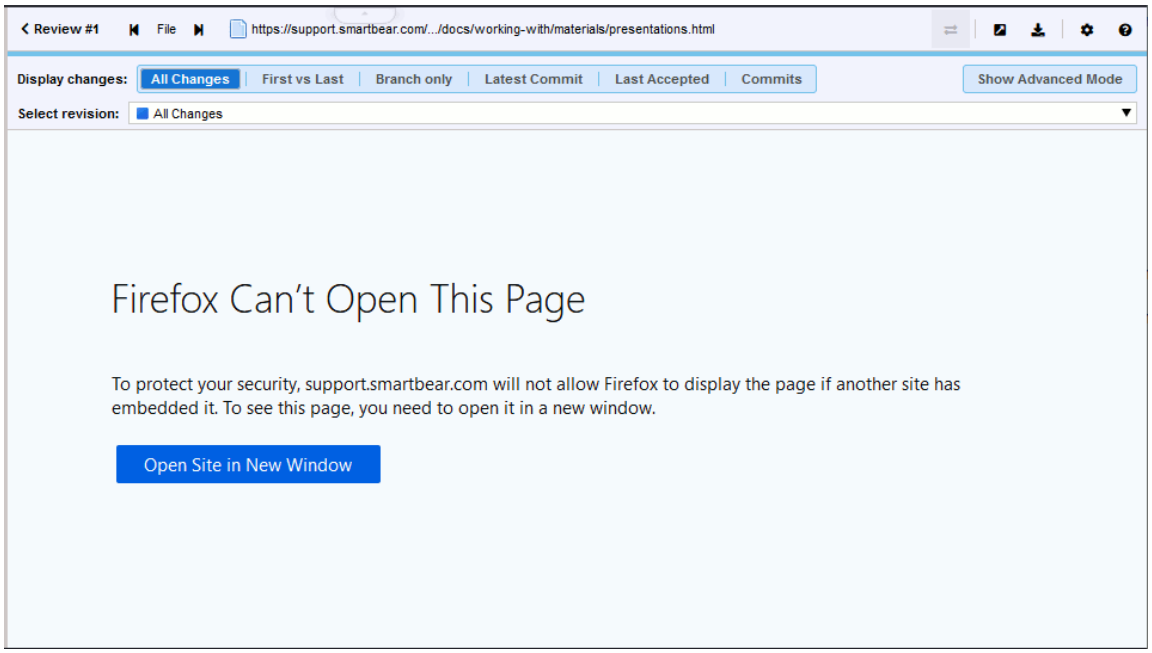
To add a web link:

- Click the **Upload** button.
- Select **URL** from the drop-down list.
- In the ensuing dialog, enter the desired URL and then click **Attach**.


**Notes:**

- The URL must be a valid web URL, which should include the protocol specification (either http:// or https://).

- You will also see an error message if the specified URL is malformed.
- Collaborator does not check whether the specified resource is available. In fact, since it is a link, the contents and availability of the document can change at any time.
- Collaborator displays the referred resource in a frame. The server can send requested contents with anti-framing headers. You will not see the referred content in this case. Depending on the web browser you use, you will see an empty frame or an error message saying that *the content cannot be displayed in a frame:*

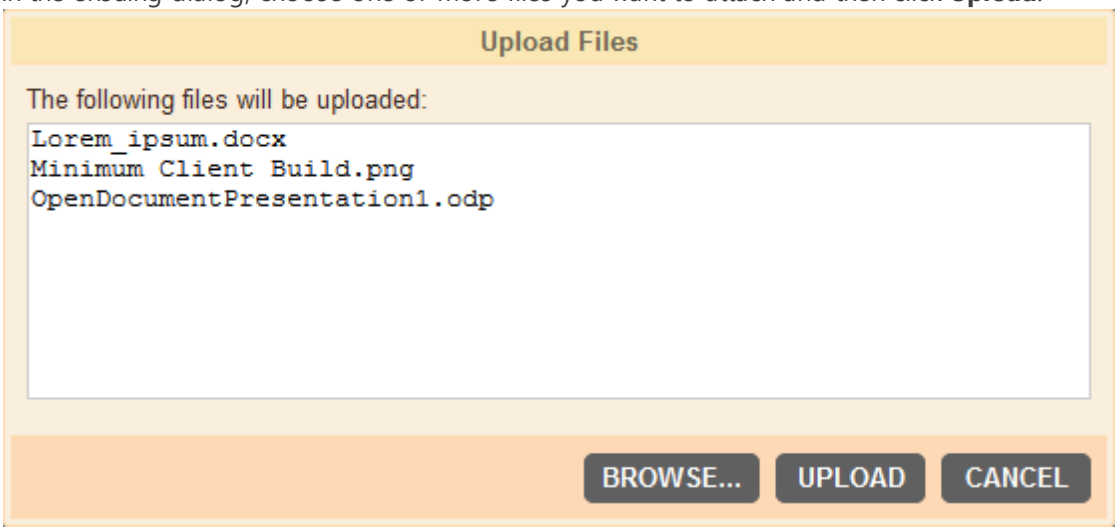



### Attaching arbitrary files

You can also attach any arbitrary file (images, PDFs, and binary file types) from your local hard drive:

- Click the **Upload** button.
- Select **Files** from the drop-down list.

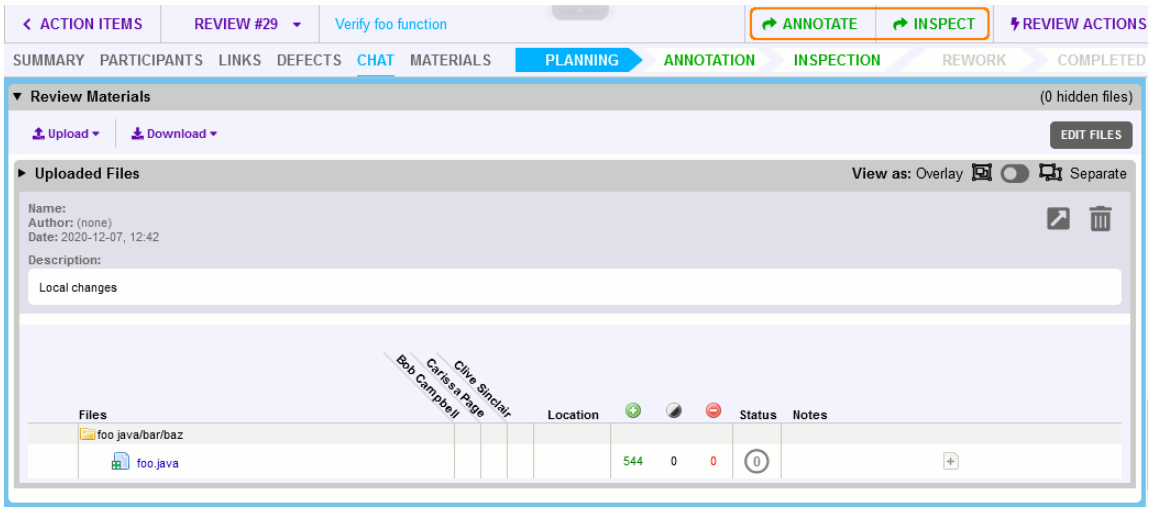
- In the ensuing dialog, choose one or more files you want to attach and then click **Upload**.




**Tip:** Alternatively, you may just drag your local files and drop them to review web page.

## Completing review creation

When the materials are uploaded, you can start the review immediately or keep it for annotating:

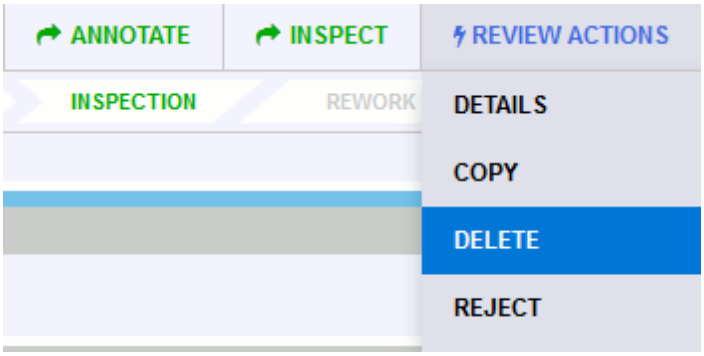



Click **Inspect** button in review header to start a review. Other participants will receive notifications about your new review, and it will be displayed in their Action Items list.

Alternatively, you can click **Annotate** button to apply the current changes, and proceed to optional Annotating phase. In this phase all participants that you have added will be notified about this review and will be invited to add their comments and review materials. The review will not begin until you select the **Inspect** button.

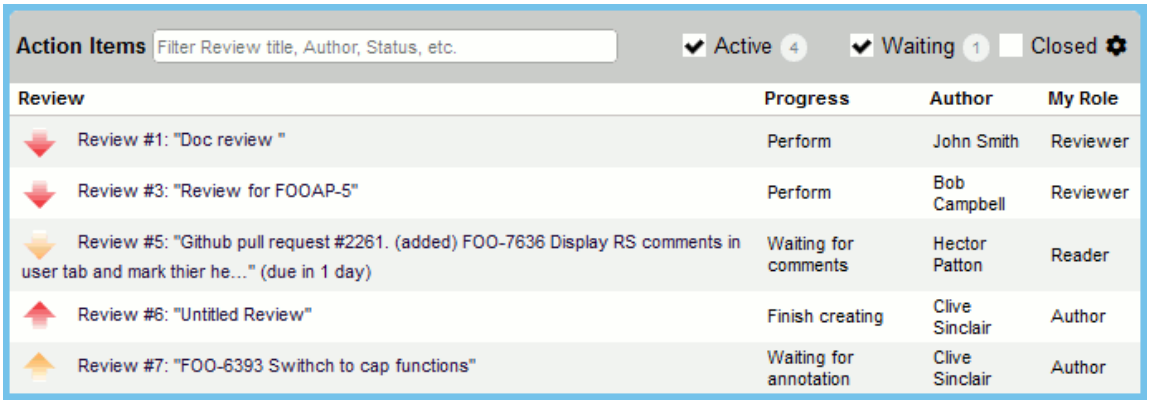
## Deleting a review before it starts

To cancel a review before it begins, select **Review Actions | Delete** in review header:




### 4.3.3 Review Summary Screen

You will know that you are part of a review because it will show up in your Action Items list[333], clicking a review in the list will take you to the Review Summary screen.

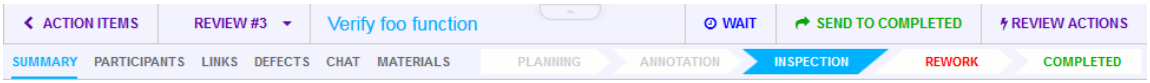



The Review Summary Screen is displayed any time you are creating, editing or participating in a review. This page includes a fixed header-toolbar and a number of sections:

- Header Toolbar[346]
- Review Steps[347]
- General Information[348]
- Checklist[350]
- Participants[352]
- Participant Custom Fields[353]

- Remote System Links 353
- Defect Log 354
- Overall Chat 356
- Review Materials 357

## Header Toolbar




The header of Review Summary Screen has fixed position and is always displayed whenever you scroll the screen up or down. It contains the following items:

| | |
|---|---|
| **Action items** | Returns back to Home page. |
| **Review #** | Displays the ID of current review. On mouse over, displays links to 10 most recent reviews, so you can navigate to another review directly. |
| **<Title>** | Displays the title of current review. |
| **Annotate, Inspect, Wait, Send to Rework, Send to Completed** | Moves the review from current phase to the next phase. The actual number of available actions depend on current review phase. See Review Steps 347 section below. |
| **Review actions** | Displays a list of various actions available for the current review. The actions are:<br><br>**Details** - Opens the Review Details Report 473 on current review.<br><br>**Copy** - Creates a new review on the basis of current review. You may copy list of participants, custom field values, review materials. See Copying Previous Review 450.<br><br>**Delete** - Deletes a review before it has started. Deleted review cannot be reactivated even by administrator.<br><br>Displayed only on planning phase and only for review authors/creators. Enabled if authors are allowed to delete their reviews 191. |

When deleting a review, Collaborator deletes files that were uploaded for this review from the content storage 97 as well (unless they are used in some other reviews).

**Cancel** - Cancels an in-progress review. Canceled review can only be reactivated by administrator. Users will still be able to find cancelled reviews when searching and can view them in read-only mode.

Displayed only for review authors/creators. Enabled if authors are allowed to cancel reviews 191, otherwise only administrators can do this.

**Reject**- Rejects an in-progress review. Could ask to specify reject reason 192 in ensuing dialog. Rejected review cannot be reactivated even by administrator.

Enabled if participants are allowed to reject reviews 191, otherwise only administrators can do this.

**Reopen** - Reopens a review after it was finished.

Displayed for closed reviews. Enabled if participants are allowed to reopen reviews 191, otherwise only administrators can do this.

**Archive** - Creates an archive 165 containing all the information related to the review.

Enabled if participants are allowed to archive reviews 188 and only on appropriate phases 189 (on completed, cancelled or rejected phase, or on any phase).

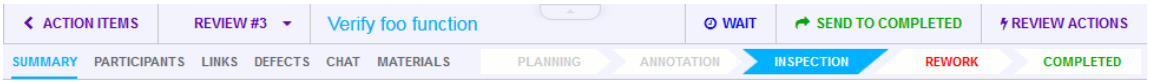
| | |
|---|---|
| **Summary, Participants, Links, Defects, Chat, Materials** | Navigates to the respective section of the Review Summary Screen. Currently displayed section is underlined. Each section will be described below. |
| **Planning, Annotation, Inspection, Rework, Completed** | Displays the review workflow and phases. Current phase of the review is highlighted by solid-fill background. |

## Review Steps

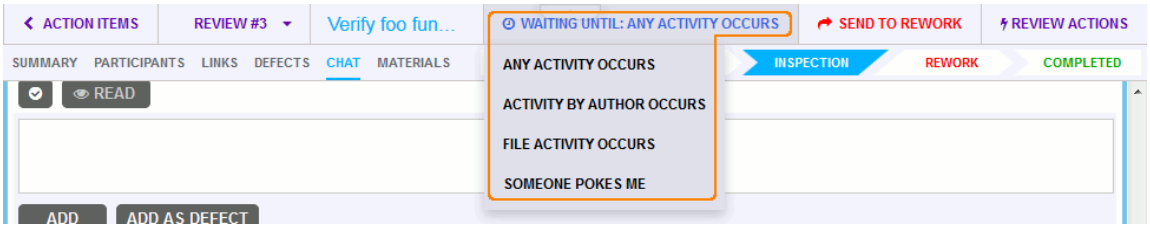
The Review Steps section of Review Summary header toolbar allows each review participant to move the review from one phase to another, or to wait in the current phase until certain specified activity occur. To learn more about review phases, see the Review Overview 17 topic.

The choice options vary depending on the current and the expected phases of the review and the role of the participant. The choice options can be: Annotate, Inspect, Wait, Send to Rework, Send to Completed, Sign, Decline and so forth.

Pressing any of the Review Steps buttons (except for Wait) indicates that you have passed through the current review phase and moves the review to another phase.




Pressing the Wait button keeps the review in its current phase.



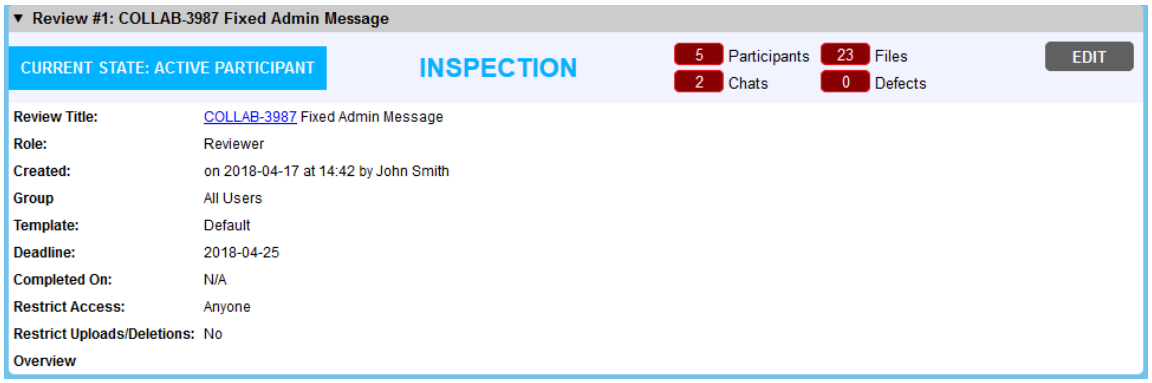

*Wait Options*

Most of the choices also allow to select in which case Collaborator will notify you of certain activities related to the given review. The action options are:

- **Any activity occurs** - any action that occurs in the review; for example, comments, file uploads, or added participants
- **Activity by author occurs** - any action that occurs in the review that is initiated by the author
- **File activity occurs** - any action that occurs to change the file content of the review; for example, the upload of a new file or the upload of a new version of an existing file
- **Someone pokes me** - only notifies you in the event that you are poked

## General Information Section

The General Information section provides basic information about the review: current state, number of participants, uploaded files, chat messages, found defects and so on.




This section includes the following fields:

- **Review title -** a brief description for the review. The title is used all over the place -- in Action Items 333, in notification emails, to web page titles, and so forth. Titles do not have to be unique in the system; reviews already have a unique ID that is assigned by the server.

- **Role** - your role in the current review. This field is read-only. Your and other participant's roles will be set through the Participants section described below.

- **Created** - a date when the review was created and a user who created it. This field is read-only and filled out automatically.

- **Group** - a group of users associated with the review. This field only appears when the Groups feature 226 is enabled and you belong to some user group. This field displays only the first 1000 groups and it excludes groups which have no members.
  Once you select a group, the participants list will be updated accordingly to show only those users belonging to the selected group. If you belong to only one user group, the group is set automatically.
  Changing the user group will cause you to lose any unsaved data applied to the review and may invalidate any saved data that is not applicable to the new group.

- **Template**- a template for the review. The template defines all the rules and workflow that governs the review. This includes custom fields for reviews and defects as well as the set of roles and rules for participants in the review. Your system administrator will have already configured 246 the templates you can pick from and probably has some guidelines as to which templates should be used under which circumstances.
  If there is only one configured template, this template will be applied automatically.

- **Deadline** - a date when the review should be completed. This field only appears if a system administrator has set the default review deadline 192 to a value other than 0. In that case, you will see the Deadline field with a default date that is calculated based on the value provided by the administrator. You can set the review deadline to any date that is on or after the current date. On the day before the deadline, participants will see a note in their action items that review is 'Due today'. This is intended to alert participants that if the review is not completed by midnight of the current day, it is overdue.

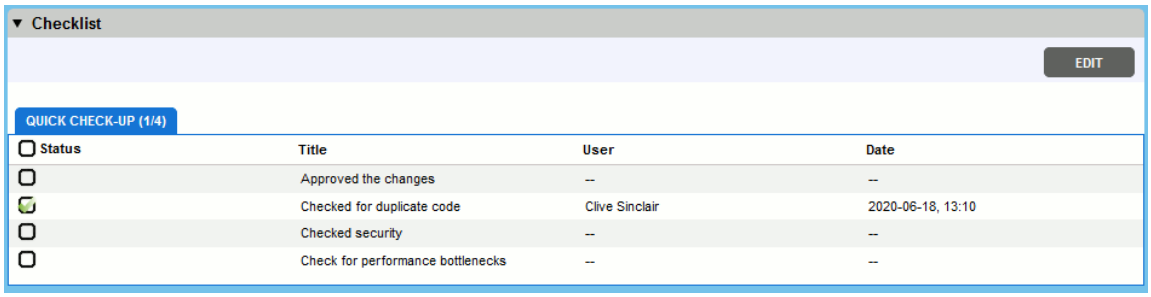
- **Completed on** - a date when the review was completed. This field is read-only and filled out automatically.
- **Restrict access** - allows you to limit those users who may access your review. By default, any user in the system can view and comment on any review at any time. However, there are circumstances where you may want to restrict viewing to those users specifically invited to the review, and this option enables you to do so. Please note that system administrators are able to view all reviews regardless of restricted access selections, and also have the ability to require 180 that all reviews be restricted. In such cases, you will not be given an option to restrict access.
- **Restrict uploads/deletions -** allows you to limit who can upload files into the review and delete files from the review. If this setting is enabled, only the creator and the administrators will be allowed to upload or delete files.
- **Overview** - a detailed description of the review. This field is optional and does not have to be filled out if you would like to leave it empty.
- **<Custom Fields>** - any other review custom fields 256 that are enabled for current template.

Unless you are creating a new review, most of section fields are in read-only mode. To modify field values, click **Edit**, make your corrections and then click **Done editing** to submit your changes.

## Checklist Section

Depending on the configuration of the review template 246, the Review Screen can have one or multiple checklists 272 for a review. A *checklist* is a set of items with check boxes. You may want to use these lists to remind review participants about certain actions or verifications to perform on the review, for instance, to check the spelling and punctuation or to verify that the reviewed files match style guides adopted in your organization.

You can see these lists in the Checklists section of the review screen (this section is hidden, if the review template doesn't use any checklists):



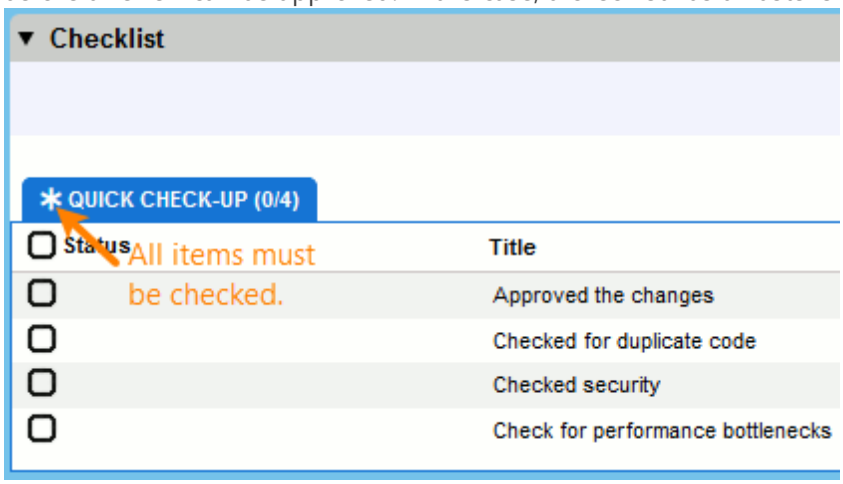

Collaborator track all activities on checklists. For example, when a review participant selects a check box in the checklist, Collaborator logs their name and the timestamp of the change. Also, it adds a message about the change to the Chat section of the review. If a participant clears some check box, Collaborator also adds a message about this to the Chat section.

For auditing purposes, you can include checklist status changes into the Review Details Report[473]. Make sure that **Checklist History** is set to *Display Checklist History* for this.

To learn how to create and modify checklists, see administrator reference[272].

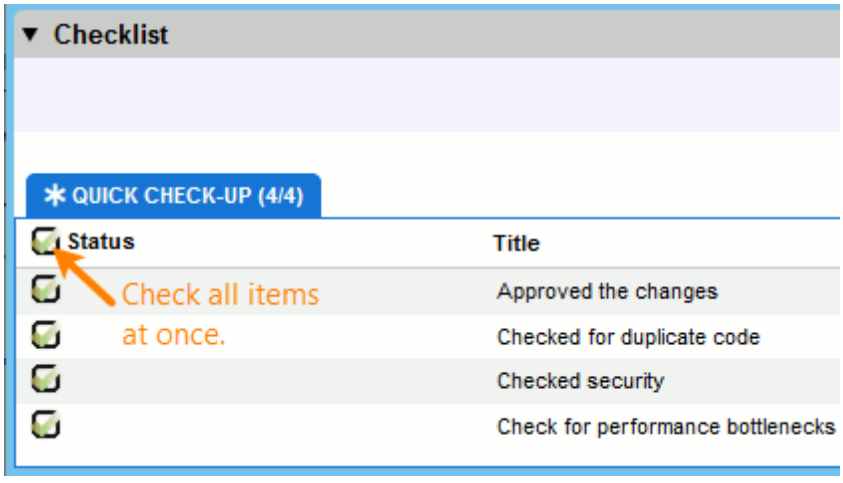
## Checklist actions

- You can sort checklist items by Status, Title, User or Date.
- Depending on the review template's settings, you may be required to check all the items before a review can be approved. In this case, a checklist has an asterisk in its title:

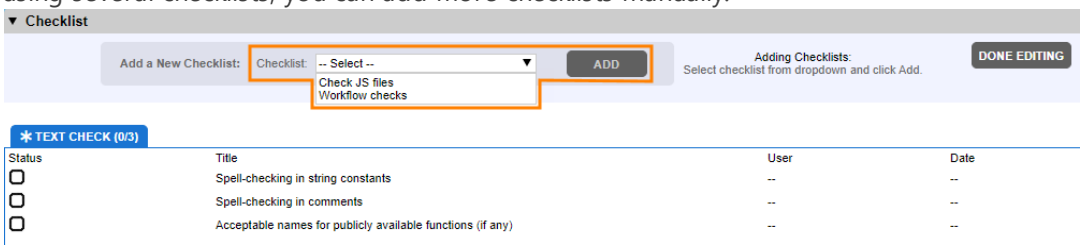



Asterisk in the checklist title

- To check all items at once, click the check mark in the Status column header:



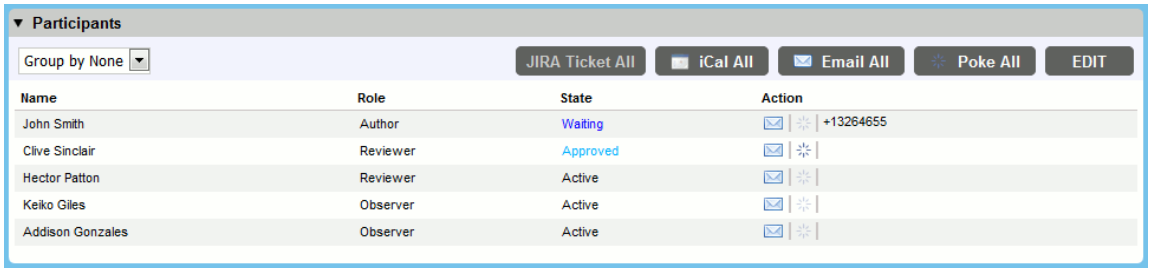
- By default, only one checklist is visible on the Review Screen. If the review template allows using several checklists, you can add more checklists manually:




Adding a checklist to a review

## Participants Section

The Participants section lists all participants in the review, their statuses, and their roles:




The drop-down list at the top left allows you to select the way participants are grouped when displayed in this section. The options are "Group by Role", "Group by State" or "Group by None". The "Group by None" option lists all participants in the review in a single list which is sortable by clicking the headers for each column (name, role, state).

From the Participants section you can send different types of notifications to other participants. Collaborator has the following types of notifications:

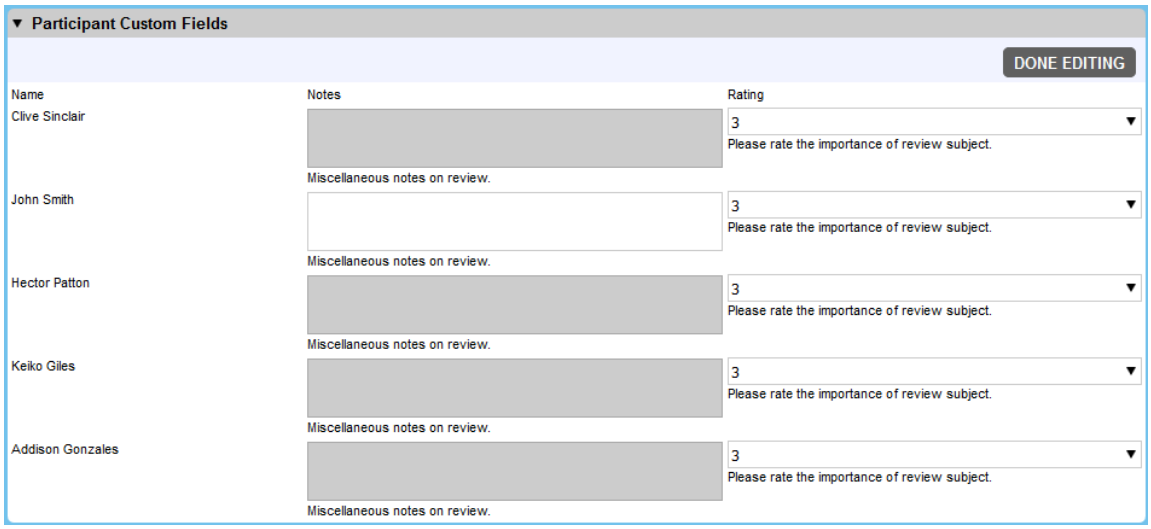
- **E-mail** - Launches your e-mail client and creates a draft letter which you can edit as you like and then send it.
  To send an e-mail notification to a specific participant, click the Email icon (✉) next to the desired participant.
  To send e-mail notifications to all participants, click the "Email All" button at the top of the Participants section.

- **Poke** - Sends an e-mail with auto-generated content[299]. Poke notifications may be useful when a certain participant has stalled the review or when you would like to invite a "finished"/"waiting" participant back into the review.
  To send poke notification to a specific participant, click the Poke icon ( ) next to the desired participant.
  To send poke notifications to all participants, click the "Poke All" button at the top of the Participants section.

- **Calendar** - Creates an email with a meeting request in iCalendar format. Calendar notifications[451] may be useful for scheduling formal meetings on review.
  To send Calendar notifications to all participants, click the "iCal All" button, in the ensuing Collaborator iCal Invite dialog specify proposed date and time and click Send.

## Participant Custom Fields Section

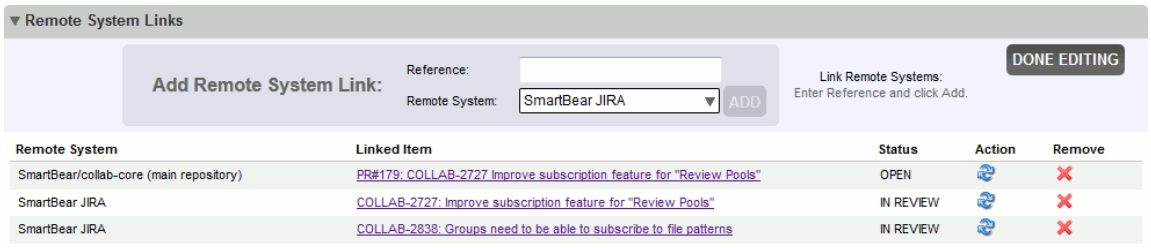
This section complements the Participants section and displays the values of participant custom fields[266] (if they are established and enabled for current review template).




By default, participants can modify only their fields (yet, this is configured[266] per each custom field). Fields of other participants are displayed in read-only mode.

## Remote System Links Section

If your current review template allows, the Remote System Links section is displayed. This section lists all pull requests, direct pushes and issue-tracking items linked with this review.




The list automatically populates with the following items:

- Links to the remote repository for reviews created on pull requests and direct pushes to remote repositories. (If any of repository hosting integrations 826 is established and running.)
- Build statuses of your CI systems (Jenkins, Travis and so forth) for reviews created on pull requests of GitHub repositories. (If GitHub integration 826 is established and running **and** your repositories use continuous integration.)
- Links to issue tracking item identifiers that occur anywhere within the review (title, custom fields, comments and so on). (If any of issue-tracking integrations 886 is established and running.)

The Remote System Links section also displays current statuses of linked items. Item statuses are updated upon webhook events (if a remote system supports webhooks), or periodically (default interval is every 2 minutes and can be adjusted via VM option 1232). Also, participants may refresh item status manually.

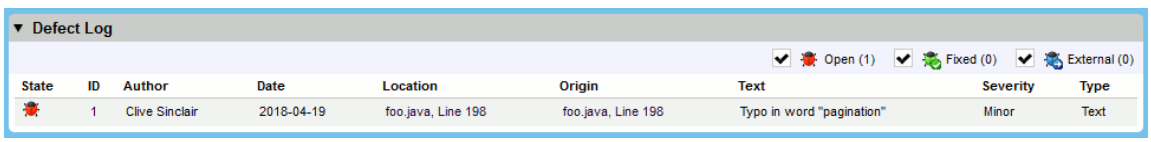
Besides that, participants may remove linked items or append links to issue-tracking items. Links from remote hosting systems, are only added automatically, they cannot be appended manually.

To append link to an issue-tracking item, do the following:

1. Type the item identifier (in JIRA it typically consists of project key and ordinal number, e.g. COLLAB-2838) into the **Reference** field,
2. Select the desired issue-tracking configuration in the **Remote System** combo box.
3. Press **Add**.

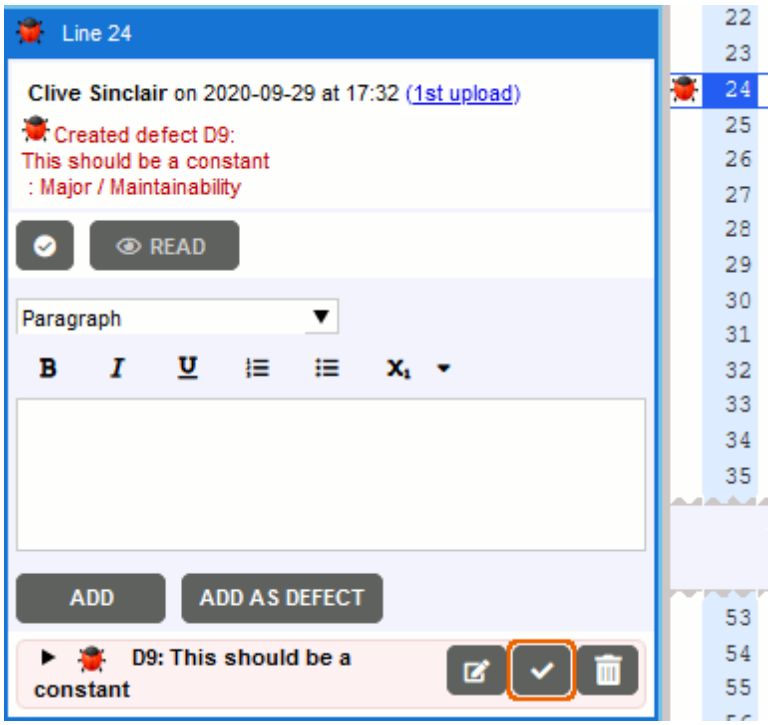
## Defect Log Section

The Defect Log lists all defects found in the review:

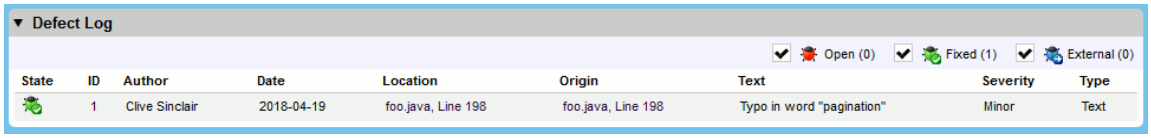


Initially the log will be empty. As defects are created for the review as a whole or on individual files and line numbers, all defects are collected and listed here. For defects associated with particular files, links to that file and line number appear in the table for fast access. All defect custom fields are shown in the table.

When, as the reviewer, you have verified that a defect you found has been fixed, you can indicate this by clicking **Mark Fixed**.




The defect is then redrawn to reflect this:



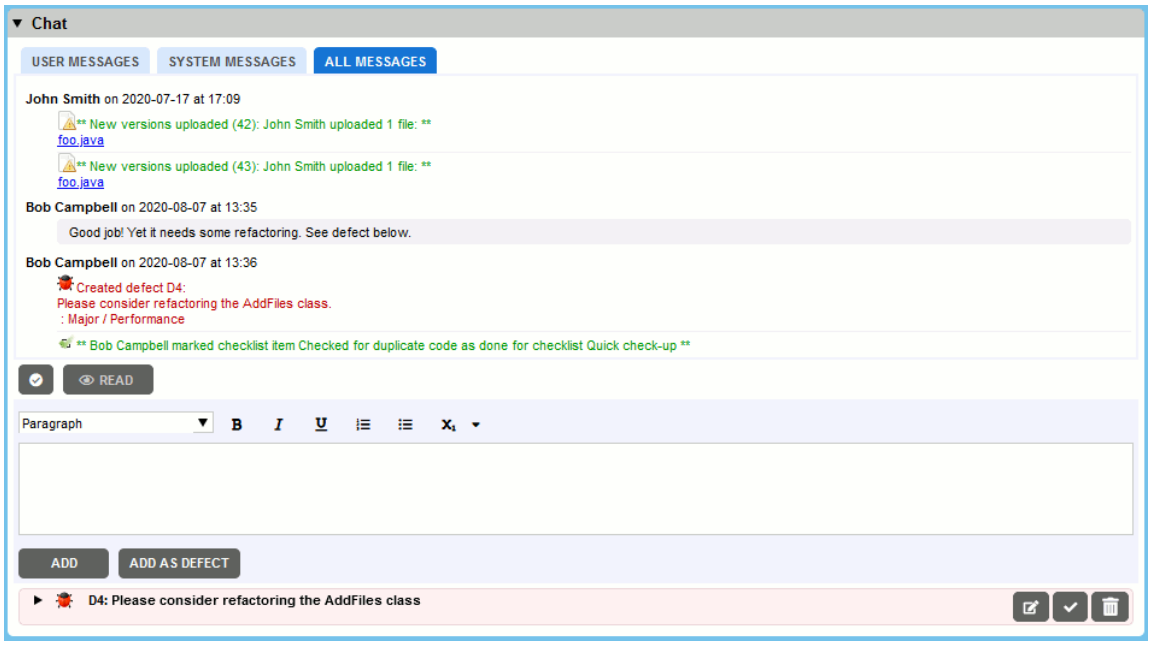

Only certain participants will be able to mark defects fixed. Which roles are allowed to do this depends on the system configuration[279]. Also the user that created the defect is allowed to mark it fixed, and administrators are always allowed to mark any defect fixed.

Most of the time you will create defects[436] and mark them fixed[438] on individual lines (in text files) or on specific areas (in other types of documents).

To learn more on creating and editing defects, see the respective section of Review Chats, Comments and Defects[436].

## Overall Review Chat

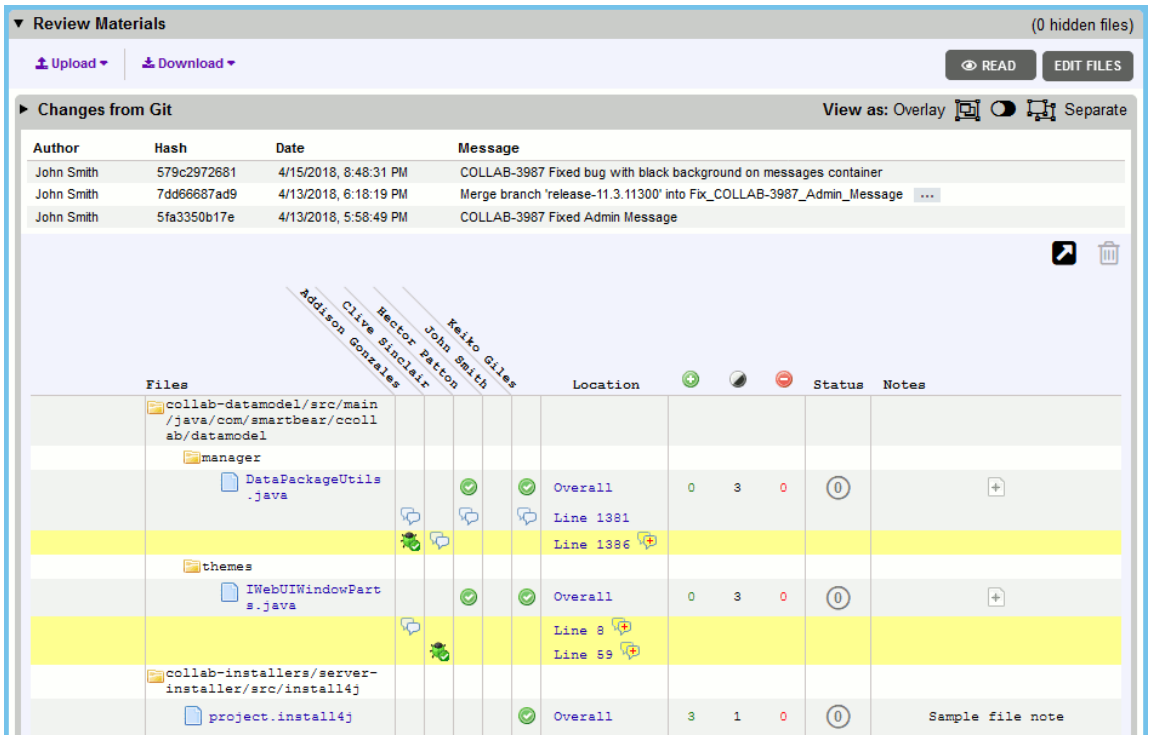
You can chat with other participants about this review as a whole, rather than chatting on individual lines[429]. Chat section contains three tabs: **User messages** tab displays comments and acceptance marks of other participants, including user comments from remote repositories, while **System messages** tab displays system information related to current review: defects, file uploads, marking checklist items and so forth, and **All messages** tab displays participant and system messages altogether.




You can also use this section to enter defects related to the entire review. Clicking **Add as defect** button will change the interface to include a drop-down list that will allow you to specify defect information like Severity, Type and so on.

There is much to say about chat and defects, but this interface is identical to the one in the Review Chats[429], so please see that section for details.

## Review Materials Section




The Review Materials section is the heart of the review: All files are displayed and participants can view content and differences and create comments and defects on specific lines or coordinates within a file.

Below, we will describe each of its elements in detail.


*Toolbar*

The toolbar of the Review Materials section holds the following items:

**Upload**

Displays a list of actions for uploading data to current review. Uploads can be disabled by review creator 338 and by administrator 188. The actions are:

**<Name of Version Control System>** - Upload changes from one of pre-configured 204 version control repositories.

**Files** - Upload arbitrary files from your computer.

**Tip:** Alternatively, you may just drag your local files and drop them to review web page.

**Simulink Archives** - Upload Simulink models[404] that have been exported as web view archives.

**URL** - Attach live URL link.

**Local Changes** - Upload local files using the command-line client, GUI Client, or P4V integration.

**Download**

Displays a list of actions for downloading review materials to your local computer. The actions are:

**Files** - Download a ZIP file containing all files *with subdirectory structure preserved* .

This means you can test the proposed file changes locally: Just download the ZIP file and expand it in your own development environment. This is also useful in the single-committer model for when you want to actually commit the changes. This can take the place of a patch file.

> Warning: Downloading files to your local hard drive can have unintended consequences. Make sure you do not have changes of your own that you are overwriting. Also, remember that the author of these changes might not be synched to exactly the same versions of all files in version control, so if your local test fails this might be the reason.

You can also download individual file versions from the Diff Viewer[377], but the Download Files toolbar link is the more common way because you get all files at once.

**Diff** - Download a unified diff of all files in the review.

**Read**

Allows to mark all comments in all review materials as read. Use with caution, as this operation makes it extremely easy for someone to mark all comments in all review materials as read without actually reading them. Therefore typically this action is disabled by administrators[191].

**Edit files**

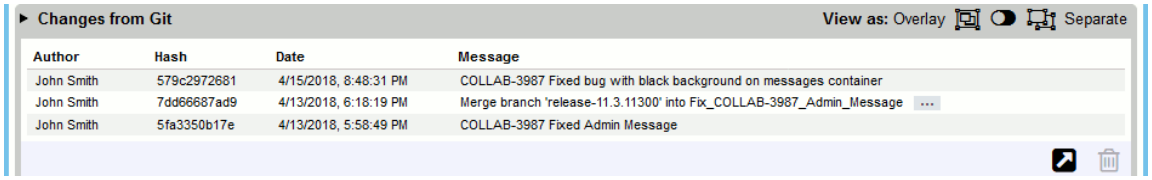
Allows to change the list of review materials by hiding files that do not need to be reviewed. Hidden files will not be listed in Files section and their content will not be displayed in the Diff Viewer.

By default, only review authors can hide files, and they can perform this only in the Planning phase. To learn more about this feature, see Hiding Files From Review[453].

**View as** Allows to change the format of how the review materials are listed. If a review contains only one changelist, this command is inactive. The view options are:

**Overlay** (default) - Display all files from multiple changelists or uploads in one list.

**Separate** - Display files from each changelist or upload in a separate list.




The **Changelist info section** displays information about the uploaded files - who and when has submitted files, version control system, changelist IDs, commit messages, diff command and so on. Depending on how the materials were added, this information may vary a little. In Overlay view this section combines information from each changelist, while in Separate view this section precedes each particular changelist. If the commit message is too long, or consists of multiple lines, then the message is trimmed at 70 characters. To see the full message, press the ellipses button. The section has two more buttons:

**External Diff** Download a diff file for viewing in an external Diff Viewer.

**Delete** Deletes a changelist from the review materials. If you have multiple changelists in the review, you will need to go to separate view in order for the button to be enabled. You cannot delete an arbitrary file from a changelist. You can only delete the entire changelist. You cannot delete changelists that have comments to any file. Ability to delete review materials can be disabled by review creator[338] and by administrator[188].

To exclude unneeded files from review, we recommend hiding files rather than deleting a changelist. Read Hiding Files From Review[453] for details.


*File list*

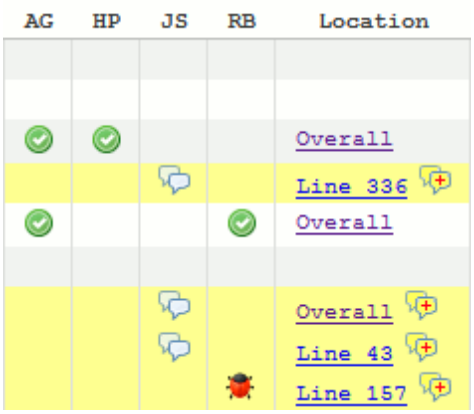
The **Files** column shows the files in this review. This part of the screen may look different depending on the selected *View as*[359] setting. The icon next to the file name indicates what operation was performed to the file.

| Icon | Meaning |
|---|---|
|  | File was modified |
|  | File was added |
|  | File was reverted before the review has started |
|  | File was deleted |
|  | File was reverted during the review |

To open the content of a file in the Diff Viewer[364] and view all comments associated with that file, just click its name in the file list. In Overlay mode of review materials, Diff Viewer will display overall changes made to the file as specified by the *Default Revision Comparison of Diff Viewer*[324] setting, while in Separate mode it will display changes made by a particular changeset.

The **Utility** column may precede the **Files** column, it groups various actions that could be performed over the specific file. For instance, it can display the **Hide** check boxes for hiding files 453, and the Upload new version button for uploading new revision under a different file name 455.

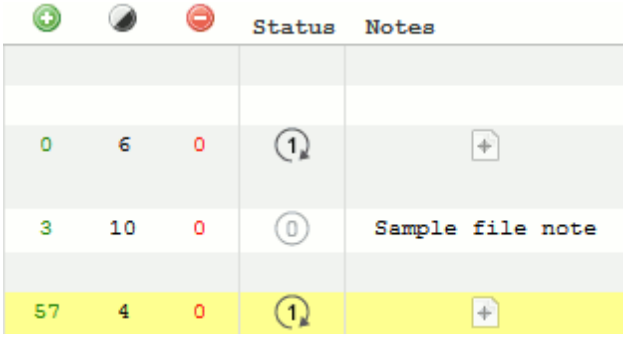
The **Conversation** column of the Files list shows where comments and defects have been made, their location within a file as well as whether these comments have been approved by other participants.




*Conversation status icons and Location column*

| Icon | Meaning |
|---|---|
| | This participant has made a comment |
| | This participant has clicked "Mark Accepted 433" for file revision or comment. When the author uploads new revision of a file, the icon will change back to denoting that the participant has not yet approved this new revision. |
| | This participant has created a defect, which is currently "Open" |
| | This participant has created a defect, which is currently marked as "Fixed" |
| | This participant has created a defect, which is currently marked as "Tracked Externally" |

The **Location** column indicates where within the file the conversation represented by this row is located. Yellow alert bubbles mean there is activity on that line that the current user has not yet marked read. Click on a line number to open the content 377 of that file and view that conversation. When the respective document is being converted, the **Location** column displays conversion progress indicator instead.




*Metrics, Status and Notes columns*

The next three columns show the number of text lines or lines of code (LOC) that have been added, changed, and deleted, with special cases for situations like added or deleted files.

| Icon | Meaning |
|---|---|
| | The number of lines that have been added |
| | The number of lines that have been modified |
| | The number of lines that have been removed |

The lines of code metrics (LOC metrics) are calculated only for source code and other text-based files. For other types of review materials (Word, Excel, PDF or Image files) the metrics are not calculated and return 0.

If the Overlay view is selected (default), the LOC metrics are calculated comparing the latest uploaded revision of file against the baseline revision of that file. Here, the baseline revision stands for the revision at the moment the review was created.

If the Separate view is selected, the LOC metrics are calculated comparing each individual file revision against its previous revision.

For reviews created by pull requests 826, file changes made by merge commits (if any) are displayed in Separate view, but they are not taken into account when calculating overall LOC metrics.

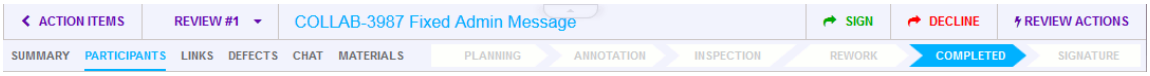
The lines of code metrics and conversation positions are calculated including whitespace characters, letter-case changes and COBOL sequence numbers (if any). Diff Viewer has options to ignore those when displaying file difference. When these options are enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports.

The **Status** column shows how many times the file has been reworked during this review. For reviews created by pull requests, file changes made by merge commits do not affect the overall rework count of a file.

The **Notes** column allows any user to annotate the files. To edit the note click in the note column. This is typically used for messages to all reviewers such as "review this file first", or "ignore this file".

#### Electronic Signatures

When electronic signatures[193] are enabled, users will see Sign and Decline buttons in the Review Steps section for completed reviews. Clicking either button requires a user to provide their login and password (or only login, when single sign-on is enabled) to complete the action. This makes the electronic signature compliant with FDA regulations.




When a reviewer signs or declines a review, the Review Steps section displays, "This review has been signed off" or "This review has been declined". The Chat section shows the signature information. Collaborator notes the name of the reviewer and the date and time the review was signed or declined.

For auditing purposes, you can go to the Review Details Report. Make sure that Comments Section Format is set to Display All Comments, and in the Overall Review Conversation section, you can see the signature information.

## 4.3.4 Reviewing Materials

The Review Materials screen, or the Diff Viewer, is where most of the reviewing takes place. The commenting format and functionality will differ depending on what file type you are reviewing. We support the following file types:

- Source code and text files[377] (any text-based formats)
- Word processing documents[383] (.doc, .docx, .rtf, .docm, .dot, .dotm, .dotx, .odt and .ott)
- Spreadsheets[389] (.xls, .xlsx, .xlsb, .xlsm, .xltm, .xltx and .ods)

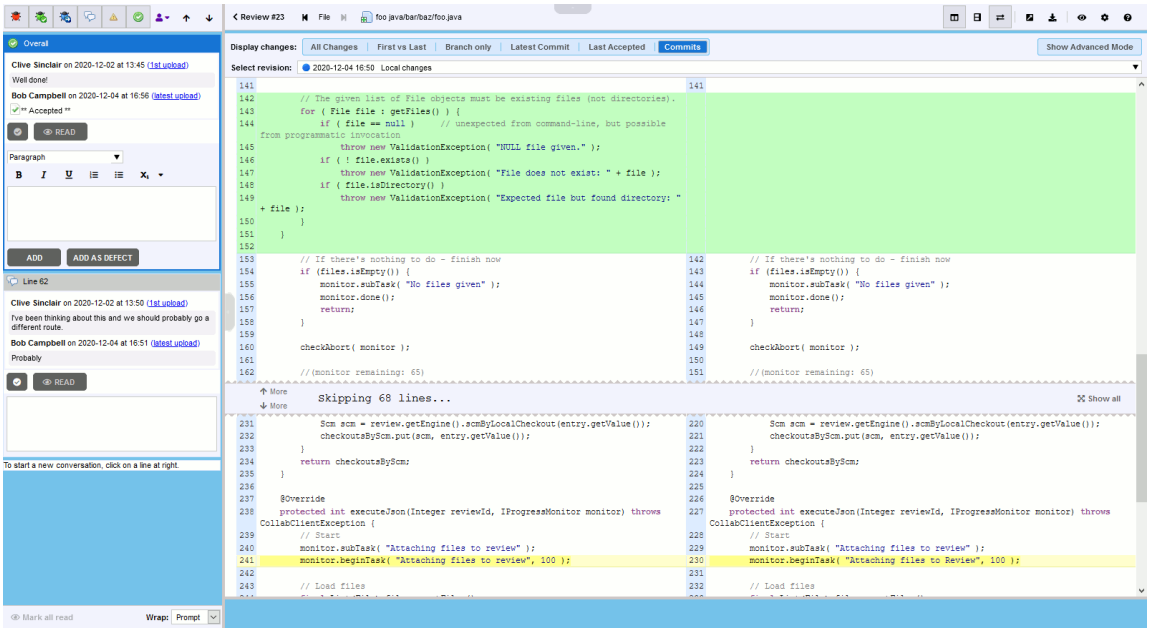
- Presentations 395 (.ppt, .pptx, .pot, .potm, .potx, .pps, .ppsm, .ppsx, .pptm and .odp)
- Visio vector graphics 400 (.vdw, .vdx, .vsd, .vsdm, .vsdx, .vss, .vssm, .vssx, .vst, .vstm, .vstx, .vsx and .vtx)
- Simulink models 404 (web view archives)
- Images 410 (.jpg, .jpeg, .png, .gif and .bmp)
- PDF documents 414 (.pdf, and any other document types converted to PDF format)
- URLs 419 (http and https)

**Important:** Document reviews (except for text-based and image files) is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

#### 4.3.4.1 Diff Viewer Overview

The Review Materials screen or Diff Viewer is where most of the reviewing takes place. It shows which exact changes were made to the review materials, how a file has changed from one revision to another.

Diff Viewer includes a number of configuration and display options that may change depending on the specific type of file that is being reviewed.




*The appearance of the Diff Viewer when reviewing source code files*

The Diff Viewer is divided into a number of sections:

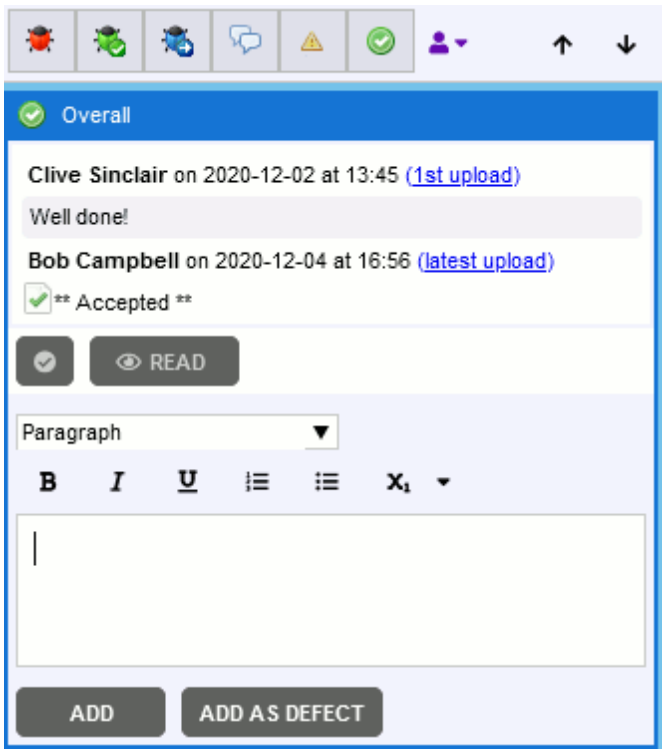
- Chat Pane and associated toolbar options 365
- Diff Viewer Toolbar 367 (Which includes the Display Options pane 372)
- Diff Viewer Header 369
- Diff Viewer Content (The actual content under review)

NOTE: The Diff Viewer opens in the same window and tab, when launched from the Review Summary screen. To open it in a new window, you can Shift+click the link.

## Chat Pane

On the left of the main Diff Viewer page, there is a pane for chat threads. This pane allows you to add both global and content-specific comments and defects to the document you are reviewing. **Comments** may hold any information concerning the review: a question, a clarification, a remark, an encouragement, whatever else. **Defects** indicate a problem that needs to be fixed. Comment and defect description could be in plain-text or use rich-text and Markdown formatting 447, they could also mention other Collaborator users 445. To learn more about global and content-specific comments and defects, see Types of Review Comments and Defects 421.

To hide or show the Chat pane, you can click the expander button in the pane border. When you click anywhere in the Diff Viewer content, the Chat pane automatically expands so that you can input your comment or see existing comment or defect. Similarly, it automatically expands when you click on a link to comment or defect (except for Overall comments and defects 424).




At the top of the chat pane, you will find a series of buttons that control whether or not certain types of comments appear in full or in collapsed states (to save viewing space). Note that defects, count as unread comments until they are accepted/marked read.

The first three icons control whether defects appear collapsed or expanded in the chat panel. The red bug controls the display for open defects, the green bug controls the display for fixed defects and the blue bug controls the display for defects tracked externally.

The next four icons control the display of non-defect conversations. The white conversation bubble controls the display for all comments. The yellow hazard sign controls the display for unread comments. The green checkmark controls the display for conversations that have been marked accepted. The pushpin icon controls whether to display markers for coordinate comments within word-processing documents, presentations, PDF documents, images and vector graphics. The user icon with a drop-down specifies whether to display comments and defects from all participants of just from the selected participant.

The next two icons allow navigating to the previous or next comment or defect within the current document.

Each comment and defect display who and when made them, as well as links to file revision where that comment or defect was submitted. For files from source-control systems, the version information link displays the commit ID, whereas for locally uploaded files it displays the upload number.

To learn how to make conversation during the review, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

## Diff Viewer Toolbar

The toolbar at the top of the Diff Viewer includes options for altering the display of information and navigation through documents within the review. The number of available toolbar items varies a little depending on the reviewed material.


*Toolbar appearance during code review*


*Toolbar appearance during document review*


*Toolbar appearance during Simulink model review*

**Back to review summary** Returns back to review summary page.

**File Navigation** Navigate to the previous or next file in the review, respectively.

**File Name** Displays the name of current file.

**Orientation** Specifies whether the Diff Viewer displays in side-by-side mode or over/under mode.

- When enabled, Before and After panes are displayed over/under one another.
- When disabled, Before and After panes are displayed side-by-side.

This button is displayed only for text-based formats and images.

**Display Order**

Determines which pane is considered the 'After' pane, for the purposes of highlighting additions and deletions.

- When enabled, the After pane is the leftmost panel in side-by-side mode and the uppermost panel in over/under mode
- When disabled, the After pane is the rightmost panel in side-by-side mode and the undermost panel in over/under mode

Toggles between adding coordinate comments and panning the file contents.

- When enabled, a single left click within a file contents will add a coordinate comment at specified position.
- When disabled, dragging the mouse pointer will pan the contents of the file.

This button is displayed only for word-processing documents, presentations, PDF documents, images and vector graphics.

Toggles between zooming and scrolling the file contents.

- When enabled, rotating mouse wheel will zoom the file contents in and out.
- When disabled, rotating mouse wheel will scroll the file contents up and down.

This button is displayed only for word-processing documents, presentations, PDF documents, images and vector graphics.

Locks mouse action for both panes.

- When enabled, both Before and After panes will react on mouse actions (panning, scrolling, zooming) simultaneously.
- When disabled, mouse actions (panning, scrolling, zooming) will affect only the pane a mouse hovers on.

This button is displayed only for word-processing documents, presentations, PDF documents, images and vector graphics.

**Zoom Level**

Specifies the zoom level for both Before and After panes. You can set the scale to various percentages, Page Width or Full Page.

This button is displayed only for word-processing documents, presentations, PDF documents, images and vector graphics.

**Search**

Performs a full-text search within the current document. Type-in the desired text and press Enter.

This button is displayed only for Word processing and PDF documents, presentations and vector graphics.

**External Diff Viewer**

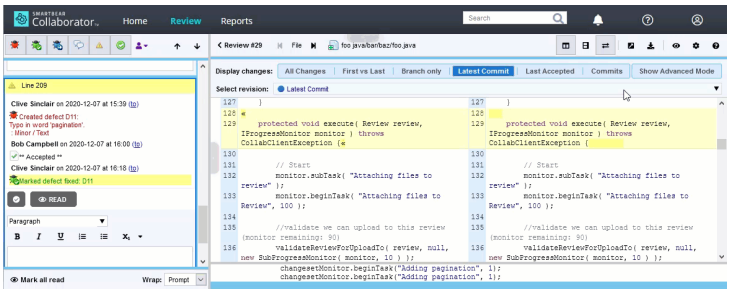
Launches the current document review in an external diff viewer.

**Download Diff**

Downloads a Diff file of the current document.

**Focus Mode On/ Off**

Expands or collapses the main toolbar and the Chat section, so that you can focus on review content.




*See how focus mode works*

**Display Options**

Opens the Display Options panel 372 which controls how Diff Viewer should display various data.

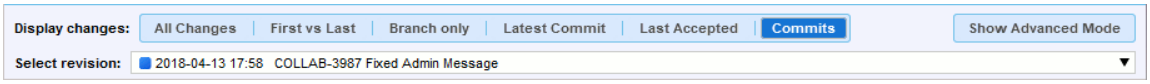
This button is displayed only for text-based formats, for documents and for Simulink models.

**Help**

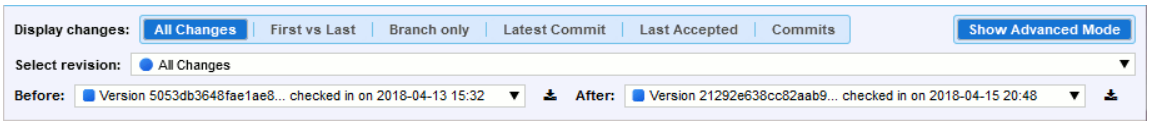
Opens the help menu with descriptions of color-code legend, keyboard shortcuts, and links to documentation.

## Diff Viewer Header

The content of the header varies depending on the type of file being reviewed, current revision selection mode, and content orientation.




*Diff Viewer header in simple revision selection mode*




*Diff Viewer header in expert revision selection mode*

**Display changes**

Allows to select which file revisions (if available) will be compared.

The available options are:

- *All Changes* - Compare the current revision of a file against its *base revision* - that is, a state of file before any changes related to current review have been made. For pre-commit reviews, base revision is the revision that you checked out from the repository. For post-commit reviews, base revision is the revision that precedes your commit.
- *First vs Last* - Compare the most recent revision of a file against its first revision that was uploaded during this review.
- *Branch only* - Compare the current revision of a file against its base revision excluding changes merged from other branches.
- *Latest Commit* - Compare the current revision of a file and its previous revision.
- *Last Accepted* - Compare the current revision of a file against its latest accepted revision. During the review, participants may accept some particular revisions of a file to denote that they agree with the changes. Uploading a further revision of that file clears the Accepted mark. If a participant has not accepted any revision yet, then compares the current revision of a file against its base revision.
- *Commits* - Compare the most recent revision against any arbitrary revision chosen in the **Select revision** drop-down.

The "Branch only" mode may produce slightly different output depending on whether the review was created manually or via repository integration. In reviews created manually it is not always possible to filter-out merge changes, so sometimes they still could be displayed. Reviews created via integration have broader access to the repository and thus can exclude merge changes more thoroughly. The drawback of the latter approach is that the contents of individual merge commits will be displayed incorrectly.

**Show Advanced Mode** — Toggles between simple and advanced modes of revision selection.

**Select revision** — Displays the current revision selection mode, or the file revision to be compared when the **Display changes** selector is set to *Commits*. Each file revision is identified according to the Revision Caption Pattern 326 setting and may include provider type, creation time, item identifier and description.

**After** — Visible in advanced revision selection mode.

Specifies the file revision to be displayed in the 'After' pane. Each file revision is identified by its SHA-1 hash value. Additionally, the file folder icon adjacent to this field can be used to download the selected revision of the file.

When you select the same file revision in 'Before' and 'After' drop-downs, Diff Viewer will unite the 'Before' and 'After' panes. In this case, added file revisions will have green background, while removed file revisions will have red background.

**Before** — Visible in advanced revision selection mode.

Specifies the file revision to be displayed in the 'Before' pane. Each file revision is identified by its SHA-1 hash value. Additionally, the file folder icon adjacent to this field can be used to download the selected revision of the file.

When you select the same file revision in 'Before' and 'After' drop-downs, Diff Viewer will unite the 'Before' and 'After' panes. In this case, added file revisions will have green background, while removed file revisions will have red background.

File revisions in Select revision, Before and After drop-down lists are further identified by an icon which appears adjacent to the revision number denoting:

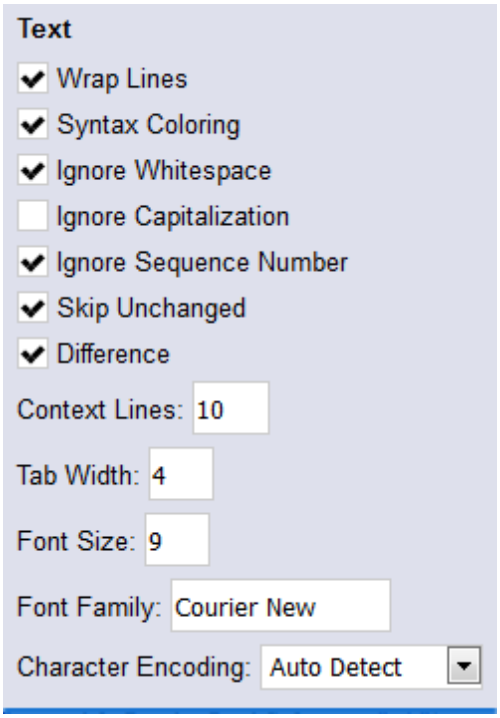
| | | |
|---|---|---|
| Icon shape: |  | Square shape indicates that file was uploaded from any source control system (within changelist, commit, diff and so on). |
| |  | Circle shape indicates that file was uploaded from local storage without any versioning. |
| Icon color: |  | Green color indicates that file was added. |
| |  | Blue color indicates means that file was modified. |
| |  | Red color indicates that file was removed. |
| Icon checkmark: |  | Unchecked indicates that file change was not accepted yet. |
| |  | Checked indicates that file change was accepted. |

## Display Options Panel

The Display Options panel contains various options that determine what data is displayed by the Diff Viewer and how it is displayed. This panel is available only for text-based formats, for spreadsheets, presentations, word processing documents and PDF documents and for Simulink models.

The list of panel option varies, depending on a format of the current file:

## Text-based formats




**Wrap Lines** — Controls whether lines of text are wrapped to the width of the pane.

NOTE: When 'Wrap Lines' is not selected, an option will appear at the bottom of the text section to enable/disable synchronized scrolling of the panes.

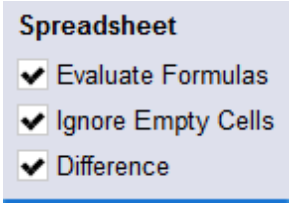
**Syntax Coloring** — Controls whether syntax-specific coloring is used, when supported.

**Ignore Whitespace** — Controls whether white-spaces are taken into account when showing differences.

This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports.

| | |
|---|---|
| **Ignore Capitalization** | Controls whether capitalization is taken into account when showing differences.<br><br>This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports. |
| **Ignore Sequence Number** | Controls whether COBOL sequence numbers are taken into account when showing differences.<br><br>This setting affects how line differences are displayed in Diff Viewer. Once enabled, line numbers and conversation position displayed in Diff Viewer may vary from line numbers and conversation position displayed in Review Summary Screen and reports. |
| **Skip Unchanged** | Determines whether large blocks of unchanged content is displayed. These gaps will appear, in the Diff Viewer as 'Skipping X lines...' and will include the two links labeled 'More', which will allow you to see more of this skipped content. |
| **Difference** | Enables highlighting of additions, deletions and differences in the viewer. |
| **Context Lines** | Defines the number of lines to show before and after the actual changes. So that you can better understand the context of changes performed. |
| **Tab Width** | Sets the number of spaces per tab. |
| **Font Size** | Allows customization of the font size. |
| **Font Family** | Allows customization of the font family |
| **Character Encoding** | Specifies the character encoding for the file. |

## Spreadsheets



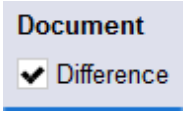

**Evaluate Formulas** — Specifies whether to evaluate formulas and display the resulting value, or to display the formula itself.

This setting affects the entire spreadsheet. To evaluate formulas for the selected cell, you can use the  button in the cell detail pane.

**Ignore Empty Cells** — Specifies whether to ignore addition and deletion of empty cells, or to highlight them.

**Difference** — Enables highlighting of additions, deletions and differences in the viewer.

## Word processing documents, PDF documents, presentations




**Difference** — Enables highlighting of additions, deletions and differences in the viewer.

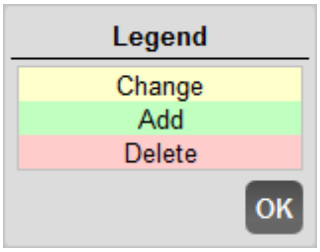
## Simulink models


**Show diff based on block properties** — Enables highlighting of additions, deletions and differences in block properties.

**Show diff based on image comparison** — Enables highlighting of additions, deletions and differences in graphical objects.

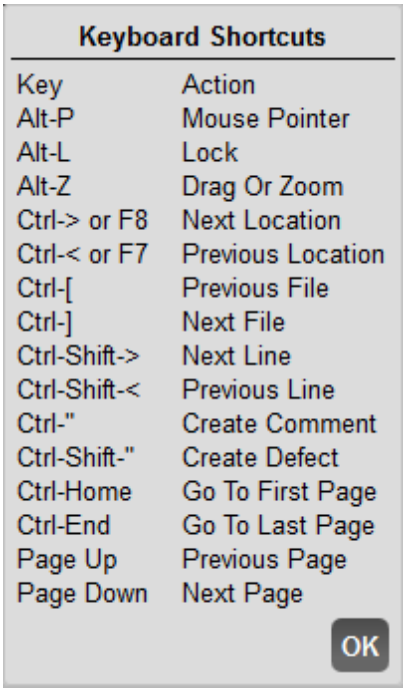
## Legends

The differences are noticeably highlighted and color coded as explained by the legend: Changes appear in a yellow, additions are green and deletions are red.




## Keyboard Shortcuts

The  Diff Viewer is completely navigable from the keyboard. You can move around the file, jump to changes, and make comments and defects. To get the list of keyboard shortcuts, click the "Help" link in the menu bar, then "Keyboard Shortcuts".




Shortcuts with ALT key are not supported on Apple Macintosh.

## Limitations

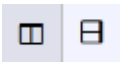
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

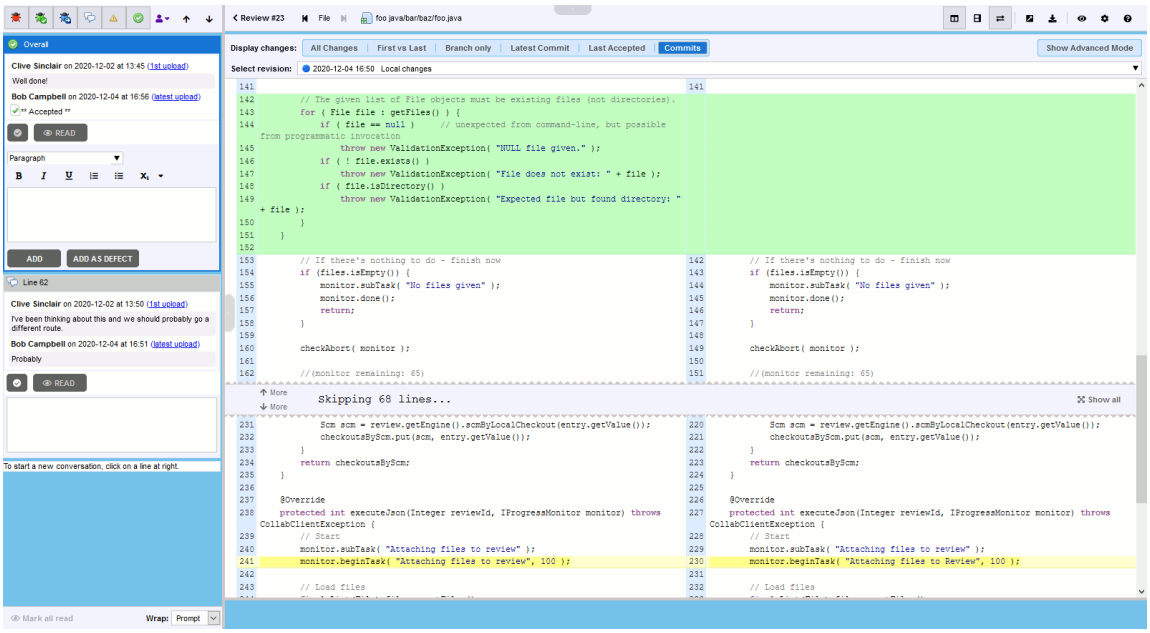
- **Subversion, Perforce, TFS, RTC, Git:** If you have "gaps" while adding subsequent atomic changelists to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- **Perforce, TFS, RTC:** If you add pending changelists from different workspaces to the same review.
- **Git:** If you add changelists from different branches or repositories to the same review.
- **RTC:** If you add changelists from different workspaces or streams to the same review.
- **All source control systems:** If you add several diffs (non atomic changelists) to the same review.

#### 4.3.4.2 Reviewing Source Code and Text Files

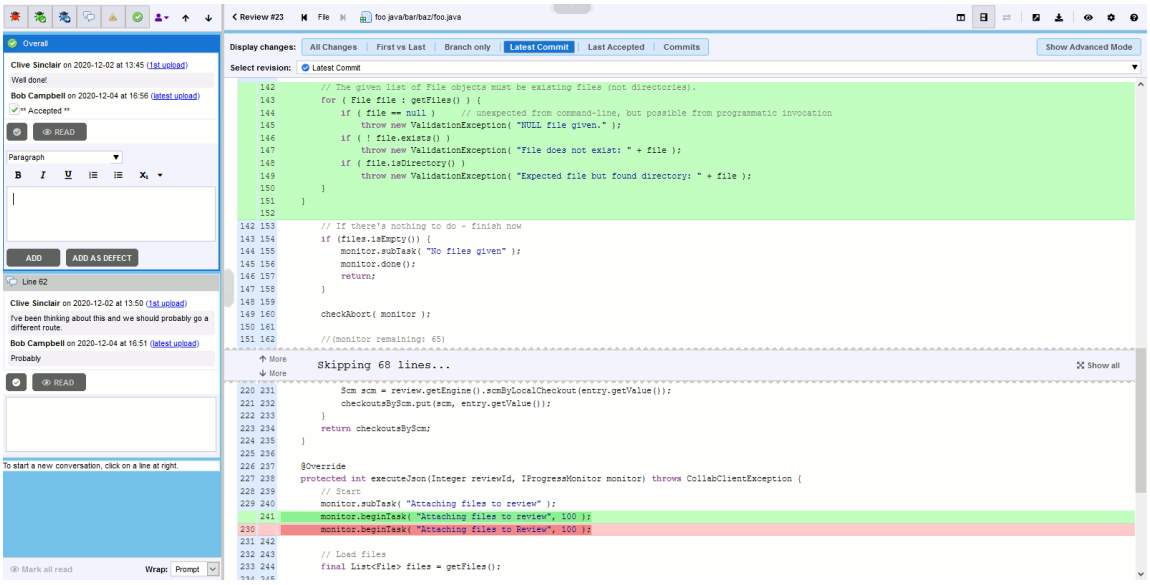
When reviewing source code and any other text-based formats, the review screen becomes the Diff Viewer. Here, you can see file content, differences with previous versions of the file, and all comments and defects for the file all on one screen.

## Viewing Differences

The Collaborator diff view displays any text either side-by-side or in an over-under orientation. Both views have all the same options and functionality, so users should feel free to use whatever is most comfortable. Since differences in the content could affect users' orientation preference, changing the orientation is simple, just click the  **Orientation** button on the toolbar.



*Standard View with Side-by-Side Orientation*



*Standard View with Over/Under Orientation*

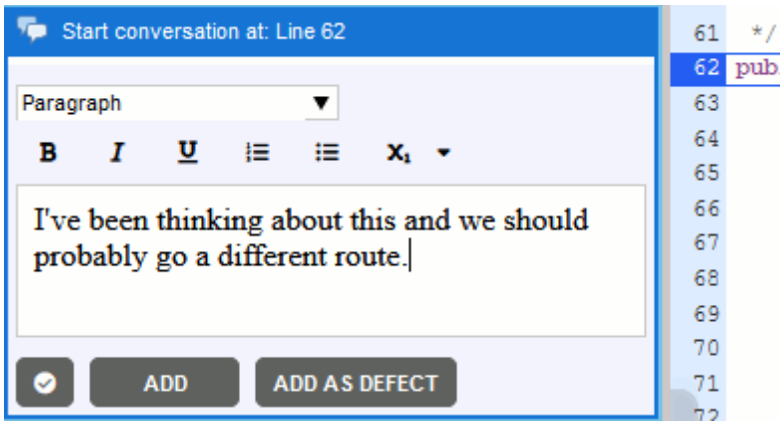
The over and under orientation merges both versions of the file together. Unchanged lines are displayed as one, while added and deleted lines are highlighted with the appropriate line number for the version.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing text files, you can create global[423], annotation[423], overall for file revision[424] and line[424] comments and defects.

Documents can have content that displays, for example, on line 2 of the original version and on line 4 of the reviewed version. The Diff Viewer recognizes these changes, promotes comments to correct places in the newer version and navigates the Before and After panes independently to make this content visible during the review.

To add line comments, click the desired line of text in the content view, type your comment in the text box and click **Add**. To add defects, click the desired line of text in the content view, type the defect description, click **Add as defect** and fill-in the required fields. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].



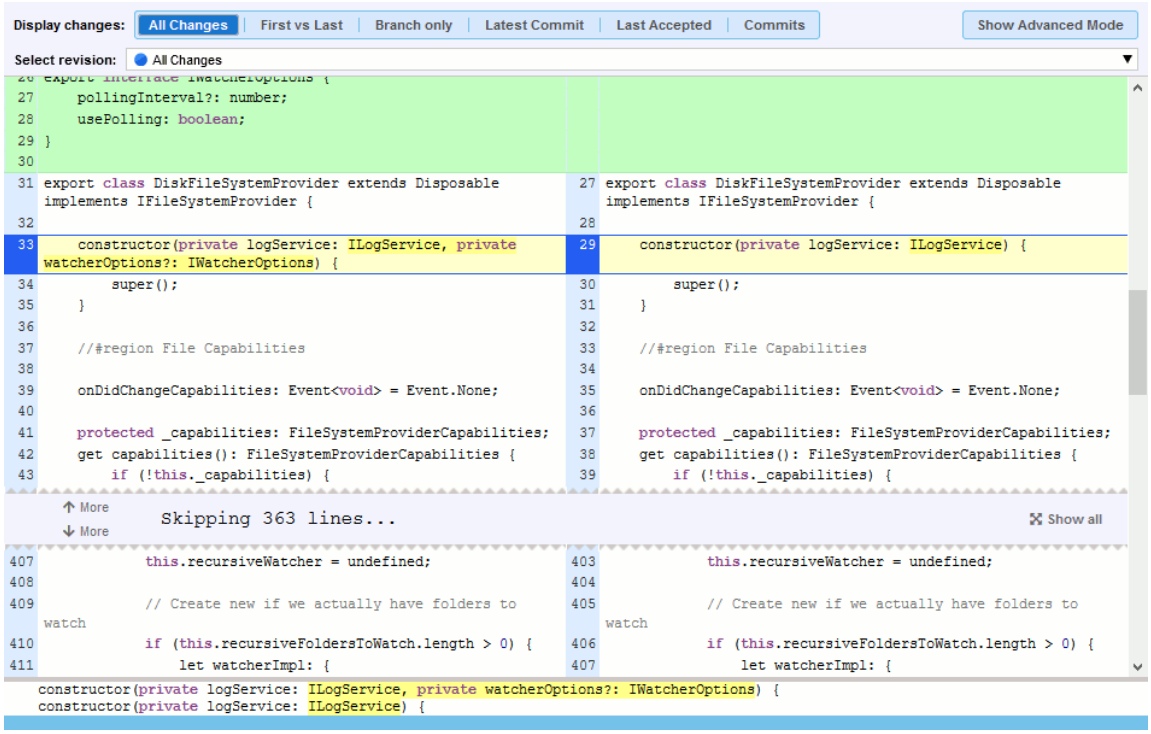

To learn more about communicating during the reviews, see Types of Review Comments and Defects[421] and Review Chats, Comments and Defects[429] topics.

## Menubar

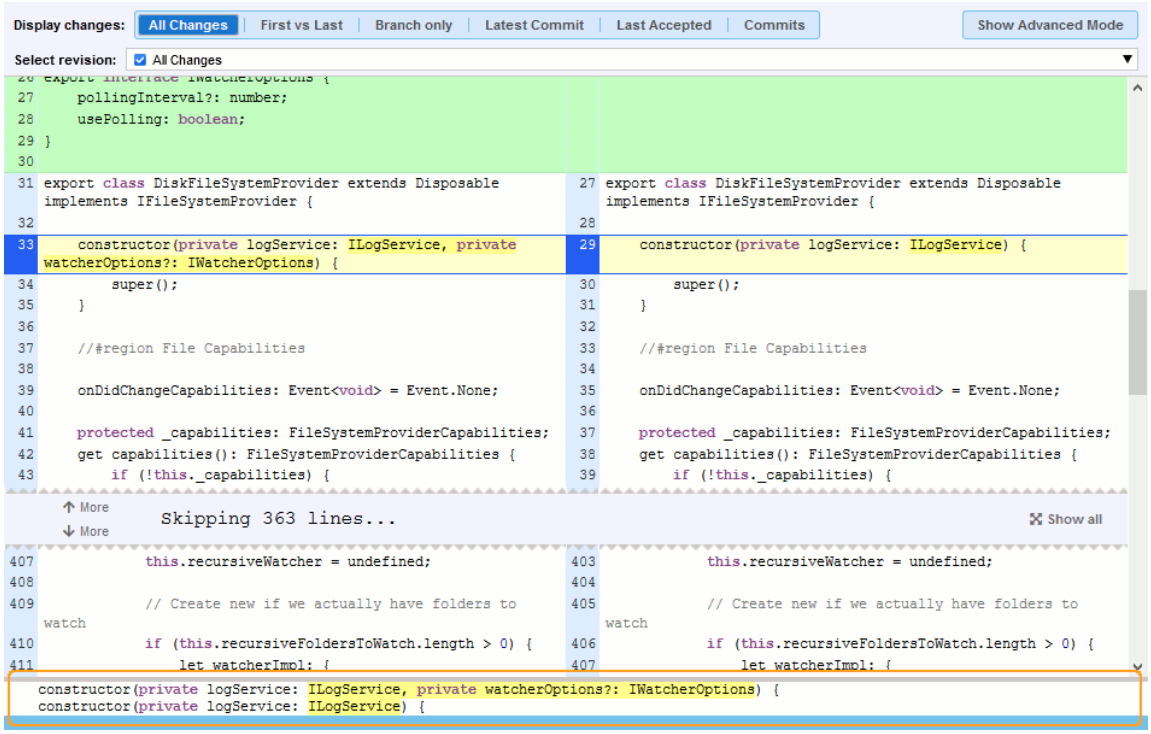
Right at the top, above the main file view, a menubar is displayed with a few possible actions. Buttons with blue triangles to the right will display drop-down menus when clicked on. For a list of all of the features of the menubar, please review the Diff Viewer Overview[364] section.

## The File Content View

To the right of the chat session, there is the main file view. This is where you can view file content and differences. The format of the main file on your local Collaborator screen may be different depending on how you have configured the Display options[372] for the Diff Viewer.

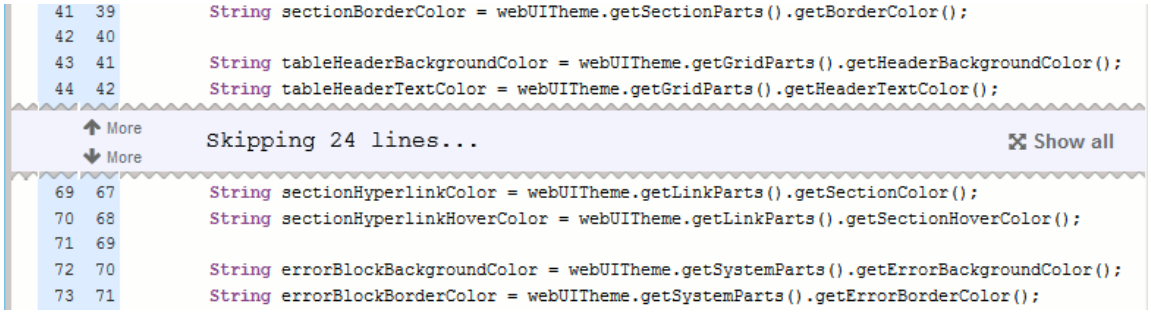



If differences are being shown, the interface automatically scrolls to the first difference (unless you got to this screen by clicking on a particular line number, in which case the screen will be centered on that line number). Below the file content is a small area which shows the currently selected line in an over-under view. This is most useful when the content view is in side-by-side mode, but the differences are not immediately clear in that view.




## Skip Unchanged

If the "Skip Unchanged[372]" option is enabled, unchanged lines will be hidden in the Diff Viewer as shown below:




If you would like to get more lines of context, just click either "More" link. The top link will show the next 25 lines of context, while the bottom link will show the last 25 lines of context.

## Syntax Coloring

When the "Syntax Coloring[372]" option is enabled, Collaborator attempts to determine a computer language of the current file and apply an appropriate syntax highlighting to it.

Collaborator has built-in support of syntax highlighting for the following computer languages:

- Ada
- ASP.NET
- Assembler
- C#
- C++
- C
- Cobol
- CSS
- Delphi
- Gosu
- HTML
- Java
- JavaScript
- JavaServer Pages (JSP)
- Kotlin
- Objective-C
- Perl
- PHP
- Python
- Ruby
- SGML
- Shell Script
- SQL Script
- Swift
- TCL-based
- Typescript

- Verilog
- VHDL
- Visual Basic
- XML

Syntax highlighting is fully configurable. Your Collaborator administrators can modify existing syntax highlighting schemas 304 or create new schemas to add highlighting for any other computer languages.

Moreover, if a DiffViewer displays a file whose type does not match any of the existing syntax highlighting schemas, Collaborator will display a prompt to create a new custom syntax highlighting schema 304 for this file type.

## Treat arbitrary file as text-based

Diff Viewer detects type of review materials 363 based on the extension of the uploaded files. For example, `.java` files typically stand for Java source code, `.rtf` and `.doc` are typically word-processing documents and so on. However, some extensions could stand for multiple types of data and this could mislead the Diff Viewer. For instance, the `.pot` extension could be either a PowerPoint template file or a portable object file.

In this case, you can use the **Text file types** 198 admin setting to specify explicitly which file types should be treated as text-based files.

### 4.3.4.3 Reviewing Word Processing Documents

Collaborator supports reviewing word processing documents of the following formats:

- Microsoft Word 97 and later (.doc, .docx, .docm, .dot, .dotm, .dotx),
- Rich Text (.rtf)
- OpenDocument (.odt and .ott)

**Note:** *Reviewing word processing documents is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3 *.*

**Important:** Document reviews is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

## Uploading Documents

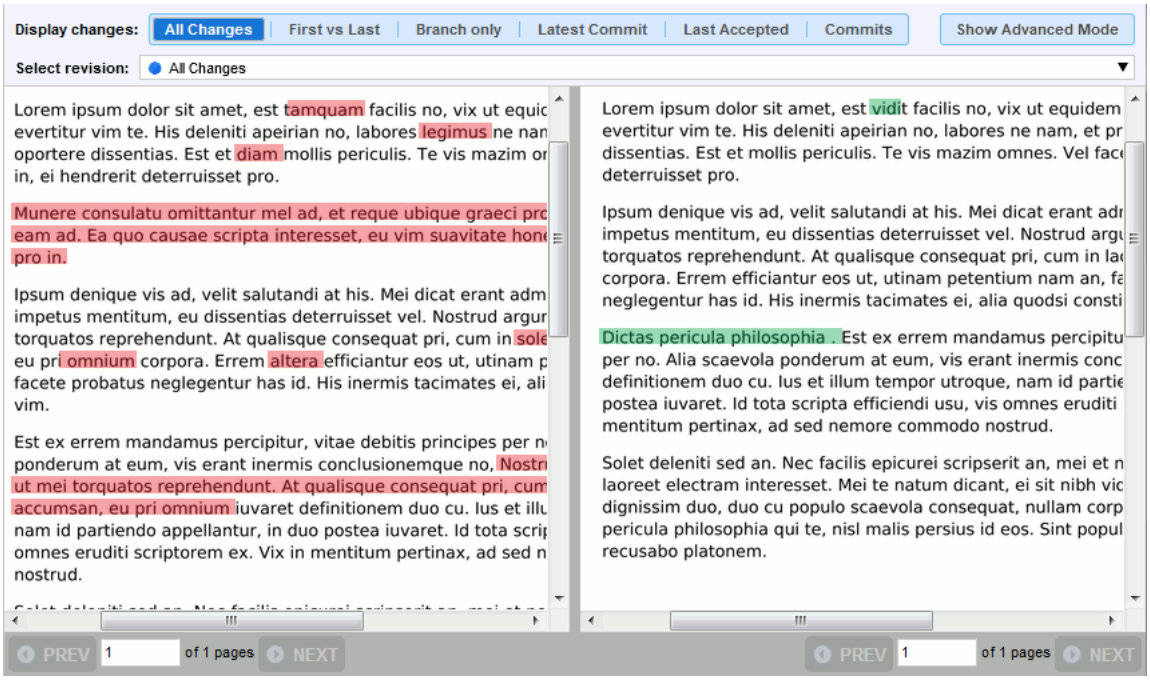
To review word processing documents, just attach 343 them to the review as you would any other file and when you open the Diff Viewer, the content area will display the document contents for review.

Alternatively, you can install SmartBear Collaborator for Word[605] plug-in and upload your documents directly from Microsoft Word.

**Note:** If you have the Track Changes setting enabled in your Word document, you have to accept the changes before uploading the document to Collaborator. Otherwise, Collaborator will treat the deleted text as if it were never deleted.

## Viewing Differences

Collaborator automatically finds and highlights differences between the document revisions that you uploaded and selected for review. Both revisions uses standard highlighting: red for deletions and green for additions.




- If a document contains tables, they are compared row by row. For more refined comparison of table data consider exporting table as spreadsheet and compare it separately[389].
- Since Collaborator converts word processing documents to the specific document format, it is impossible to use it to find diffs in some kinds of complex content, including these:
  - macros,
  - automatically generated formulas,
  - codes used for value calculation,
  - embedded objects,

- built-in Word comments,
- and so on.

Scrollbars are available for each revision of the document. You are also able to move the separator to show more or less of either revision. To do this, hover the mouse over the separator until the arrows appear, and click and drag until it reaches the desired location.

## Changing Pages

To change pages, use the arrows at the bottom left of the comparison screen.


The arrows will allow you to go to the **Previous Page** or the **Next Page**. You can also manually enter a **page number** and hit enter to jump to any arbitrary page. The page numbers will refer to the latest upload of the document shown.

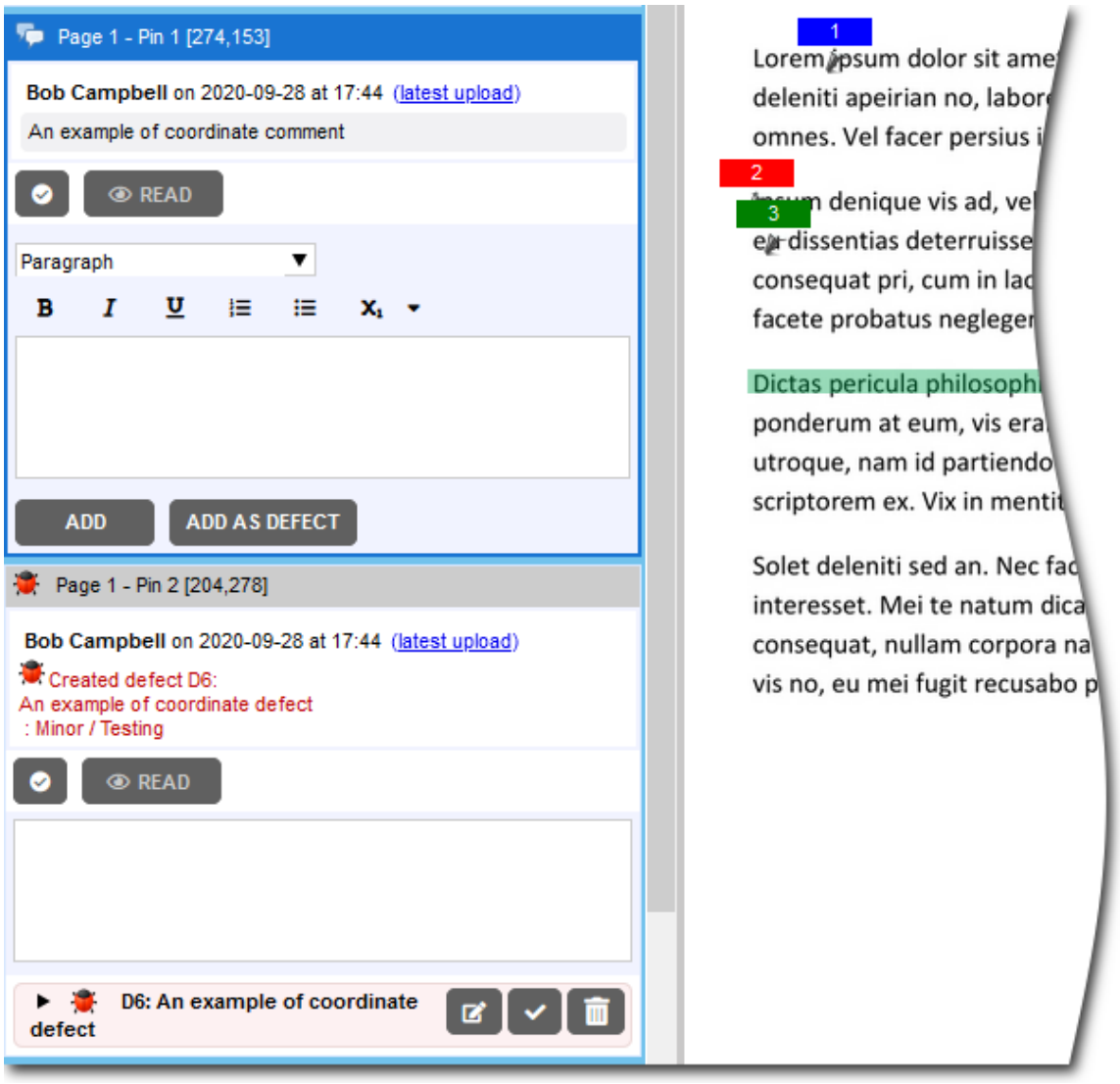
You can also use the mouse wheel to scroll through the pages (unless disabled by the respective option[325]).

Page navigation is disabled when the document is loading or when it is converting to Collaborator's internal format. In that case, Collaborator displays progress indicators near the navigation buttons.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing word processing documents, you can create global[423], annotation[423], overall for file revision[424], coordinate (pushpin)[425] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on specific text in a document, simply click the document at a point where you would like your comment to appear and begin typing. Collaborator will insert a pushpin (  ) to indicate the comment. The number in the pushpin head corresponds to the comment's order within the current page of the document.




You can hide pushpins by clicking the pushpin toggle button  on the chat toolbar. Use this feature if a pushpin overlaps an important part of the document.

Alternatively, you may specify an arbitrary label, such as "Section 5.1", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in a text document resulting in the pushpins not being adjacent to the corresponding text. Label text may not be updated.

Except the fact that documents are highlighted with pushpins and text files are highlighted by line, the way you use review chats 429 when reviewing Word documents is very similar to the way you use them when reviewing text files.

To learn more about communicating during the reviews, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

It's obvious that document content will change from one revision to another. For example, a fragment could move from top of the page to its bottom, or even to another page. The Diff Viewer attempts to recognize these changes and promote comments to correct places in the newer version. Besides it navigates the Before and After panes independently to make this content visible during the review. Yet in some cases, for example when multiple pages were inserted into the document, Diff Viewer may promote comment positions incorrectly. In this case you could move comments manually, as described below.

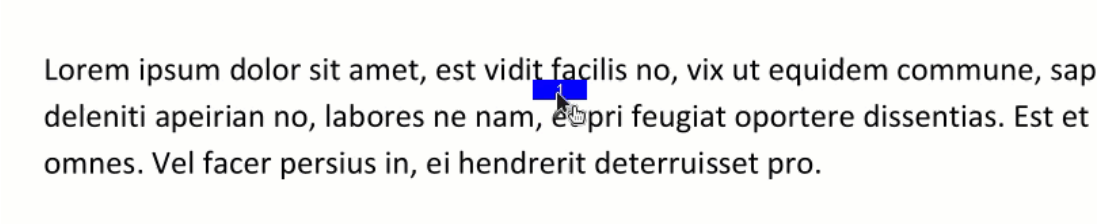
## Moving Comment Pushpins

You can move an existing pushpin to a new position in a document. The feature should be enabled by administrator 185 for all participants or for review creators only.

There are two ways to move a comment:

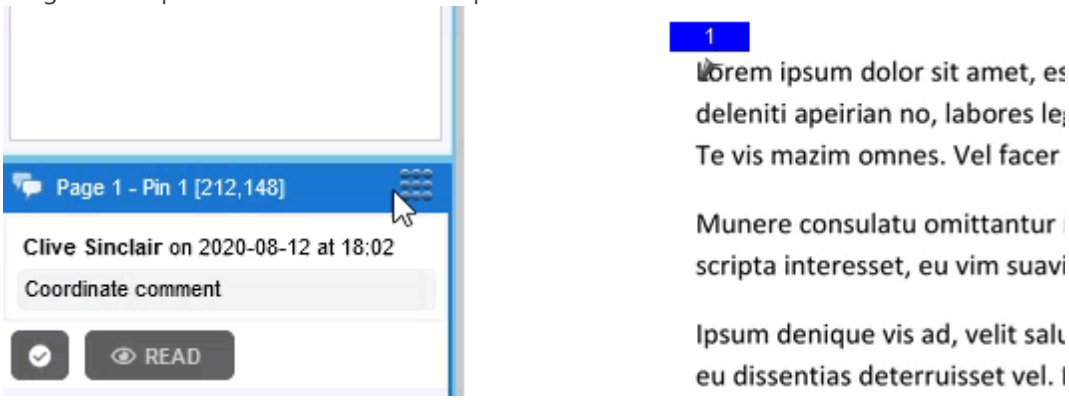
### via Drag and Drop

1. Drag the pushpin mark and drop it over the new position in the document.




### via Move Comment handle

1. Select the desired comment.
2. Press and hold the Move Comment handle ( ).
3. Drag and drop the handle over the new position in the document.



## Downloading Original Files

Collaborator stores documents with the review. It converts them to PDF to render them in the Diff Viewer 364. You can download the original files from the Review Materials section of Review Summary Page, or in Advanced revision selection mode of Diff Viewer by clicking the Download File button next to revision selector:


## Document Review Notes

- If you have the Track Changes setting enabled in your Word document, you have to accept the changes before uploading the document to Collaborator. Otherwise, Collaborator will treat the deleted text as if it were never deleted.
- After you have uploaded a Word document, Collaborator starts converting it to a specific format that it uses to find differences. This takes some time (several minutes for large documents). During the conversion, you can open the document in the Diff Viewer. However, you are not allowed to make comments, browse pages and see differences until the conversion is over. Once it finishes, Collaborator enables the page navigation controls and highlights the diffs.
- Due to conversion procedure, the formatting and appearance of some fragments may alter from the original document. Typically these are auxiliary parts of a document, like headers, footers, footnotes and similar.
- To keep the computer performance reasonable Collaborator limits the number of documents it will convert concurrently to four by default. If you upload a document while the conversion pool is full, the document will wait until there is room for it. This can take several minutes.
- If you experience server performance degradation, try configuring Java VM memory settings 1241. We recommend running the Collaborator server on a computer that has at least 4 CPUs (or cores) and has more than 4GB of memory allocated for the Java virtual machine. The memory requirement may increase depending on the typical document size your users upload. If you continue experiencing problems or need further assistance, please contact our Support Team 32.

- If the DiffViewer has troubles displaying small characters, diagrams or documents look blurry, you will need to increase the resolution scale via the `com.smartbear.diff.image.resolution.scale` 1232 Java VM option, restart Collaborator server and clear browser cache.

#### 4.3.4.4 Reviewing Spreadsheets

Collaborator supports reviewing spreadsheets of the following formats:

- Microsoft Excel 95 and later (.xls, .xlsx, .xlsb, .xlsm, .xltm, .xltx),
- OpenDocument Spreadsheet (.ods)

**Note:** *Reviewing spreadsheets is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

**Important:** Document reviews is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

## Upload Spreadsheets

To review spreadsheets, just attach 343 them to the review as you would any other file and when you open the Diff Viewer, the content area will display the spreadsheets contents for review.

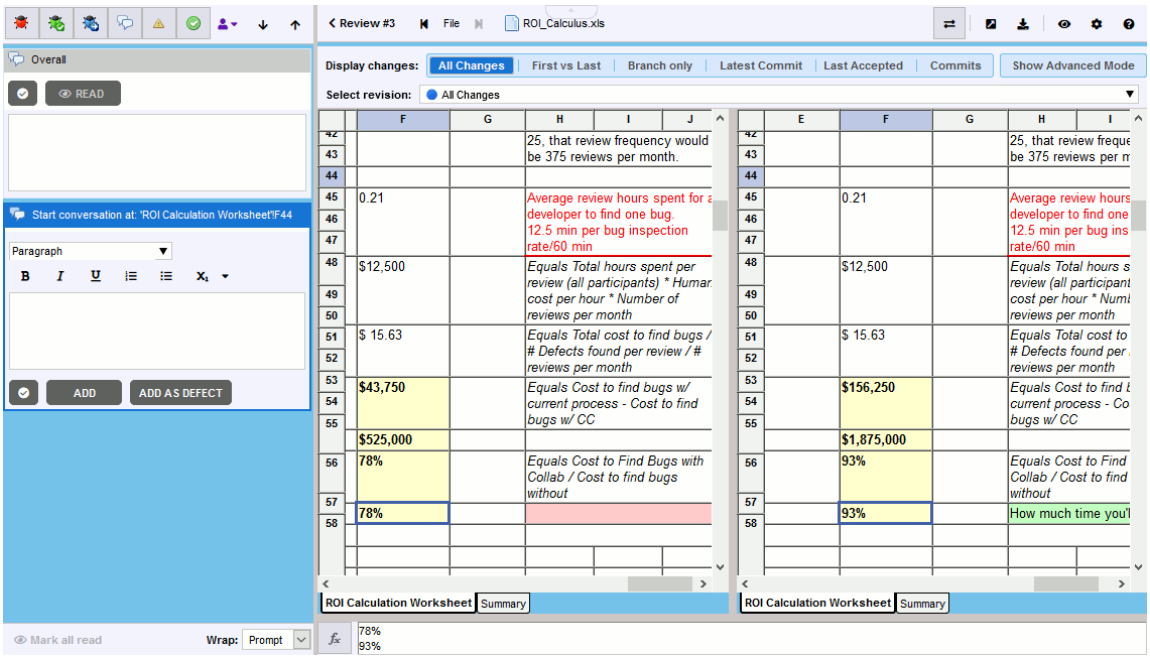
Alternatively, you can install SmartBear Collaborator for Excel 605 plug-in and upload your spreadsheets directly from Microsoft Excel.

**Notes:**

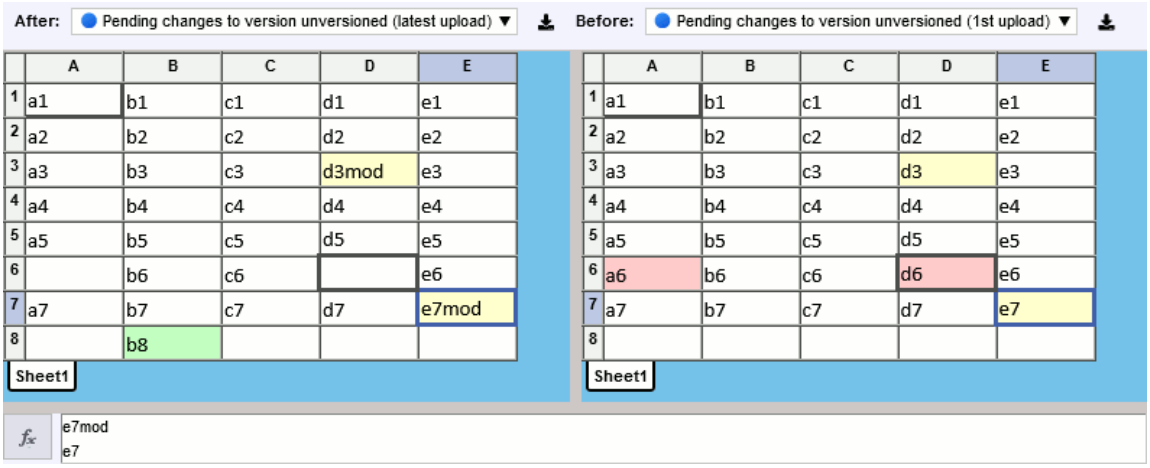
- When a file is uploaded for review, hidden rows are shown.
- You may need to save your file with word wrap enabled to more easily see the content during reviews.
- Do not rename spreadsheets between uploading revisions. If a spreadsheet's name was changed, Collaborator cannot map them. As a result, it will display cell location's values as deleted, and will show the "Sheet not present in this version" warning.

## Viewing Differences

The UI displays a diff of two versions of a spreadsheet side-by-side. There is a panel at the bottom that shows the contents of the selected cell in both versions of the spreadsheet on top of one another for easy visual comparison. When you click in a cell, the value is displayed in the cell detail pane (equivalent to the formula bar in Excel). The values for the selected cell appear in the cell detail pane with the value for the "after" version on top:




The use of color allows you to quickly identify changes in diffs so that you can address just what has been modified, added or deleted between spreadsheet revisions.

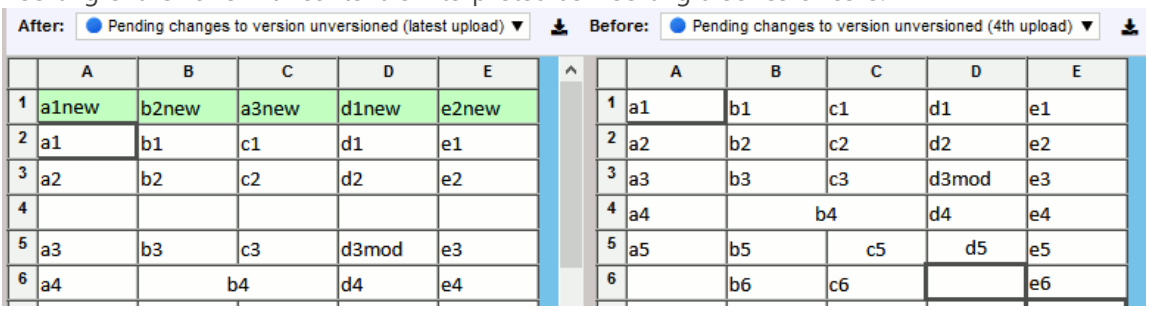



*Highlighting cell differences*

- When cell content was added it is highlighted with a green background in current revision panes.
- When cell content was modified it is highlighted with a yellow background both in current and previous revisions panes.
- When cell content was removed it is highlighted with a red background in previous revisions pane.
- All other cells (empty or not) display with a white background.

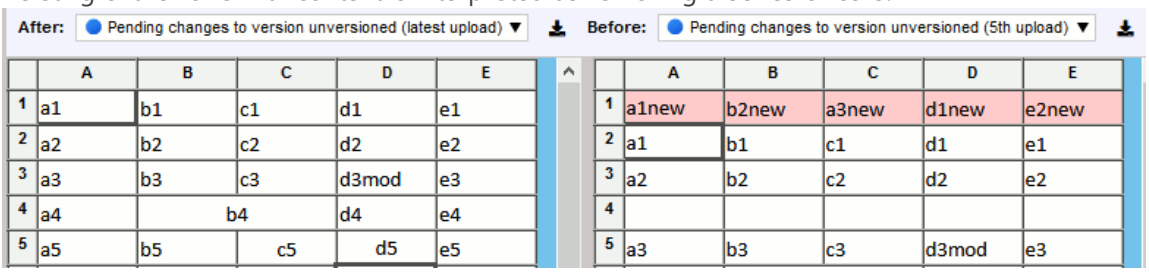
- Empty cells (cells without content) could be highlighted when the **Ignore Empty Cells** option of the Display panel 369 is disabled. In this case they would have the same color as added or deleted cells.

- Inserting entire rows with content is interpreted as inserting a series of cells.



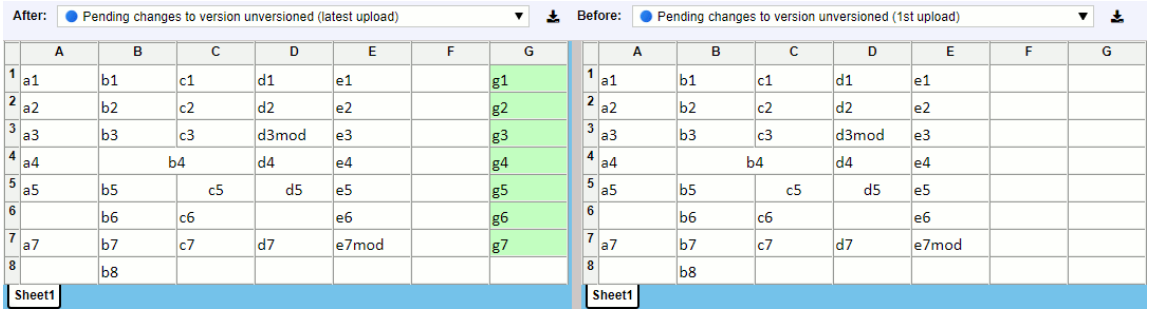

*Adding rows*

- Deleting entire rows with content is interpreted as removing a series of cells.




*Deleting rows*

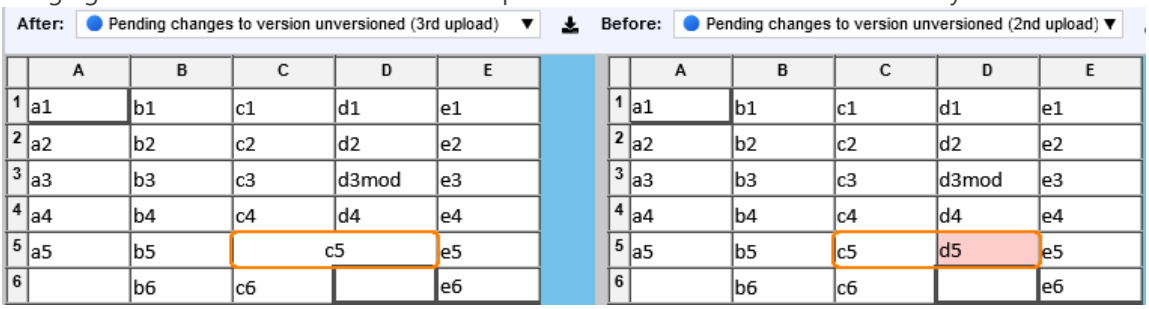
- Inserting entire columns with content could be interpreted either as inserting a new series of cells (just inserting new columns) or as modifying a series of cells (inserting columns and modifying nearby cells).




*Adding columns*

- Deleting entire columns with content could be interpreted either as deleting a series of cells (just removing columns) or as modifying a series of cells (deleting columns and modifying nearby cells).

- Merging two or more cells into one is interpreted as removal of content of nearby cells.




*Merging cells*

- Respectively, un-merging a cell into two or more cells is interpreted as adding content of nearby cells.

*Un-merging cells*

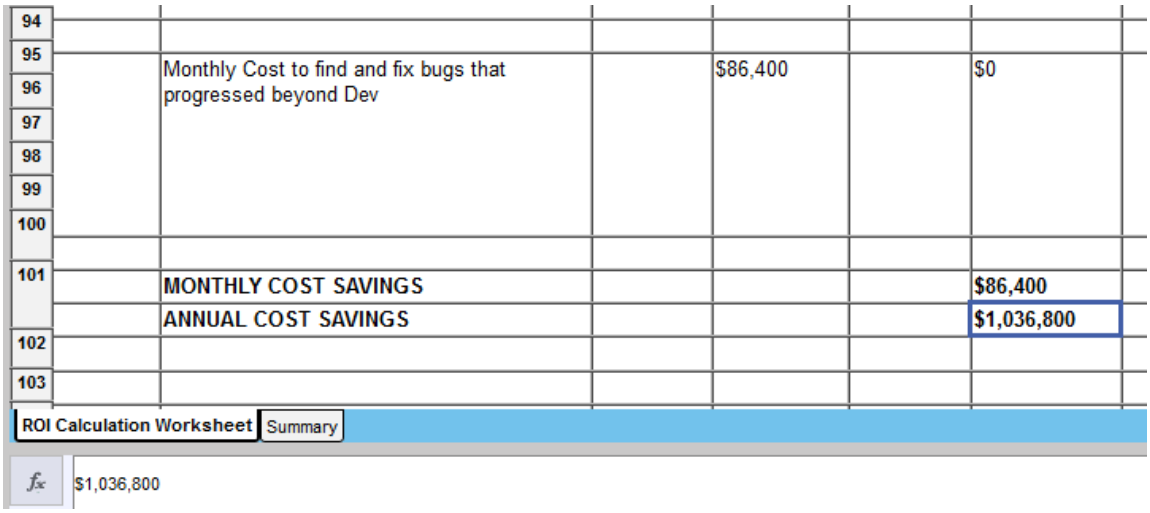
Since it is difficult to compare spreadsheet revisions to meet all use cases, some changes may not be highlighted in diffs in the way that people would expect. We will continue to work on the spreadsheet diff feature to meet the broadest set of needs.

**Notes:**

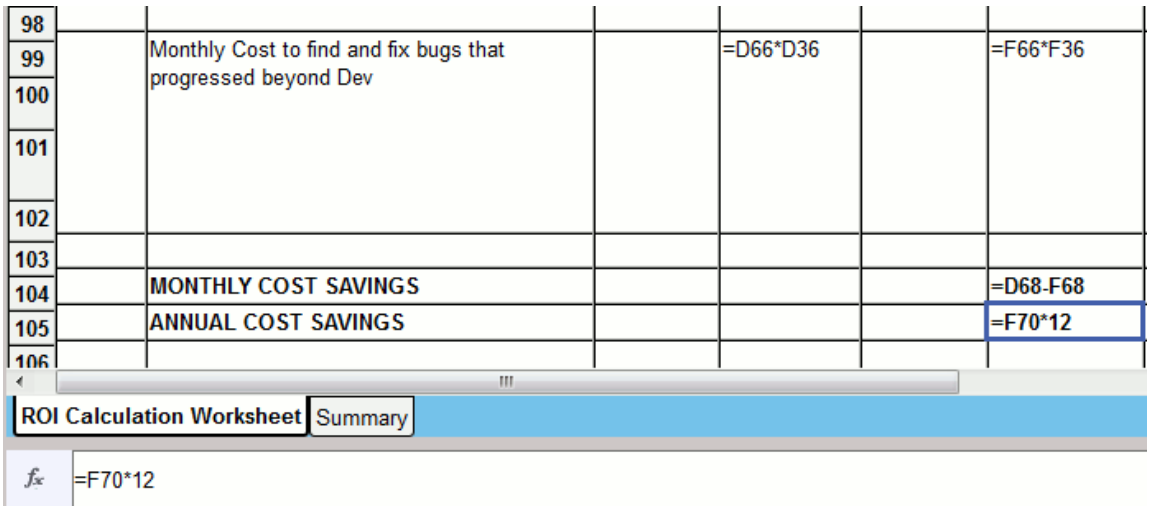
- The specific document format used by Collaborator makes it impossible to display diffs in complex objects, such as diagrams, charts and embedded images. If you are using those and want to see their differences, consider converting the spreadsheet to PDF format[414].
- DiffViewer displays table borders. Yet, some borders, for example “dot-dash” borders, or too thin borders could not be displayed properly because of html-rendering limitations.
- Some formatting options are ignored by DiffViewer.
- If the column widths of spreadsheets are not sized to show the entire text or set to wrap text, the numbering on the rows in Diff Viewer may not line up perfectly.
- Worksheet is often abbreviated as "sheet". As in Excel, the UI displays a list of tabs at the bottom of the workbook for switching between worksheets. Only one worksheet is displayed at a time.

## Reviewing Formulas

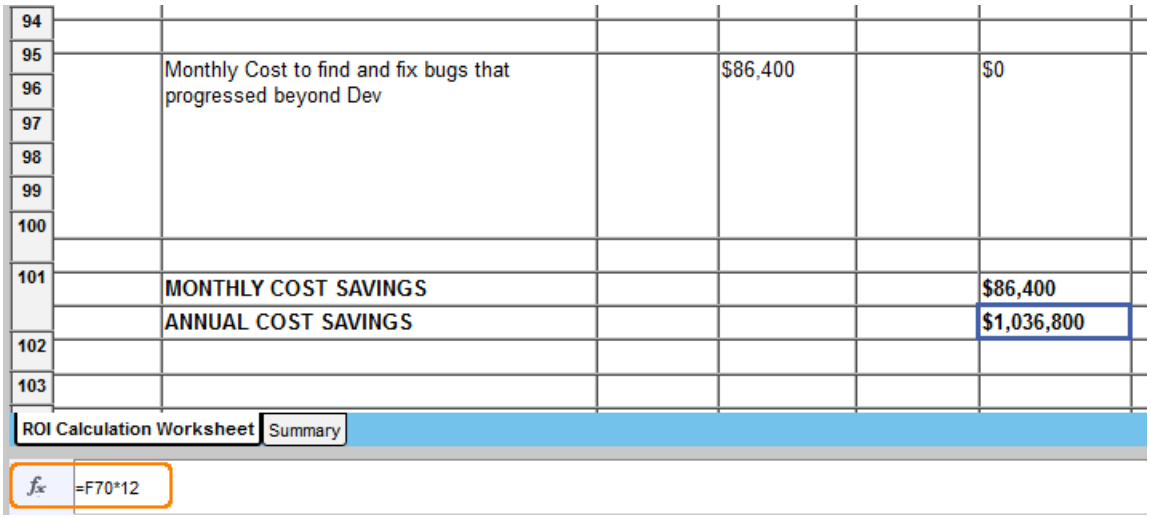
By default, Collaborator evaluates formulas and displays the resulting values.




To review formulas on the entire spreadsheet, disable the **Evaluate Formulas** option of the Display panel 369. In this case, all spreadsheet cells will display their formulas (if any) instead of the resulting value.




To review formula of some particular cell, select the cell and toggle the  button in the cell detail pane.




## Display Options

The Display panel[369] has specific options for spreadsheet review.

- When the **Evaluate Formulas** option is enabled spreadsheets will only show the evaluated results of the formulas, not the formulas itself. This setting affects the entire spreadsheet.

  To evaluate formulas for the selected cell, you can use the  button in the cell detail pane.

- When the **Ignore Empty Cells** option is enabled Diff Viewer does not highlight addition and deletion of empty cells.

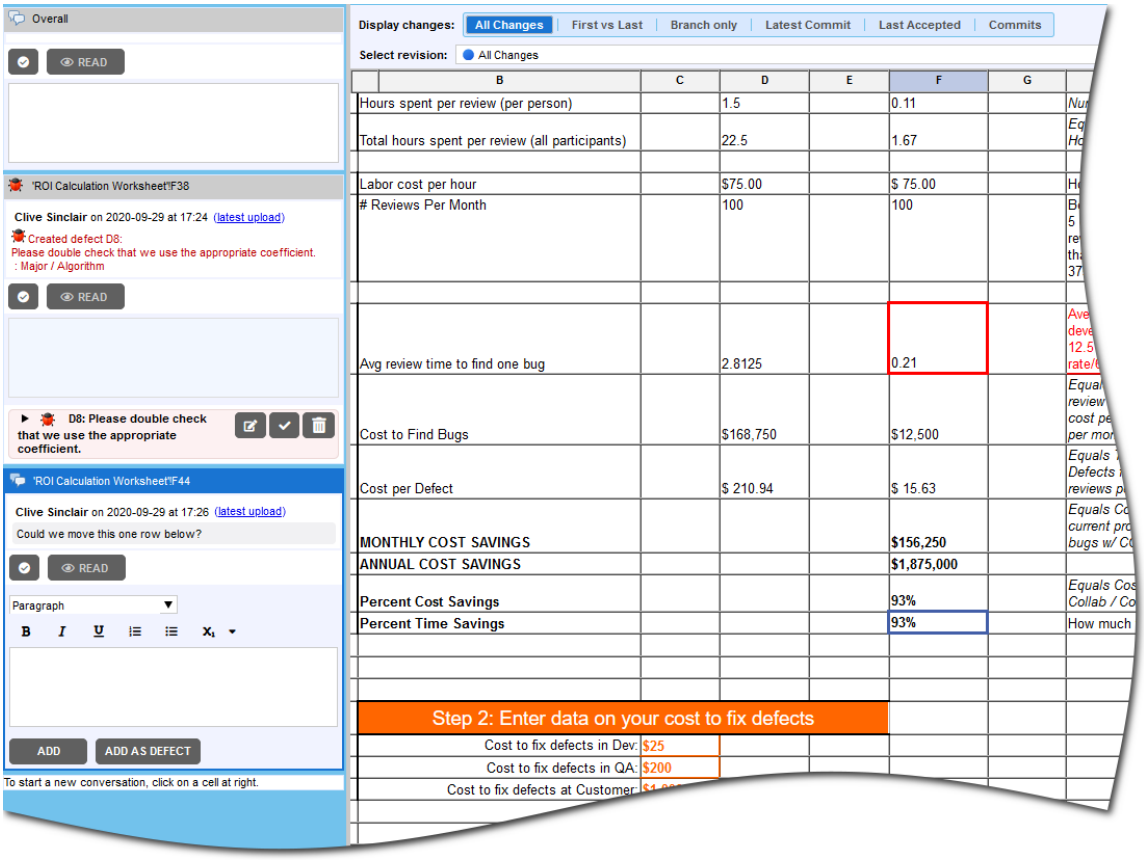
## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing spreadsheets, you can create global[423], annotation[423], overall for file revision[424] and cell[427] comments and defects.

To comment on specific cell within a spreadsheet, click the desired cell in the content view, type your comment in the text box and click **Add**. To add defects, click the desired cell in the content view, type the defect description, click **Add as defect** and fill-in the required fields. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

Conversations on cells are displayed in alphabetical order by sheet name, not in the sheet order found in the version content. In the example above, conversations for "ROI Calculation Worksheet" would come before "Summary" work sheet. The name of the chat includes the name of the sheet followed by the location of the cell being discussed. For example, Summary!C10.

Spreadsheets can have content that displays, for example, in cell E5 of the original version and in cell G7 of the reviewed version. The Diff Viewer recognizes these changes, promotes comments to correct places in the newer version and navigates the Before and After panes independently to make this content visible during the review.




To learn more about communicating during the reviews, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

### 4.3.4.5 Reviewing Presentations

Collaborator supports reviewing presentations of the following formats:

- Microsoft PowerPoint 97 and later (.ppt, .pptx, .pot, .potm, .potx, .pps, .ppsm, .ppsx, .pptm),
- OpenDocument Presentation (.ods)

**Note:** *Reviewing presentations is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

**Important:** Document reviews is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

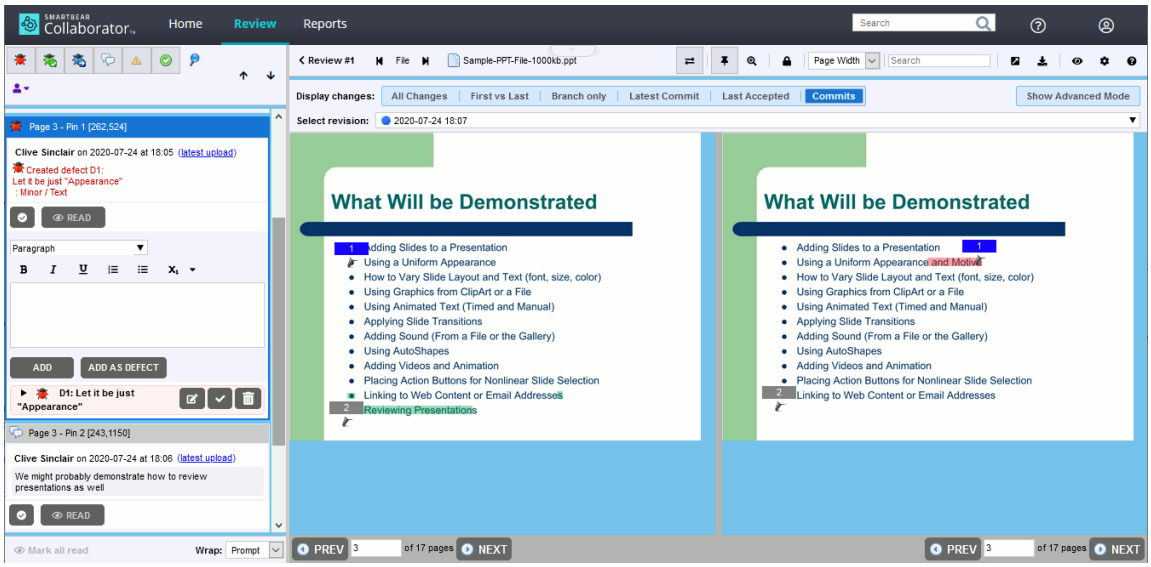
## Uploading Presentations

To review presentations, just attach[343] them to the review as you would any other file and when you open the Diff Viewer, the content area will display the presentation contents for review.

Alternatively, you can install SmartBear Collaborator for PowerPoint[605] plug-in and upload your presentations directly from Microsoft PowerPoint.

## Viewing Differences

Collaborator automatically finds and highlights differences between the document revisions that you uploaded and selected for review. Both revisions uses standard highlighting: red for deletions and green for additions. Presentations can only be viewed in a **single** or **side-by-side** layout.

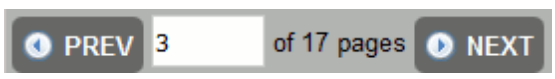



*Reviewing Presentations*

Scrollbars are available for each revision of the document. You are also able to move the separator to show more or less of either revision. To do this, hover the mouse over the separator until the arrows appear, and click and drag until it reaches the desired location.

## Changing Slides

To change presentation slides or pages, use the arrows at the bottom left of the comparison screen.



The arrows will allow you to go to the **Previous Page** or the **Next Page**. You can also manually enter a **page number** and hit enter to jump to any arbitrary page. The page numbers will refer to the latest upload of the document shown.

You can also use the mouse wheel to scroll through the pages (unless disabled by the respective option[325]).

Page navigation is disabled when the document is loading or when it is converting to Collaborator's internal format. In that case, Collaborator displays progress indicators near the navigation buttons.

## Zooming In And Out

Both the side-by-side and single view support zooming[368] the images in the display. Zooming out is useful if the page is too large to display in the browser window without scrolling. Zoom is also useful for more precise placement of the comment markers. However, whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

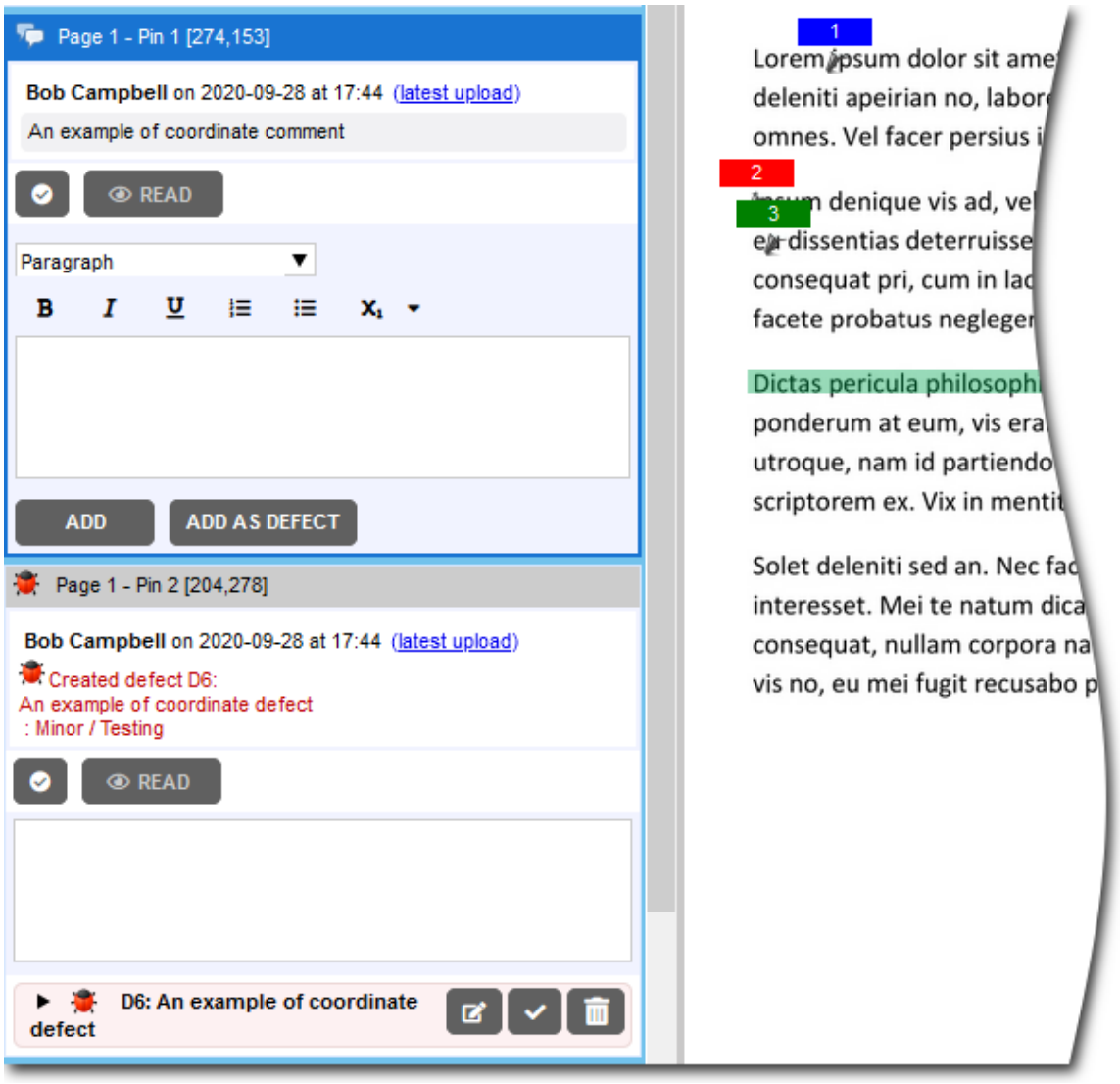
Rescaling of documents is performed by rerendering the document on the Collaborator server and could be slow depending on the size of the document and the selected image scale.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing presentations, you can create global[423], annotation[423], overall for file revision[424], coordinate (pushpin)[425] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on specific text in a document, simply click the document at a point where you would like your comment to appear and begin typing. Collaborator will insert a pushpin ( 

 ) to indicate the comment. The number in the pushpin head corresponds to the comment's order within the current page of the document.




You can hide pushpins by clicking the pushpin toggle button on the chat toolbar. Use this feature if a pushpin overlaps an important part of the document.

Alternatively, you may specify an arbitrary label, such as "Section 5.1", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in a text document resulting in the pushpins not being adjacent to the corresponding text. Label text may not be updated.

To learn more about communicating during the reviews, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

It's obvious that document content will change from one revision to another. For example, a fragment could move from top of the slide to its bottom, or even to another slide. The Diff Viewer attempts to recognize these changes and promote comments to correct places in the newer version. Besides it navigates the Before and After panes independently to make this content visible during the review. Yet in some cases, for example when multiple pages were inserted into the document, Diff Viewer may promote comment positions incorrectly. In this case you could move comments manually, as described below.

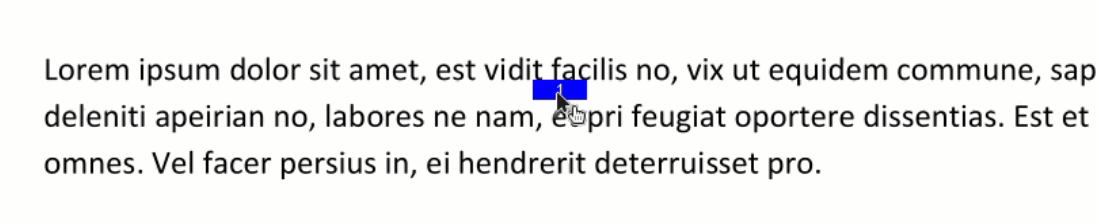
## Moving Comment Pushpins

You can move an existing pushpin to a new position in a document. The feature should be enabled by administrator[185] for all participants or for review creators only.

There are two ways to move a comment:

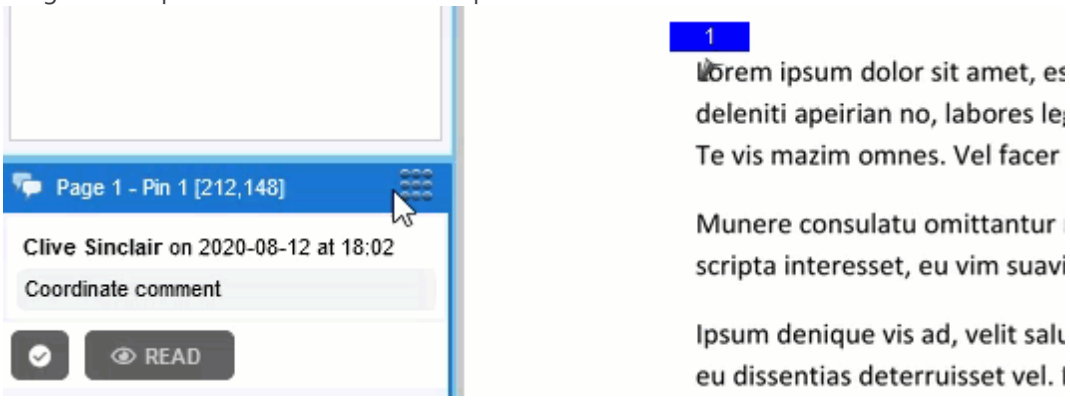
### via Drag and Drop

1. Drag the pushpin mark and drop it over the new position in the document.




### via Move Comment handle

1. Select the desired comment.
2. Press and hold the Move Comment handle ( ).
3. Drag and drop the handle over the new position in the document.



## Document Review Notes

- If a presentation contains tables, they are compared row by row. For more refined comparison of table data consider exporting table as spreadsheet and compare it separately[389].
- If the DiffViewer has troubles displaying small characters, diagrams or documents look blurry, you will need to increase the resolution scale via the `com.smartbear.diff.image.resolution.scale`[1232] Java VM option, restart Collaborator server and clear browser cache.

#### 4.3.4.6 Reviewing Vector Graphics

Collaborator supports reviewing vector graphics of the following formats:

- Microsoft Visio (.vdw, .vdx, .vsd, .vsdm, .vsdx, .vss, .vssm, .vssx, .vst, .vstm, .vstx, .vsx and .vtx.)

**Note:** *Reviewing vector graphics is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page*[3].

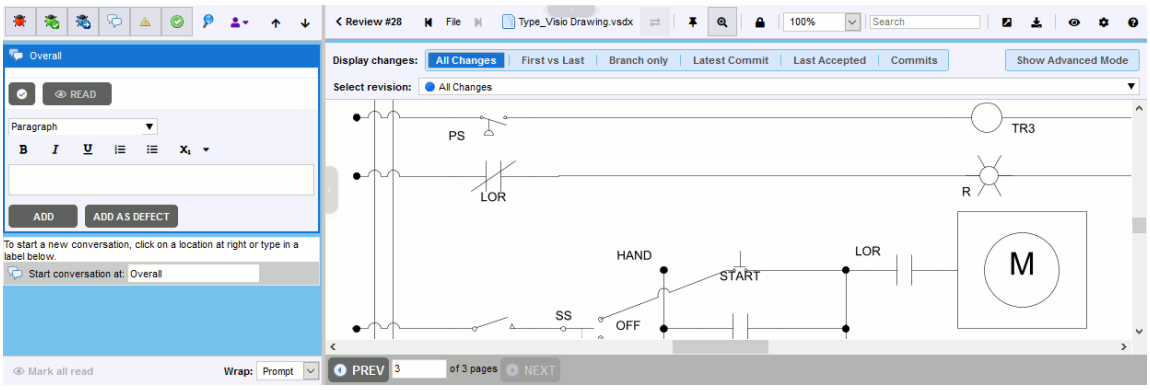
**Important:** Document reviews is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

## Uploading Vector Graphics

To review Visio vector graphics, just attach[343] them to the review as you would any other file and when you open the Diff Viewer, the content area will display the presentation contents for review.

## Viewing Differences

Vector graphics can only be viewed in a **single** or **side-by-side** layout. Diff Viewer cannot highlight differences in graphical objects, however it is able to highlight changes in text contents: c hanges are yellow, additions are green and deletions are red.




*Reviewing Presentations*

Scrollbars are available for each revision of the document. You are also able to move the separator to show more or less of either revision. To do this, hover the mouse over the separator until the arrows appear, and click and drag until it reaches the desired location.

## Changing Pages

To change pages (when a drawing consists of multiple pages), use the arrows at the bottom left of the comparison screen.


The arrows will allow you to go to the **Previous Page** or the **Next Page**. You can also manually enter a **page number** and hit enter to jump to any arrybitrary page. The page numbers will refer to the latest upload of the document shown.

You can also use the mouse wheel to scroll through the pages (unless disabled by the respective option[325]).

Page navigation is disabled when the document is loading or when it is converting to Collaborator's internal format. In that case, Collaborator displays progress indicators near the navigation buttons.

## Zooming In And Out

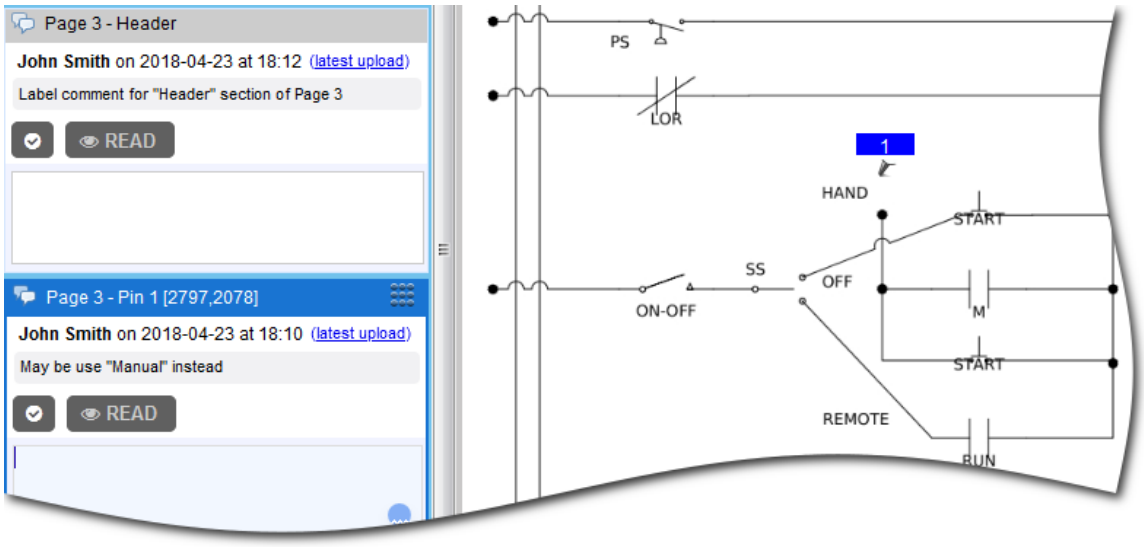
Both the side-by-side and single view support zooming[368] the images in the display. Zooming out is useful if the page is too large to display in the browser window without scrolling. Zoom is also useful for more precise placement of the comment markers. However, whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

Rescaling of documents is performed by rerendering the document on the Collaborator server and could be slow depending on the size of the document and the selected image scale.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing vector graphic files, you can create global[423], annotation[423], overall for file revision[424], coordinate (pushpin)[425] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on specific location, simply click the image at a point where you would like your comment to appear and begin typing. Collaborator will insert a pushpin ( 1 ) to indicate the comment. The number in the pushpin head corresponds to the comment's order within the current image.




You can hide pushpins by clicking the pushpin toggle button on the chat toolbar. Use this feature if a pushpin overlaps an important part of the document.

Alternatively, you may specify an arbitrary label, such as "Header section", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in a text document resulting in the pushpins not being adjacent to the corresponding text. Label text may not be updated.

To learn more about communicating during the reviews, see Types of Review Comments and Defects[421] and Review Chats, Comments and Defects[429] topics.

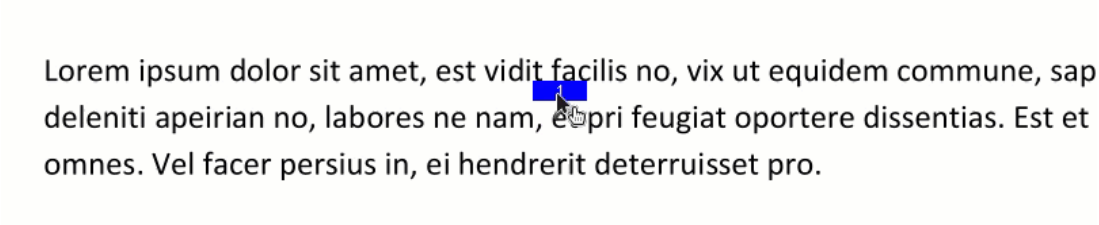
## Moving Comment Pushpins

You can move an existing pushpin to a new position in a document. The feature should be enabled by administrator[185] for all participants or for review creators only.

There are two ways to move a comment:

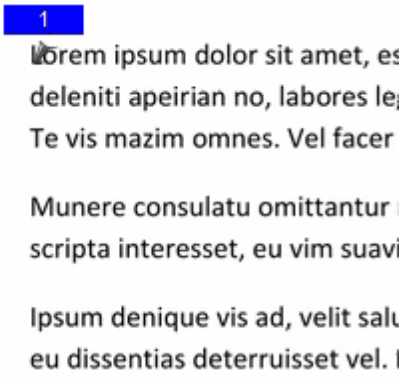
### via Drag and Drop

1. Drag the pushpin mark and drop it over the new position in the document.




### via Move Comment handle

1. Select the desired comment.
2. Press and hold the Move Comment handle ( ).
3. Drag and drop the handle over the new position in the document.


## Document Review Notes

- In multi-page drawings, the first page has smaller scale than other pages. The third-party library used for processing Visio graphics, mimics the behaviour of the original Microsoft Visio application which also saves the first page smaller than other pages.

- If the DiffViewer has troubles displaying small characters, diagrams or documents look blurry, you will need to increase the resolution scale via the `com.smartbear.diff.image.resolution.scale` 1232 Java VM option, restart Collaborator server and clear browser cache.

#### 4.3.4.7 Reviewing Simulink Models

Collaborator supports reviewing MathWorks Simulink models that are exported as web view archives.

**Note:** *Reviewing Simulink models is only supported in Collaborator Enterprise edition with additional Simulink integration licenses purchased and assigned to users. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

The Collaborator Visual Studio Extension and Eclipse Plug-in do not support reviewing Simulink models. Use the Web Client instead.

## Requirements

- MathWorks Simulink version 2018 and newer with Simulink Report Generator is installed.
- Collaborator Enterprise edition with additional Simulink integration licenses is installed.
- Simulink integration license is assigned 210 to your user account.
- (Optional.) Collaborator Simulink Reviewer App 595 is installed in Simulink and configured.

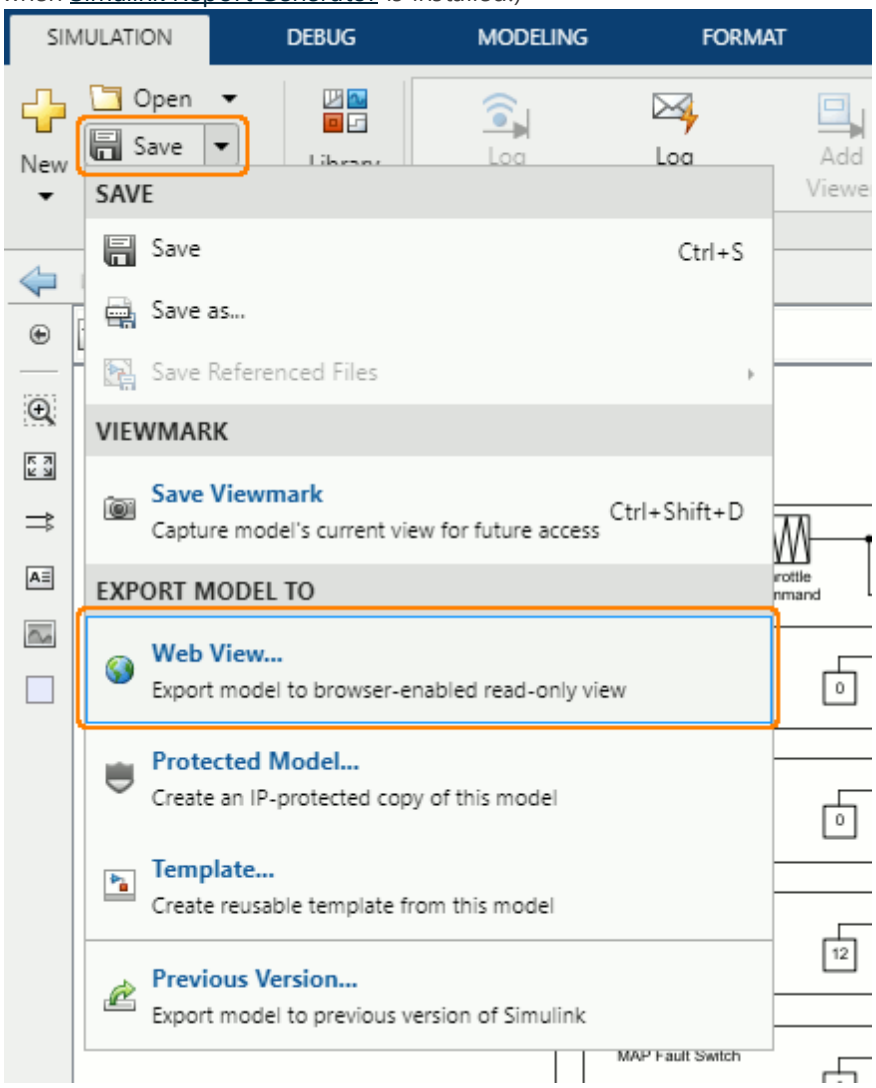
## Uploading Simulink models

There are two ways to upload a Simulink model to a Collaborator review: export it manually with the Report Generator and then upload it to Collaborator server, or use the Collaborator Simulink Reviewer App to export and upload model directly form Simulink.

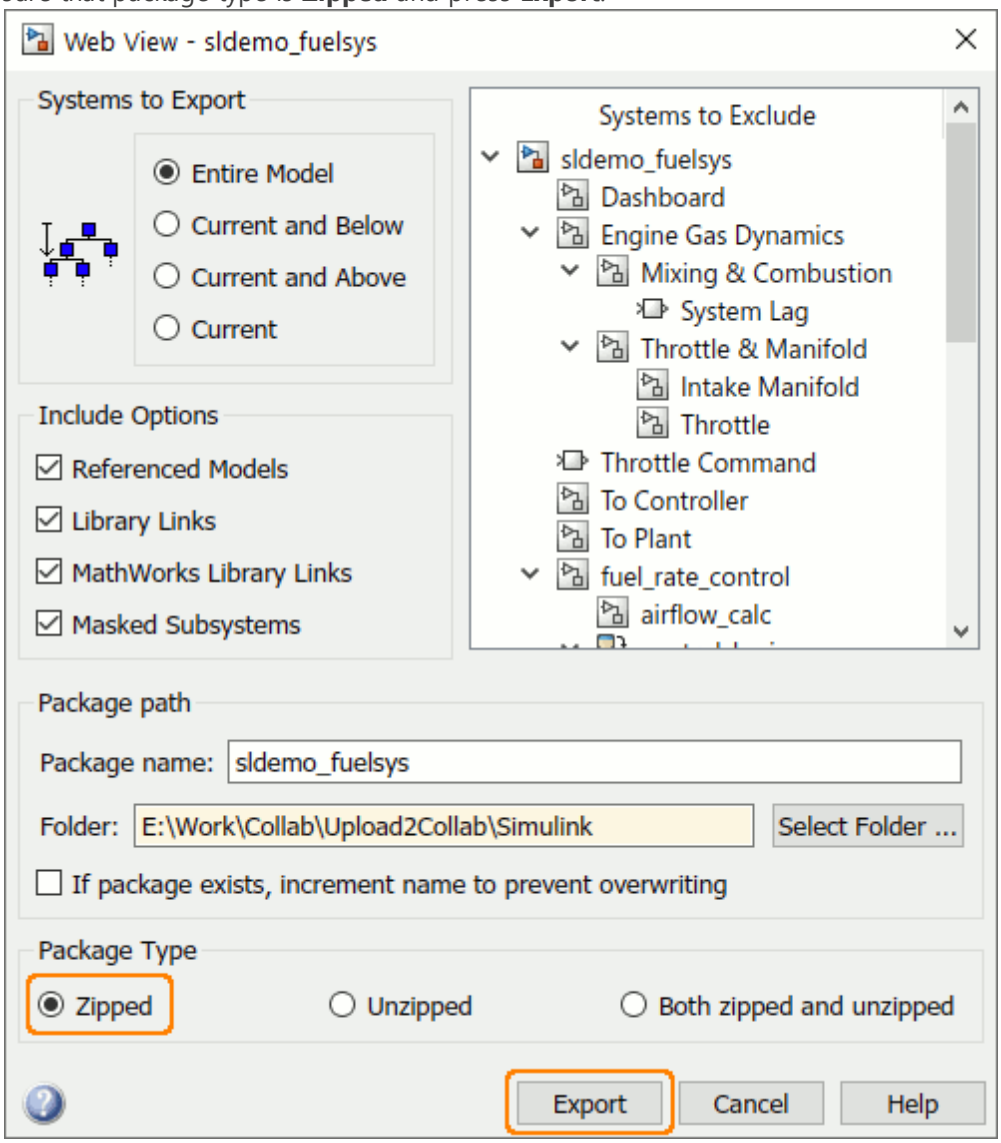
### Export and upload manually

1. Open the desired model in MathWorks Simulink.
2. Switch to the **Simulation** tab and click the down-arrow next to **Save** button.

3. In the drop-down menu select **Export model to > Web View**. (This menu item is available when Simulink Report Generator is installed.)



4. In the subsequent dialog, select which systems to export, specify output folder path, make sure that package type is **Zipped** and press **Export**.




5. Log into Collaborator Web Client.

6. Create new review or open an existing one.

7. Scroll to the **Review Materials** section.

8. Click **Upload** and then select **Simulink Archives** from the drop-down list.

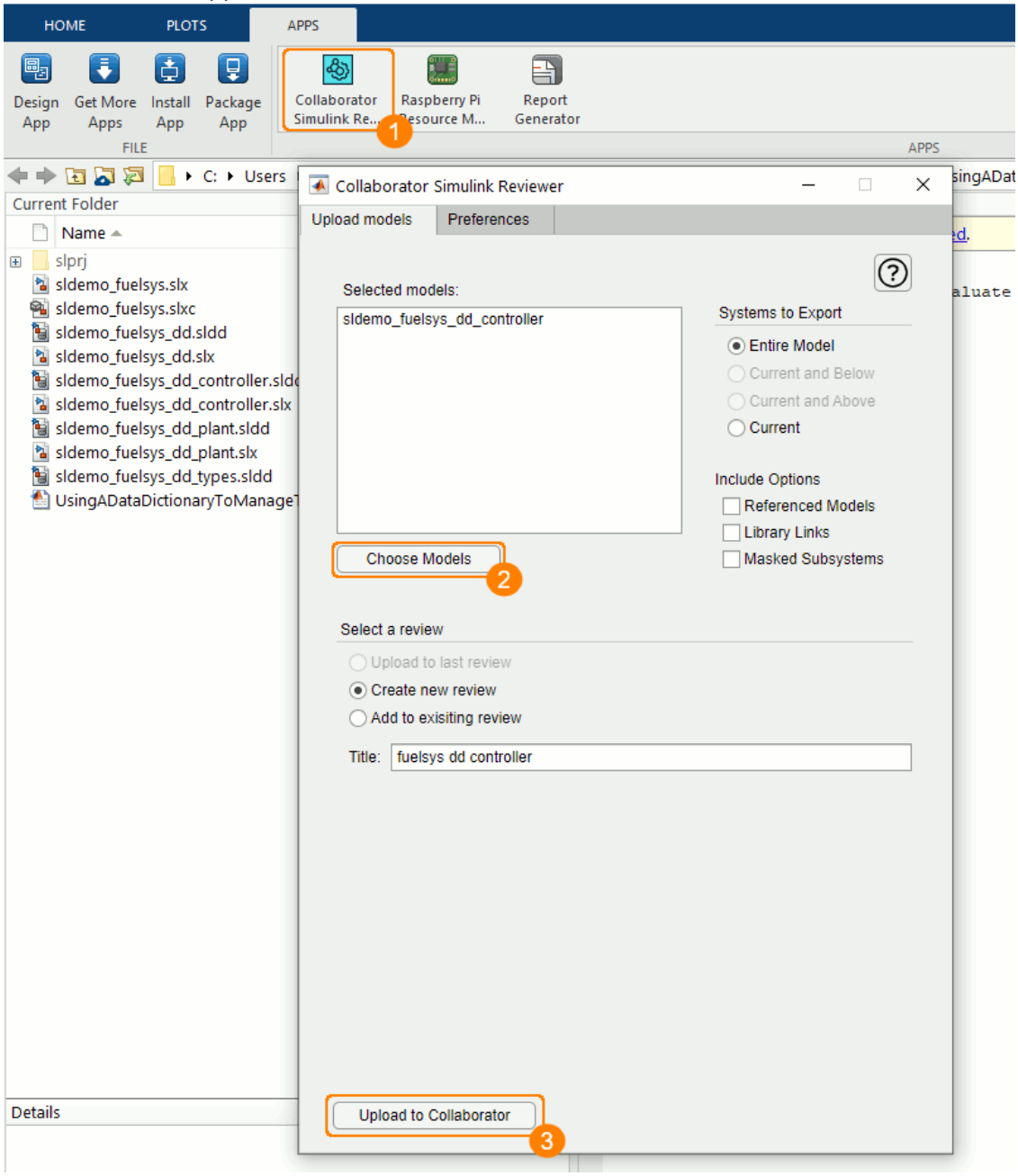
9. In the ensuing dialog, choose one or more Simulink web view archives you want to attach and then click **Upload**.

## Export and upload with Collaborator Simulink Reviewer App

1. Install and configure Collaborator Simulink Reviewer App 595.

2. (Optional.) Open the desired model in MathWorks Simulink.

3. Switch to the **Apps** tab and click **Collaborator Simulink Reviewer**. This will launch the Simulink Reviewer App.



4. In the subsequent dialog, switch to the **Upload models** tab, press **Choose models** and select one or more models you want to upload.

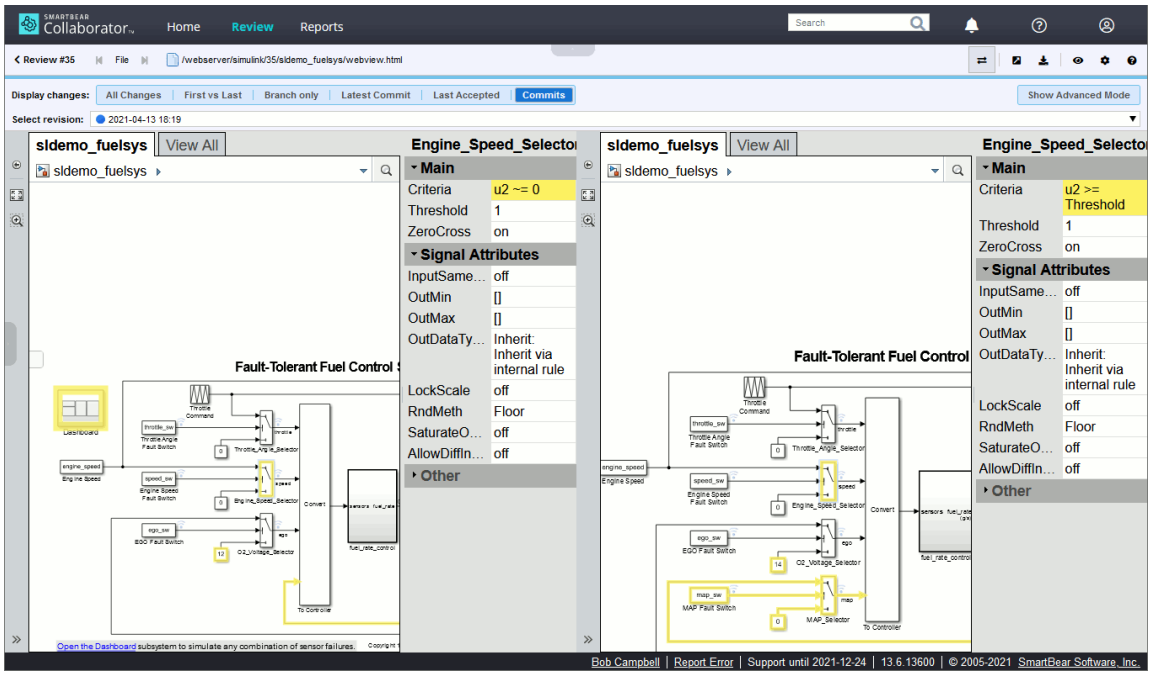
5. Select which systems to export and specify include options.

6. Specify whether to upload model to the last review that was created with Simulink Reviewer App, to create a new review or to upload it to some existing review.

7. Click **Upload to Collaborator**.

## Viewing differences

The UI displays differences between two revisions of a Simulink model in a **side-by-side** layout. Diff Viewer can highlight differences in block properties, as well as differences in graphical objects: changes are yellow, additions are green and deletions are red.




*Reviewing Simulink models*

Selecting a block will display its properties in the right side of the revision pane. Changes in property values are highlighted as well.

## Display options

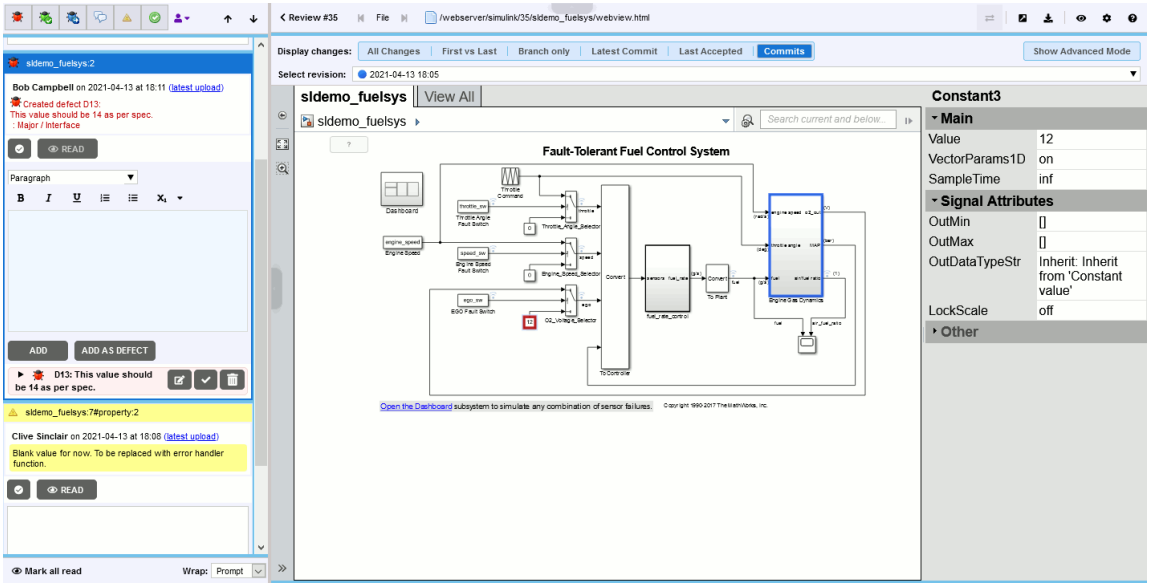
The Display panel 369 has specific options for Simulink model review.

- When the **Show diff based on block properties** option is enabled Diff Viewer will highlight blocks whose properties have been added, deleted or changed.
- When the **Show diff based on image comparison** option is enabled Diff Viewer will highlight graphical objects that have been added, deleted or changed.

## Making comments and marking defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing Simulink models, you can create global[423], annotation[423], overall for file revision[424], element[428] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on a specific element of a model, click the desired element (block, property, line and so on) in the content view, type your comment in the text box and click **Add**. To add defects, click the desired element in the content view, type the defect description, click **Add as defect** and fill-in the required fields.




Alternatively, you may specify an arbitrary label, such as "Header section", to describe the content you are referring to.

To learn more about communicating during the reviews, see Types of Review Comments and Defects[421] and Review Chats, Comments and Defects[429] topics.

## Technical details and limitations

- To export Simulink models as web view archive you should have MathWorks Simulink version 2018 and newer with Simulink Report Generator installed.
- To use the Collaborator Simulink Reviewer App 595 you should have MathWorks Simulink version 2019 and newer with Simulink Report Generator installed.
- Any Collaborator user can upload Simulink archive to Collaborator server. However, to review the archive contents, users should have Simulink integration license. These licenses are obtained from SmartBear additionally to Collaborator Enterprise licenses and are assigned to particular users 210 by your Collaborator administrators.
- Simulink integration licenses can be re-assigned only once in 24 hours.
- When exporting models as web view archive, Simulink does not export complete information about a model. Consequently, some information will not be available to Collaborator and cannot be reviewed.

### 4.3.4.8 Reviewing Images

Collaborator supports reviewing images of the following formats:

- Joint Photographic Expert Group File (.jpg, .jpeg),
- Portable Network Graphics (.png),
- Graphics Interchange Format (.gif),
- Bitmap Image Files (.bmp).

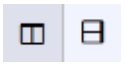
***Note:*** *Reviewing images is supported in Collaborator Team and Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3 *.*

Image files must be renderable by the browser and be configured by the administrator 197 to be treated as images.

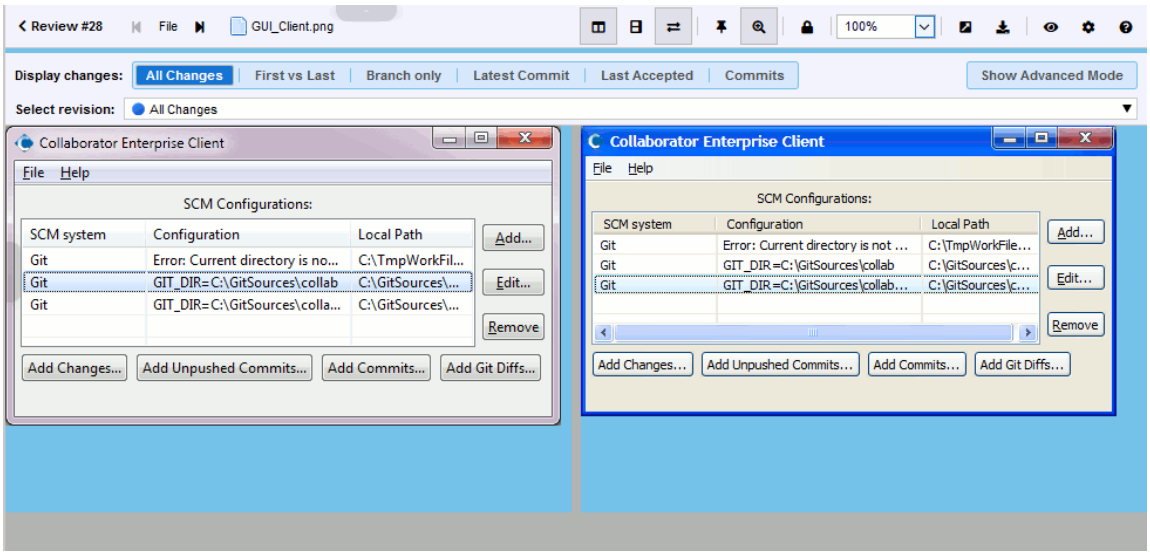
## Uploading Images

To review images, just attach 343 them to the review as you would any other file, and when you open the Diff Viewer, the content area will display the images for review.

## Viewing Differences

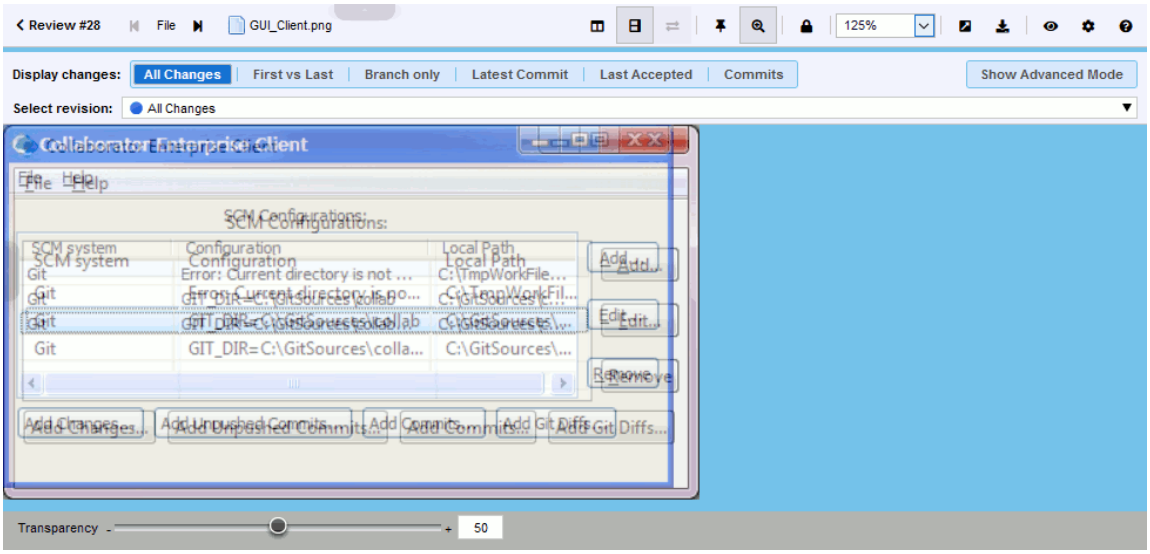
Images can be reviewed in two different manners: **side-by-side** or **overlaid**. Toggling between the two layouts is as simple as clicking the  **Orientation** button on the toolbar.

The **side-by-side view** displays the before and after images separately, one next to another. This layout is useful to see both images at the same time.




*Side-by-Side Image Review*

The **overlaid view** displays the after image over the before image (as image layers on photo editors). The **Transparency** selector adjusts the opacity of the top image and thus allows you to blend the two images together. This layout is useful to see the subtle differences between two images, including alignment differences and color variation.




*Overlaid Image Review*

## Zooming In And Out

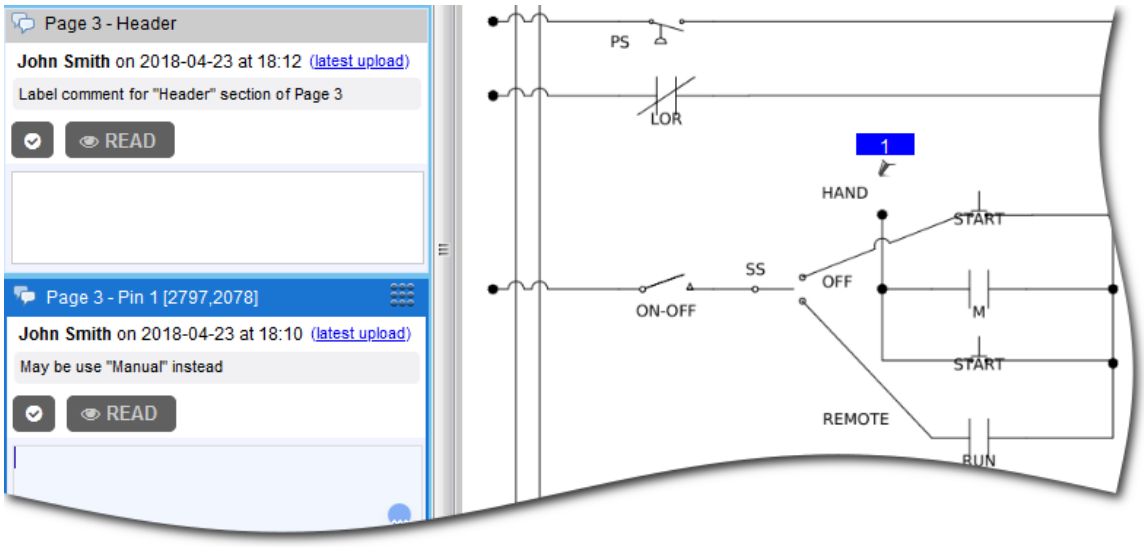
Both the side-by-side and overlaid view support zooming the images[368] in the display. Whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

The image rescaling for the zoom function is provided by the browser and may introduce artifacts depending on the type of image and the scaling algorithm chosen by the browser.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing image files, you can create global[423], annotation[423], overall for file revision[424], coordinate (pushpin)[425] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on specific location, simply click the image at a point where you would like your comment to appear and begin typing. Collaborator will insert a pushpin ( 1 ) to indicate the comment. The number in the pushpin head corresponds to the comment's order within the current image.




The pins may be turned off by clicking the pushpin toggle in the image review toolbar at the bottom of the content pane. This is useful when the pushpins cover important parts of the image.

Alternatively, you may specify an arbitrary label, such as "Header section", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in an image resulting in the pushpins not being adjacent to the corresponding context. Label text may not be updated.

To learn more about communicating during the reviews, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

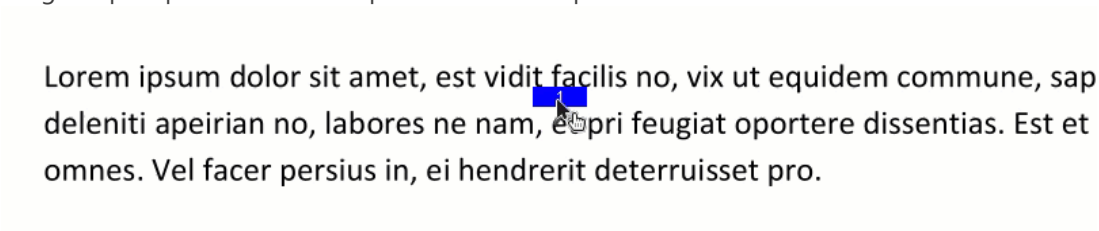
## Moving Comment Pushpins

You can move an existing pushpin to a new position in a document. The feature should be enabled by administrator 185 for all participants or for review creators only.

There are two ways to move a comment:

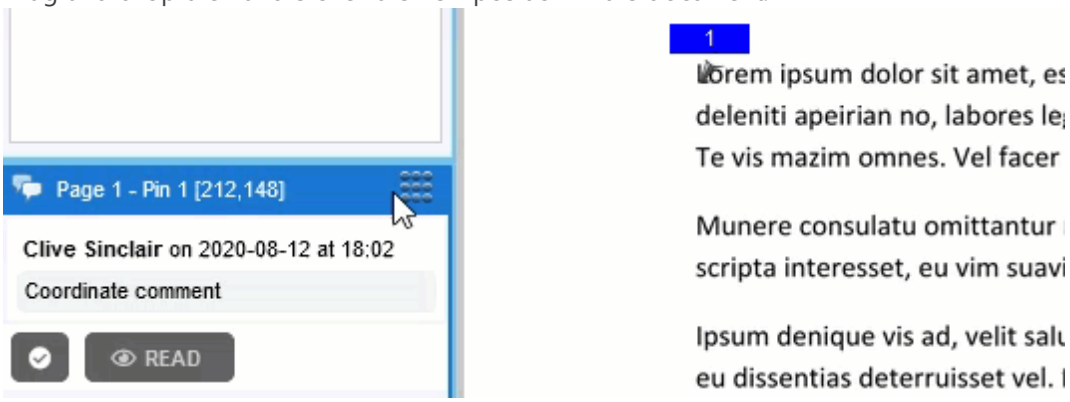
### via Drag and Drop

1. Drag the pushpin mark and drop it over the new position in the document.




### via Move Comment handle

1. Select the desired comment.
2. Press and hold the Move Comment handle ( ).
3. Drag and drop the handle over the new position in the document.



#### 4.3.4.9 Reviewing PDFs and Documents of Other Types

Collaborator includes native support for PDF files. Additionally, you may convert any other document type into PDF format and upload it to Collaborator.

**Note:** *Reviewing PDF documents is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

**Important:** Document reviews is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

## Uploading PDFs

To review PDF documents, just attach 343 them to the review as you would any other file and when you open the Diff Viewer, the content area will display the documents for review.

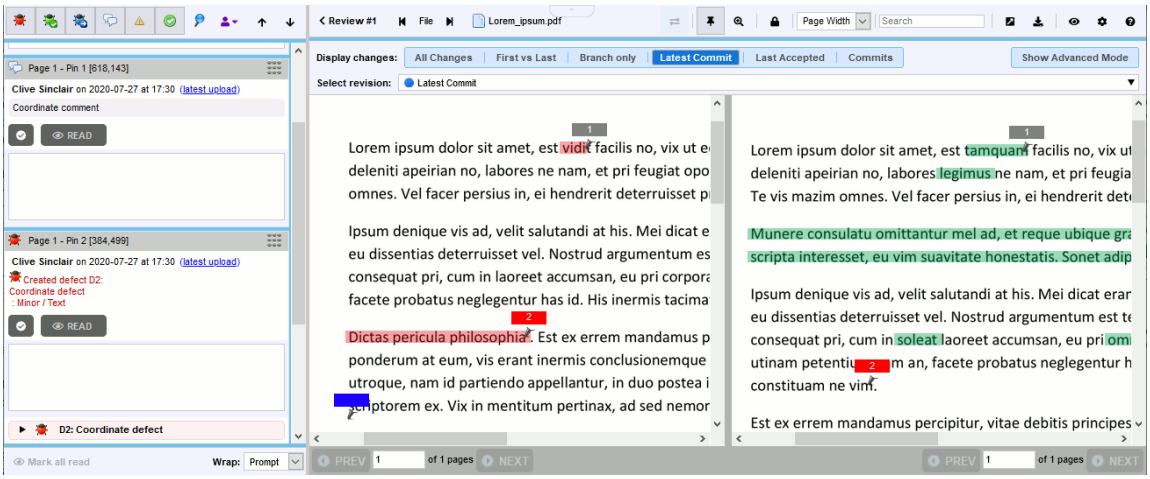
## Uploading Arbitrary Document Types

To upload any arbitrary document, you need to convert the desired document to PDF format using a conversion tool and then upload it to the review. Popular converters include (but are not limited to):

- OpenOffice
- PrimoPDF
- PDFtk

## Viewing Differences

Collaborator automatically finds and highlights differences between the document revisions that you uploaded and selected for review. Both revisions uses standard highlighting: red for deletions and green for additions. Documents can only be viewed in a **single** or **side-by-side** layout. There is no over-under view available currently for PDF documents.




*Side-by-Side PDF Review*

If a document contains tables, they are compared row by row. For more refined comparison of table data consider exporting table as spreadsheet and compare it separately[389].

Scrollbars are available for each revision of the document. You are also able to move the separator to show more or less of either revision. To do this, hover the mouse over the separator until the arrows appear, and click and drag until it reaches the desired location.

## Changing Pages

To change pages, use the arrows at the bottom left of the comparison screen.


The arrows will allow you to go to the **Previous Page** or the **Next Page**. You can also manually enter a **page number** and hit enter to jump to any arbitrary page. The page numbers will refer to the latest upload of the document shown.

You can also use the mouse wheel to scroll through the pages (unless disabled by the respective option[325]).

Page navigation is disabled when the document is loading or when it is converting to Collaborator's internal format. In that case, Collaborator displays progress indicators near the navigation buttons.

## Zooming In And Out

Both the side-by-side and single view support zooming[368] the images in the display. Zooming out is useful if the page is too large to display in the browser window without scrolling. Zoom is also useful for more precise placement of the comment markers. However, whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

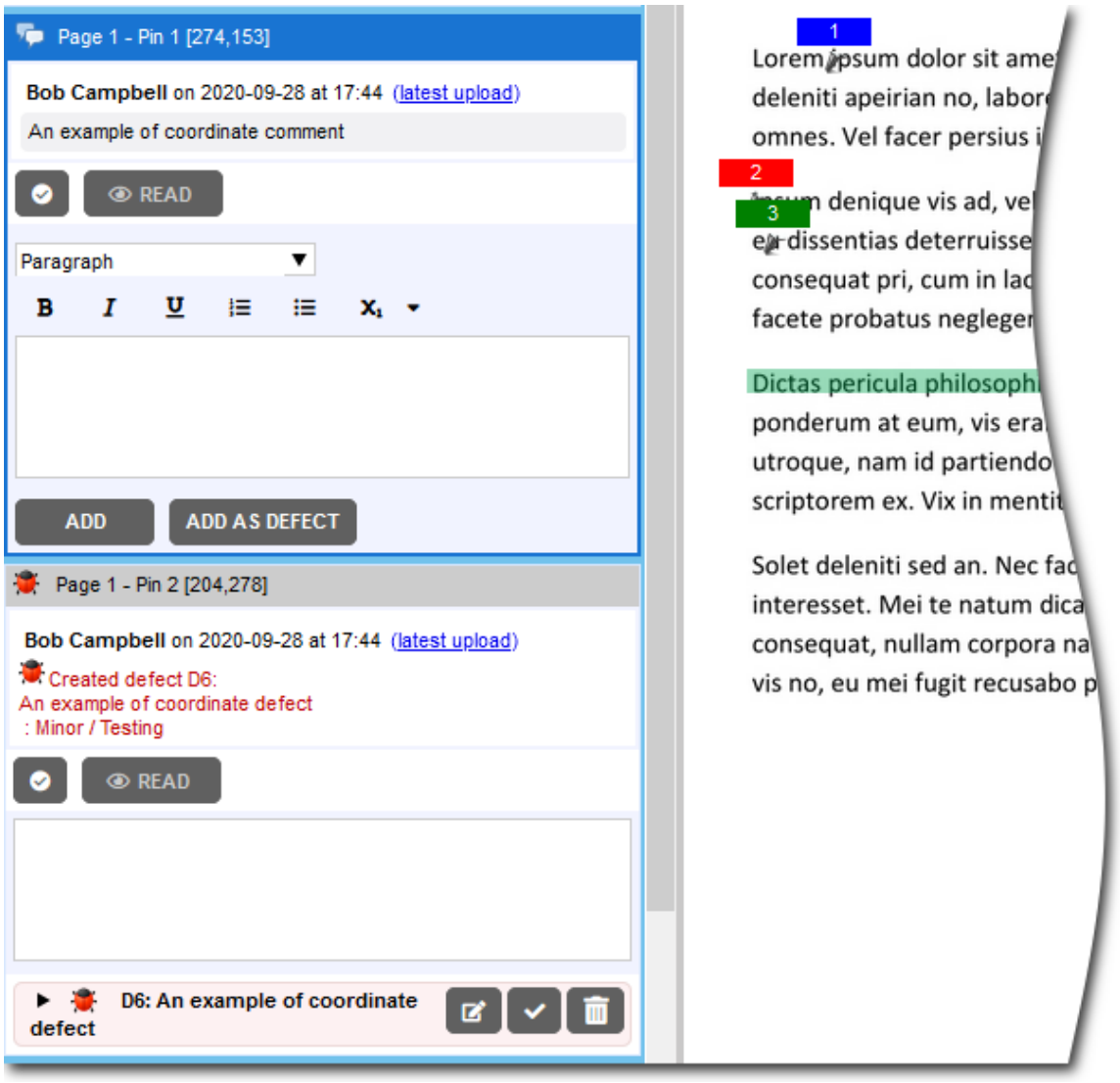
Rescaling of documents is performed by rerendering the document on the Collaborator server and could be slow depending on the size of the document and the selected image scale.

## Making Comments and Marking Defects

On the left of the main Diff Viewer page, there is a pane for chat threads, where you can view and make comments and mark defects that should be fixed. When reviewing PDF files, you can create global[423], annotation[423], overall for file revision[424], coordinate (pushpin)[425] and label[427] comments and defects. Comment and defect description could be in plain-text or use rich-text and Markdown formatting[447], they could also mention other Collaborator users[445].

To comment on specific text in a document, simply click the document at a point where you would like your comment to appear and begin typing. Collaborator will insert a pushpin ( 

) to indicate the comment. The number in the pushpin head corresponds to the comment's order within the current page of the document.




You can hide pushpins by clicking the pushpin toggle button  on the chat toolbar. Use this feature if a pushpin overlaps an important part of the document.

Alternatively, you may specify an arbitrary label, such as "Section 5.1", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in a text document resulting in the pushpins not being adjacent to the corresponding text. Label text may not be updated.

To learn more about communicating during the reviews, see Types of Review Comments and Defects 421 and Review Chats, Comments and Defects 429 topics.

It's obvious that document content will change from one revision to another. For example, a fragment could move from top of the page to its bottom, or even to another page. The Diff Viewer attempts to recognize these changes and promote comments to correct places in the newer version. Besides it navigates the Before and After panes independently to make this content visible during the review. Yet in some cases, for example when multiple pages were inserted into document, Diff Viewer may promote comment positions incorrectly. In this case you could move comments manually, as described below.

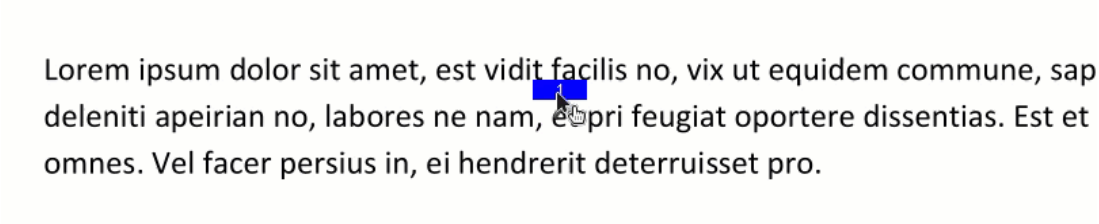
## Moving Comment Pushpins

You can move an existing pushpin to a new position in a document. The feature should be enabled by administrator[185] for all participants or for review creators only.

There are two ways to move a comment:

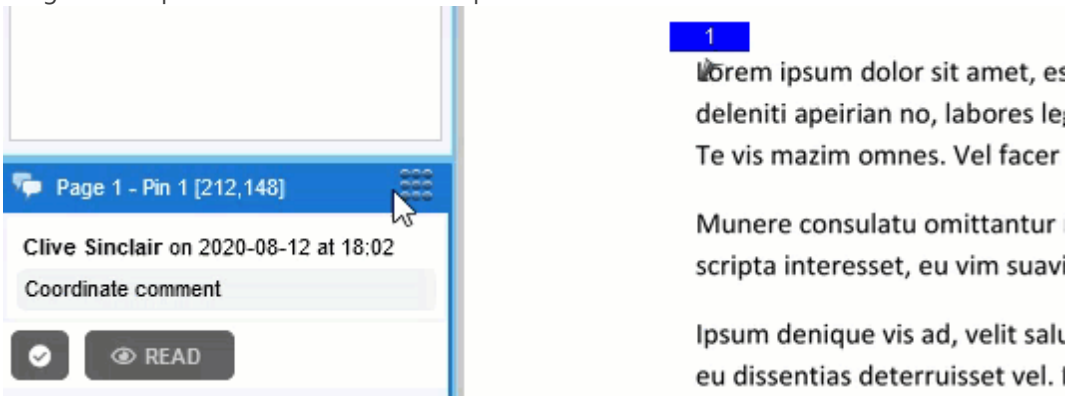
### via Drag and Drop

1. Drag the pushpin mark and drop it over the new position in the document.




### via Move Comment handle

1. Select the desired comment.
2. Press and hold the Move Comment handle ( ).
3. Drag and drop the handle over the new position in the document.



## Document Review Notes

- If the DiffViewer has troubles displaying small characters, diagrams or documents look blurry, you will need to increase the resolution scale via the `com.smartbear.diff.image.resolution.scale` 1232 Java VM option, restart Collaborator server and clear browser cache.

#### 4.3.4.10 Reviewing URLs

URLs can also be reviewed in Collaborator.

**Note:** *Reviewing live URLs is only supported in Collaborator Enterprise. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

However, it is important to note certain limitations. Currently, URLs are rendered in the browser as "live". This implies that clicking on a link on the URL page will direct you to a new page. This also means that you will not be able to anchor comments in the same manner as in image and document reviews.

With our current URL review functionality, newer versions of the URL will not be displayed with older versions, meaning you cannot display multiple versions of the same web page. If you would like to compare multiple URL versions, we recommend that you take static snap shots of the web page and use the image review functionality 410 of Collaborator.

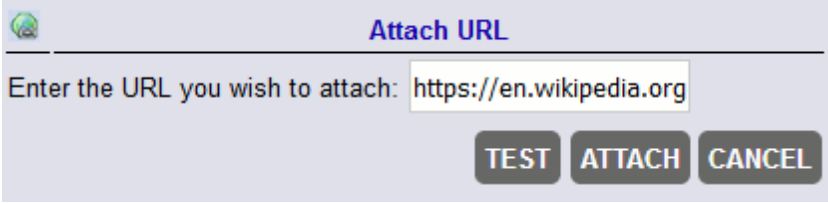
## Uploading URLs

To add a web link:

1. Create a new review or open an existing review,
2. Scroll to the Review Materials section of the Review Summary Screen,
3. Click the **Upload** button on the toolbar,


4. Select **URL** from the drop-down list.

5. In the ensuing dialog, enter the desired URL and then click Attach.




## Making Comments and Marking Defects

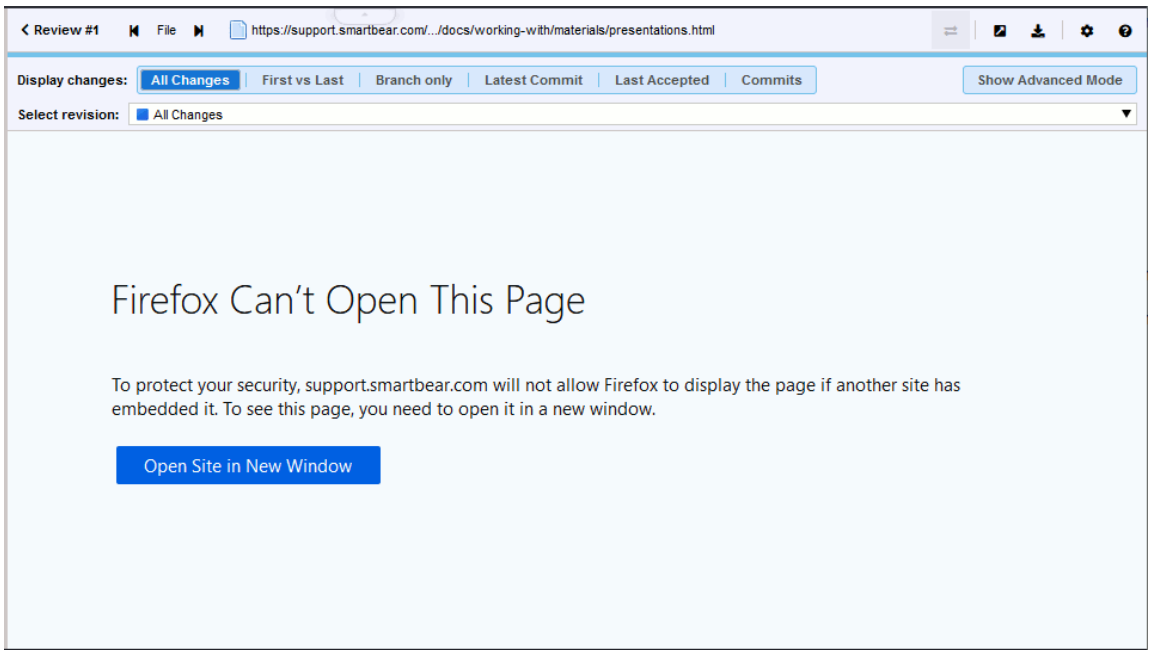
As mentioned above, you will not be able to anchor comments. Instead, you can specify an arbitrary label describing a part of the page or otherwise indicating to other participants what you are commenting on. In older versions of Collaborator, comments in URLs were anchored to a fake "line". Those comments will continue to be displayed, but will be sorted after the new labeled locations.


*URL Review*

**Notes:**

- The URL must be a valid web URL, which should include the protocol specification (either http:// or https://).
- If the specified URL is malformed, you will see an error message.
- Reviewing URLs is intended only for html documents (web pages), all other formats are not supported (for instance, you can not upload pdf document or code file as URL).
- Collaborator does not check whether the specified resource is available. In fact, since it is a link, the contents and availability of the document can change at any time.
- If your Collaborator server uses **https** connection, you will not be able to upload **http** links.

- Collaborator displays the referred resource in a frame. The server can send requested contents with anti-framing headers. You will not see the referred content in this case. Depending on the web browser you use, you will see an empty frame or an error message saying that *the content cannot be displayed in a frame:*




### 4.3.5 Types of Review Comments and Defects

There are two main types of messages in Collaborator: comments and defects.

**Comments** may hold any information concerning the review: a question, a clarification, a remark, an encouragement, whatever else. These are the major mean of communication during the review process.

**Defects** indicate a problem that needs to be fixed. Defects have some text that describes the problem and also can have any number of additional fields. These are all completely configurable 256 by the system administrator and can also vary depending on the review workflow 338 that was originally selected for the review. Because this is completely configurable, this manual cannot say exactly what the fields will be or what they mean; ask your administrator for details. It is common to see fields like severity, type, checklist item, and phase-injected.

Every defect is given a unique number, on the server. There will only ever be a single defect with the given ID, no matter how many defects, documents or reviews exist on the server.

All defects found during a review are listed in the Defect Log 354 section of the Review Summary Screen 345, (so you will not overlook any of them).

**Note:** The word "defect" has many connotations that are inappropriate for peer review. This does not mean the problem will be mirrored in an external issue-tracking system, and it does not necessarily mean it was a bug! Even "bad documentation" can be a defect.

A "defect" is just a way of identifying something that needs to be fixed.

Moreover, if the word "defect" has a negative connotation in your environment, your Collaborator administrator can change it to another term 184.

## Color Highlighting of Comments and Defects

To indicate the status of the comment or defect they are highlighted in different colors. The following color notations are used for message backgrounds in the Chat pane, for line indents of text files, for pushpins and for cell borders:

| Color | Meaning |
|---|---|
| Blue | The currently selected conversation. |
| Red | An open defect. |
| Yellow | An unread comment. |
| Green | A fixed defect or an external defect. |
| Gray | No defects. |

## Global and Content Specific Comments and Defects

Comments and defects can be *global* or *context specific*. That is, they may relate to the entire review, or to certain review materials or even to particular line, coordinate, cell in one of the reviewed files.

Collaborator has the following sub-types of comments and defects:

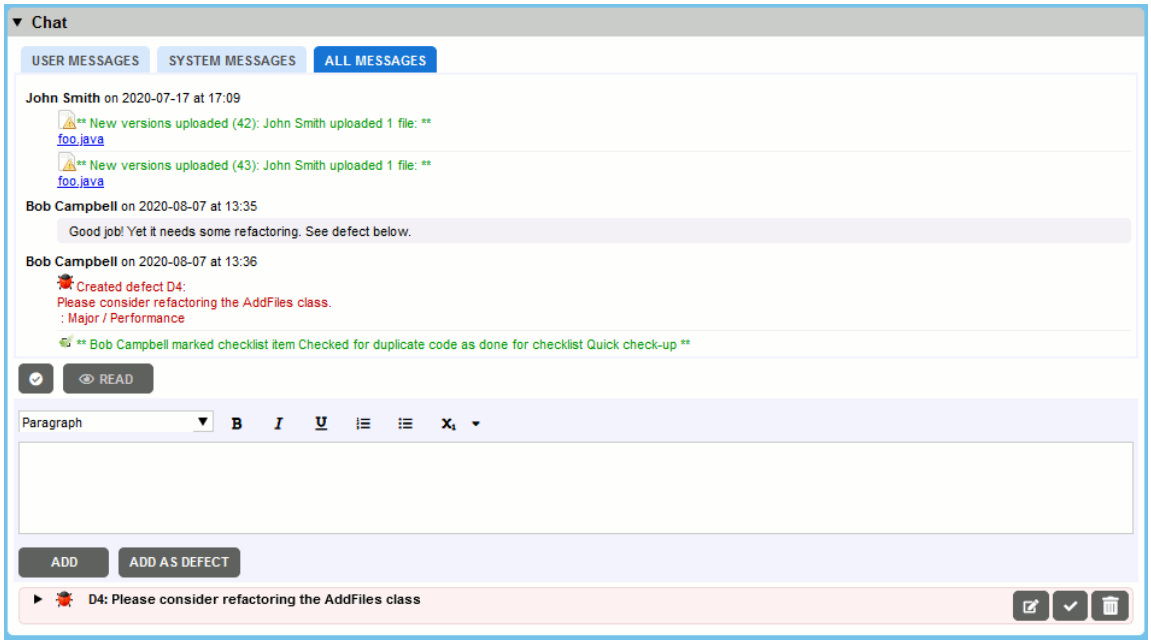
- Global (Review Chat) 423
- Annotation 423
- Overall for File Revision or URL 424
- Line 424

- Coordinate (Pushpin) 425
- Label 427
- Cell 427
- Element 428

## Global Comments and Defects(Review Chat)

These comments or defects relate to the entire review as a whole (not to some specific review materials). Can be used with all reviews.

Global comments and defects are displayed in the Chat 356 section of the Review Summary Screen 345. Besides that, this section displays some helper information: who has marked checklist items, who has signed the review, information about pull requests and branches and so forth. Support rich-text and Markdown formatting 447 and user mentioning 445.

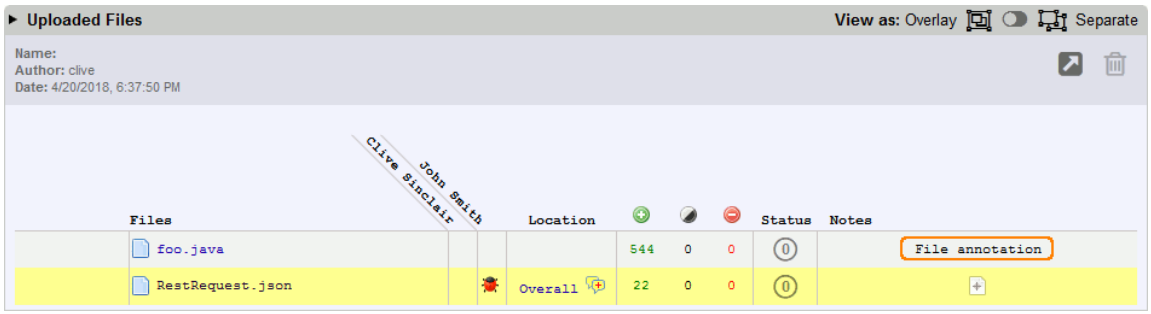



To add comments to the Chat section, simply type your comment in the text box and click "Add". To add defects, type the defect description, click "Add As Defect" and fill-in the required fields.

## Annotation Comments

These type of comments provide short description (annotation, notes or some other short info) about a particular file revision or URL. Can be used with all review materials. Defects cannot be of annotation type.

Annotation comments are displayed in the Notes column of the Review Materials 357 section of the Review Summary Screen 345.

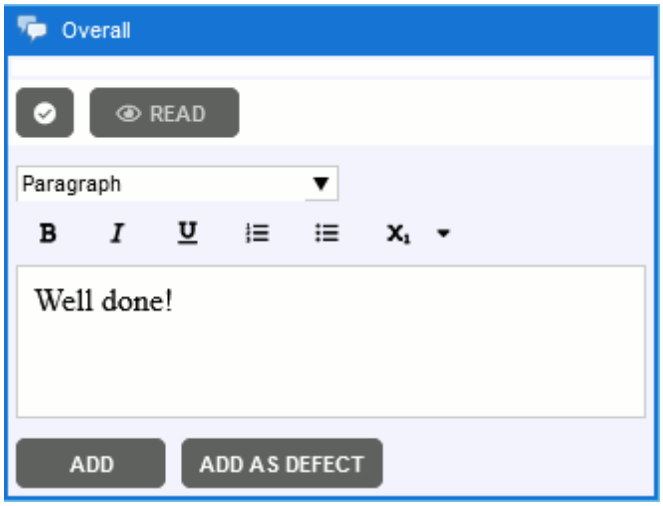



To add annotation comments, click the plus button in the Notes column, enter annotation text and click "Save".

## Overall Comments and Defects for File Revision or URL

These comments or defects relate to a particular file revision or URL (as a whole, but not to some specific part if it). Can be used with all kinds of review materials.

Overall comments and defects are displayed in the Overall section of Chat pane in the Diff Viewer 364. Besides that, this section displays some helper information: a notice that a new file revision has been uploaded, a notice that the revision was approved and so on. Support rich-text and Markdown formatting 447 and user mentioning 445.

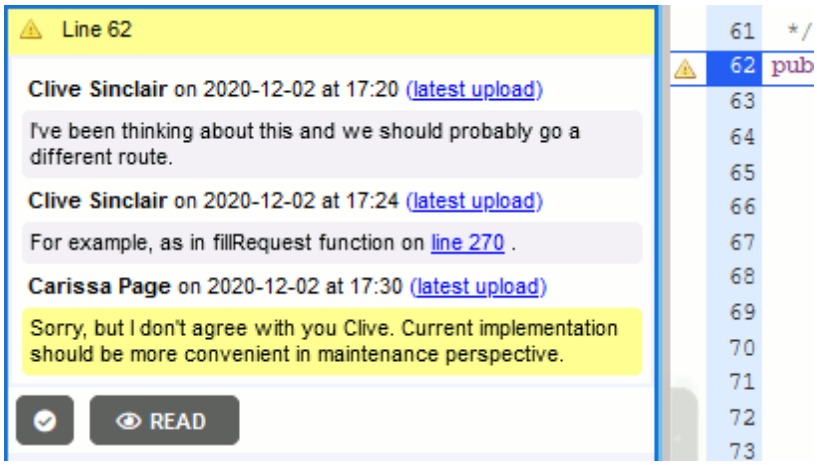



To add comments to the Overall section, simply type your comment in the text box and click "Add". To add defects, type the defect description, click "Add As Defect" and fill-in the required fields.

## Line Comments and Defects

These comments or defects relate to a particular line within a file. Can be used with text-based files and their revisions. Multiple comments and defects can relate to the same line of text.

Line comments and defects are displayed in the Chat pane of the Diff Viewer[364]. Support rich-text and Markdown formatting[447] and user mentioning[445].

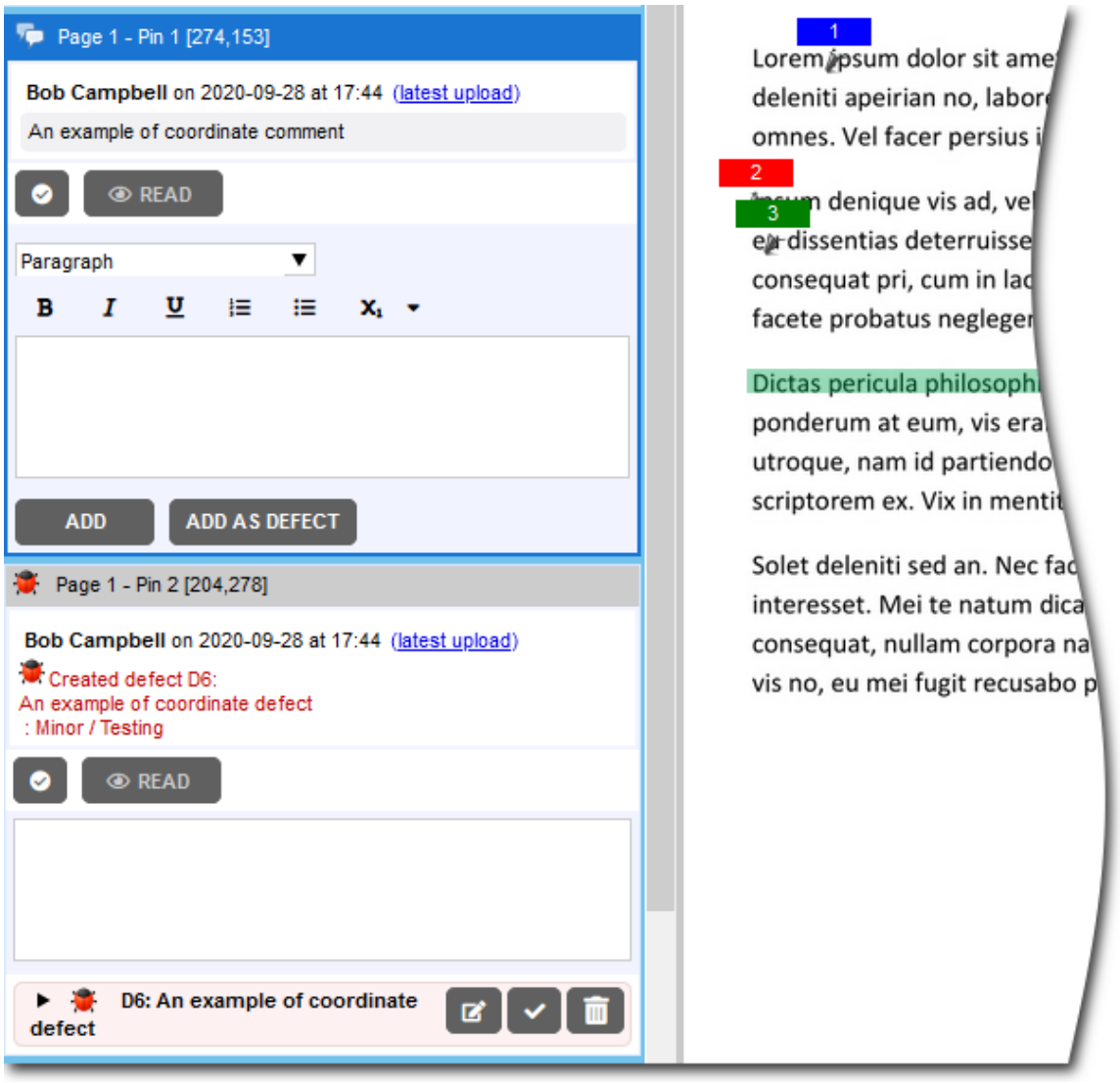



To add line comments, click the desired line of text in the content view, type your comment in the text box and click "Add". To add defects, click the desired line of text in the content view, type the defect description, click "Add As Defect" and fill-in the required fields.

## Coordinate Comments and Defects (Pushpins)

These comments or defects relate to a particular coordinates within a file. Can be used with word-processing documents, presentations, PDF documents, images and vector graphics. Multiple comments and defects can relate to the same coordinate.

Pushpin comments and defects are displayed in the Chat pane of the Diff Viewer[364]. Support rich-text and Markdown formatting[447] and user mentioning[445].




To add pushpin comments, click the desired location in the content view, type your comment in the text box and click "Add". To add defects, click the desired location in the content view, type the defect description, click "Add As Defect" and fill-in the required fields.

Pushpins created via Web Client or command-line client display an integer number in their head: 4 The number corresponds to the order in which that pushpin was added to the document page or image.

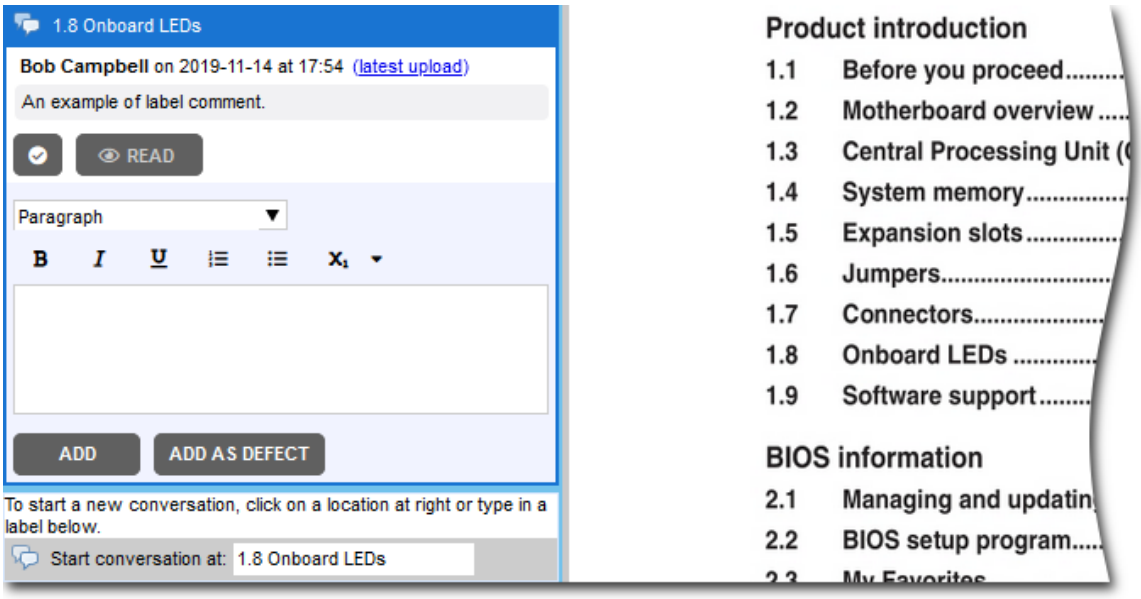
Pushpins created via Eclipse Plug-in do not display any ordinal numbers:

## Label Comments and Defects

These comments or defects relate to an arbitrary label within a file. Can be used with word-processing documents, presentations, PDF documents, images, vector graphics and Simulink models. The label text **must** describe the content (for example, "Section 5.1 on page 23") it relates to. (Since, label comments/defects do not have any other markers that bind the content and the label.) Multiple comments and defects can relate to the same label.

Using labeled locations decreases the confusion that can occur when large changes occur in a document or image resulting in the pushpins not being adjacent to the corresponding context. Label text may not be updated.

Label comments and defects are displayed in the Chat pane of the Diff Viewer 364. Support rich-text and Markdown formatting 447 and user mentioning 445.

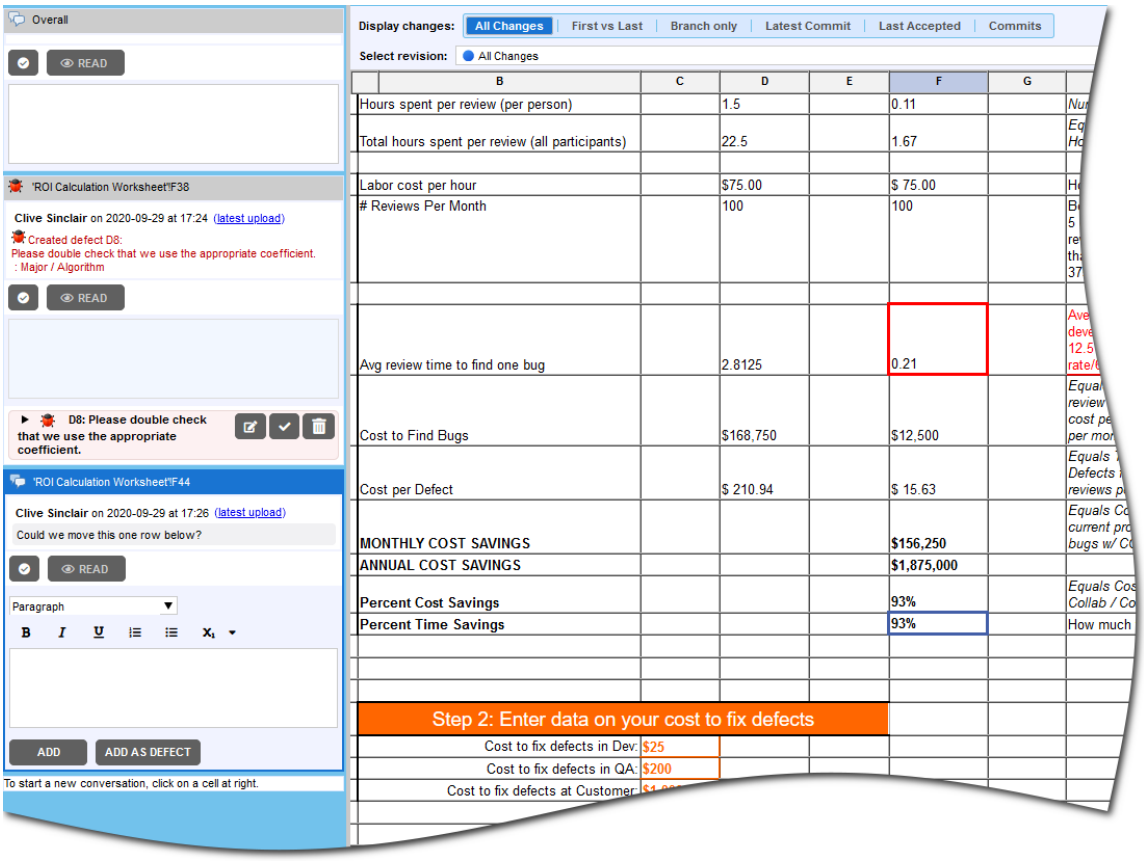



To add label comments, scroll the Chat pane to the "Start conversation at" box, specify label text and text of the comment and click "Add". To add defects, perform the same steps as for label comments, but click click "Add As Defect" and fill-in the required fields.

## Cell Comments and Defects

These comments or defects relate to a particular cell within an Excel spreadsheet. Can be used with Excel files and their revisions. Multiple comments and defects can relate to the same cell. Conversations on cells are displayed in alphabetical order by sheet name, not in the sheet order found in the version content. The name of the chat includes the name of the sheet followed by the location of the cell being discussed. For example, 3sheetC10.

Cell comments and defects are displayed in the Chat pane of the Diff Viewer 364. Support rich-text and Markdown formatting 447 and user mentioning 445.

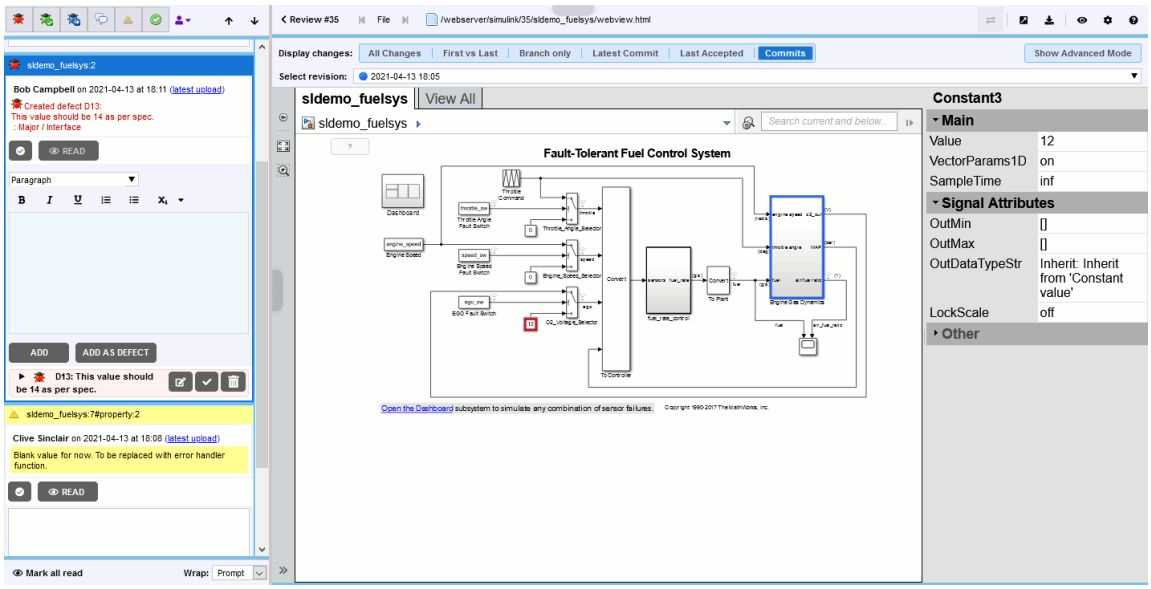



To add cell comments, click the desired cell in the content view, type your comment in the text box and click "Add". To add defects, click the desired cell in the content view, type the defect description, click "Add As Defect" and fill-in the required fields.

## Element Comments and Defects

These comments or defects relate to a particular elements of a Simulink model 404: block, property, line and so on. Can be used with Simulink models. Multiple comments and defects can relate to the same element.

Element comments and defects are displayed in the Chat pane of the Diff Viewer 364. Support rich-text and Markdown formatting 447 and user mentioning 445.




To add element comments, click the desired element (block, property, line and so on) in the content view, type your comment in the text box and click "Add". To add defects, click the desired element in the content view, type the defect description, click "Add As Defect" and fill-in the required fields.

### 4.3.6 Review Chats, Comments and Defects

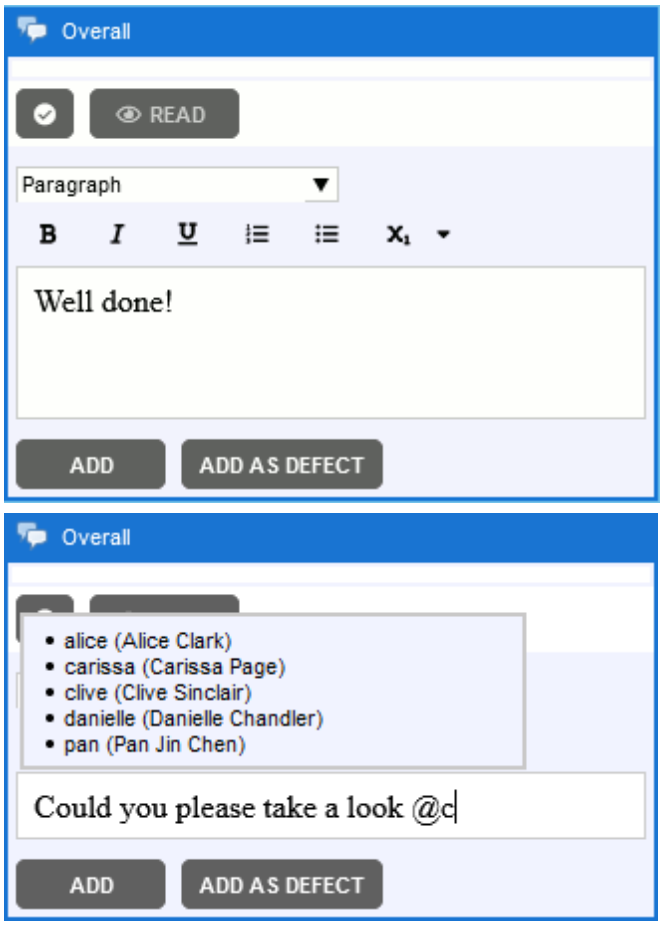
Viewing differences 364 between documents is merely the beginning. The key to reviewing documents is communication.

There are various global and content-specific types of comments and defects in Collaborator. See Types of Review Comments and Defects 421 for detailed description of each type.

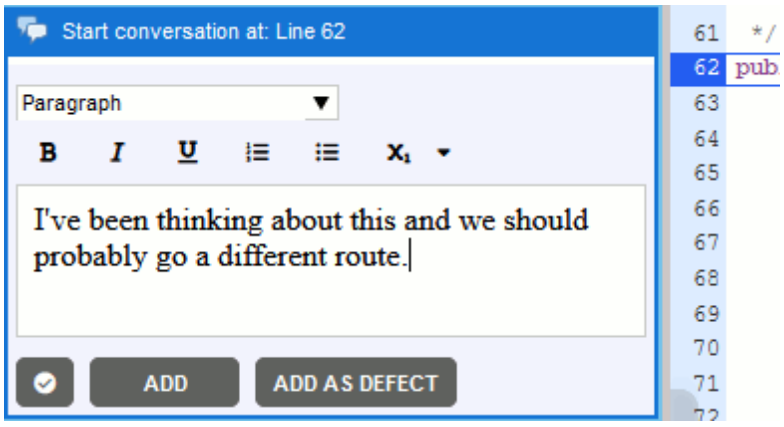
This topic describes how to communicate during a review.

## Making Conversation -- Chatting at particular files, lines, coordinates, cells

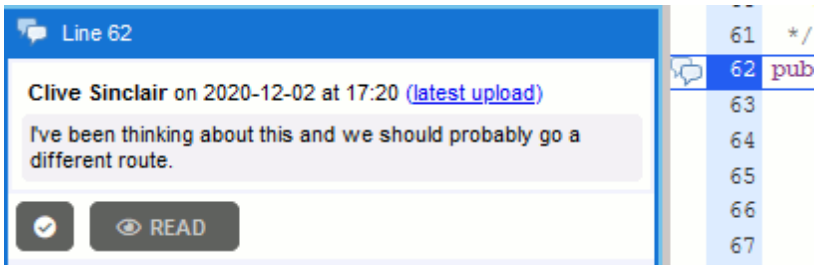
To start a general conversation about some particular file, open the desired file in the Diff Viewer 364, type your comment in the Overall section and click **Add**. Comment could be in plain-text or use rich-text and Markdown formatting 447 and user mentioning 445.




To start a conversation about some particular location (line, coordinates, cell), just click the desired location and start typing. The interface will open up automatically and accept your chat message:

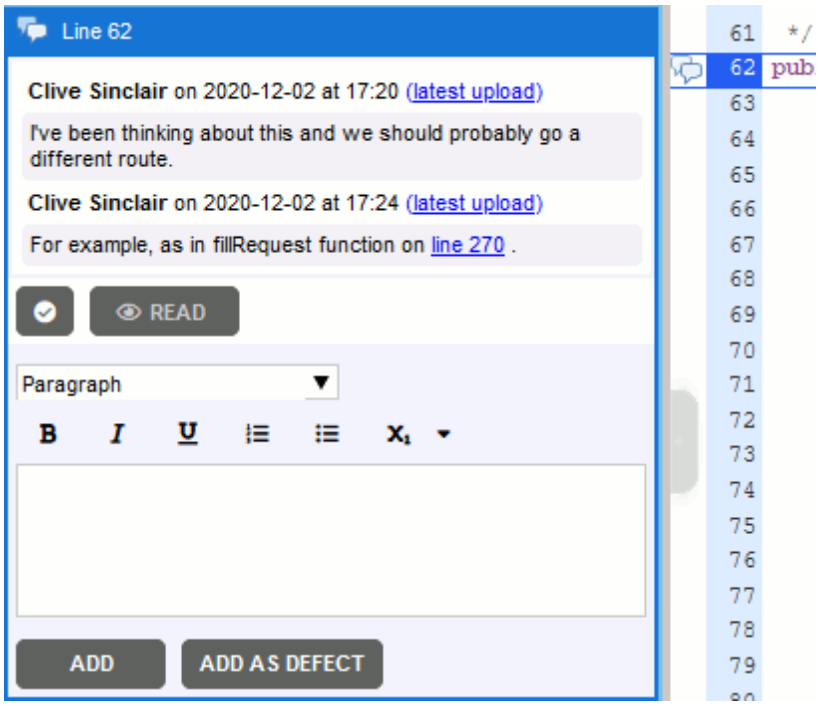



When you click **Add**, your message is transmitted to all other users and you have started a *threaded* conversation for that particular location (line, coordinates, cell). In this case, it would look something like this:




Notice that the conversation is now threaded by line 62 and that a chat icon ( ) now appears next to the source line where the conversation was made.

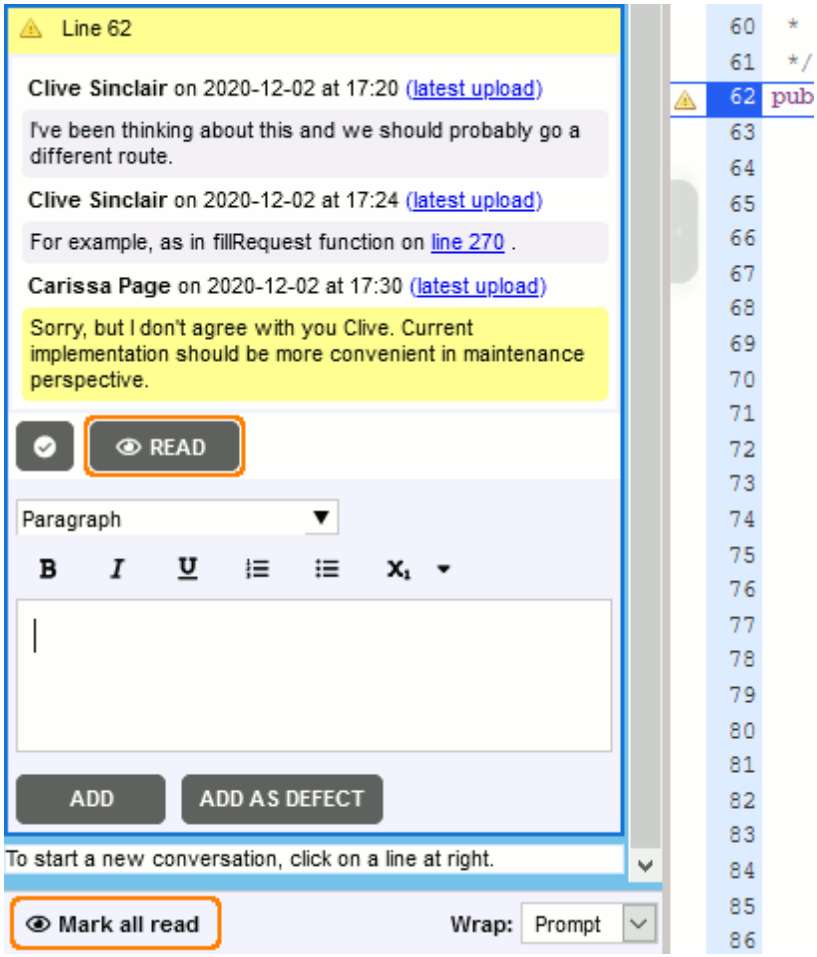
When you type the name of a file in the same review, it will turn into a link to that file. You can also link to a different line number in the current file by typing "line NNN".




When another user chats on that same line, that message appears in yellow and yellow bubbles help to make the conversation stand out as needing to be read. The screen updates immediately without the need for a browser "refresh":

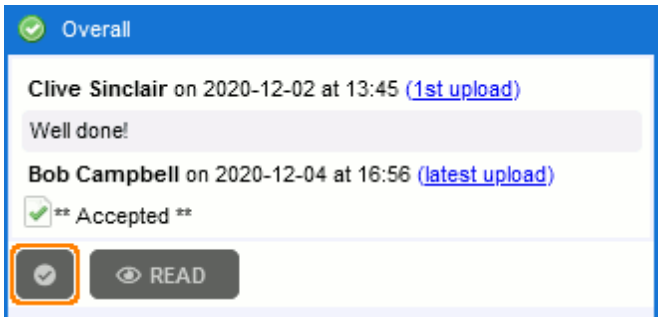

To answer a message, just type your answer in the chat window.

To clear the "unread chat" state without saying something, click the **Read** button (there is also a **Mark all read** button at the bottom of the chat pane):

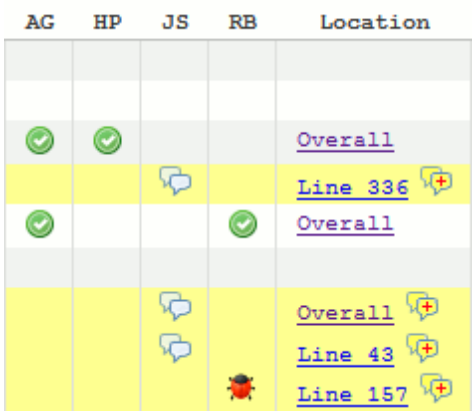


## Accepting Files and Comments

In addition to marking a conversation as read, you can also click the **Mark Accepted** button in the chat to accept the comment or file revision. The **Mark Accepted** button affects conversations/ revision that is currently displayed in the After pane of the DiffViewer 364. The button can be inactive when the conversation/revision is already accepted, or when the revision currently displayed in After pane actually precedes (is earlier than) the revision displayed in Before pane.




"Accept" means whatever you want it to mean. If you want, you can never use the Mark Accepted button at all. The actual effect of clicking Mark Accepted on a conversation is to mark that conversation as read, and to put a green check mark next to that conversation in chat and in the Review Material 357 section of the Review Summary Screen 345. Those check marks are visible to all users.




*Conversation status icons in the Review Materials section*

Some people use it to mean explicit agreement with the conversation (as opposed to simply not replying, which is implicit agreement).

Many people use "Accept" as sort of a bookmark within the review to keep track of what file revisions they have already looked at and agree with its changes.

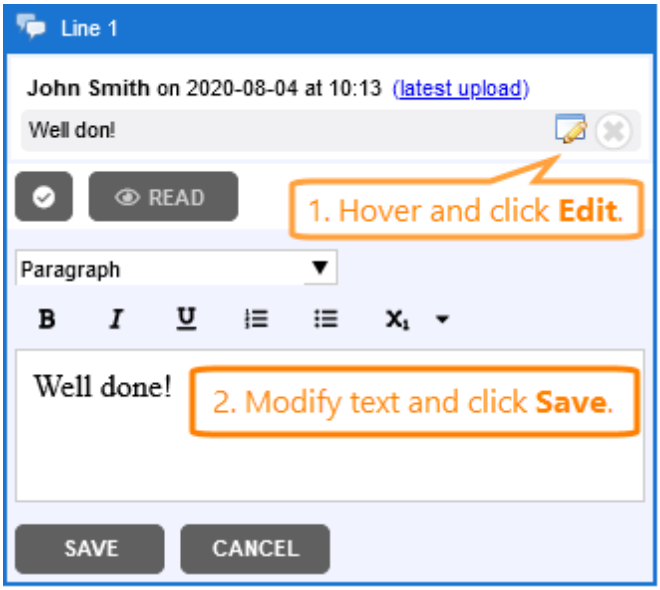
- When the author uploads new revision of a ***document***, the accepted icon ( ) will change back to chat icon ( ) denoting that the participant has not yet approved changes in this new revision.
- When the author uploads new revision of a ***source code or text file*** Collaborator acts a little bit smarter: if the respective line was not changed, the accepted icon ( ) will remain indicating the changes were already accepted; otherwise the accepted icon ( ) will change back to chat icon ( ) denoting that the participant has not yet approved changes in this new revision.

## Edit Comments

By default comment editing is disabled in order to keep all comments intact for auditing purposes. If the the Allow to edit/delete comments 186 server option is enabled, review participants could modify their own comments. Comment could be modified unless it is followed by another participant's comment.

To edit a comment:

1. Mouse-over the specific comment you would like to modify and click **Edit**.
2. Modify comment text and click **Save**.

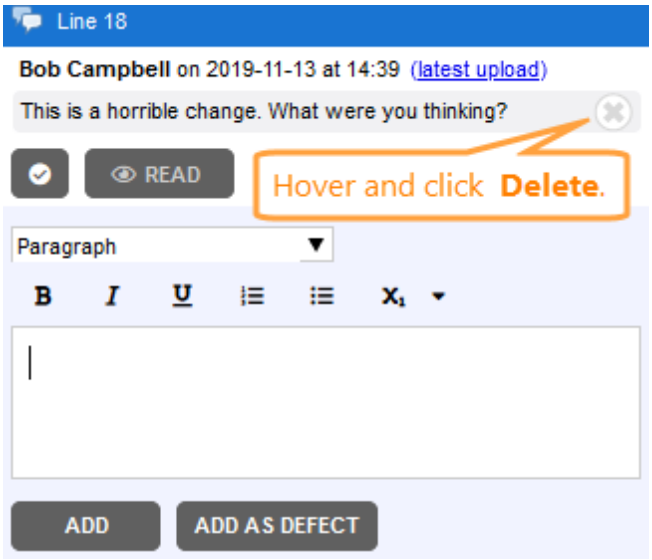



## Delete Comments

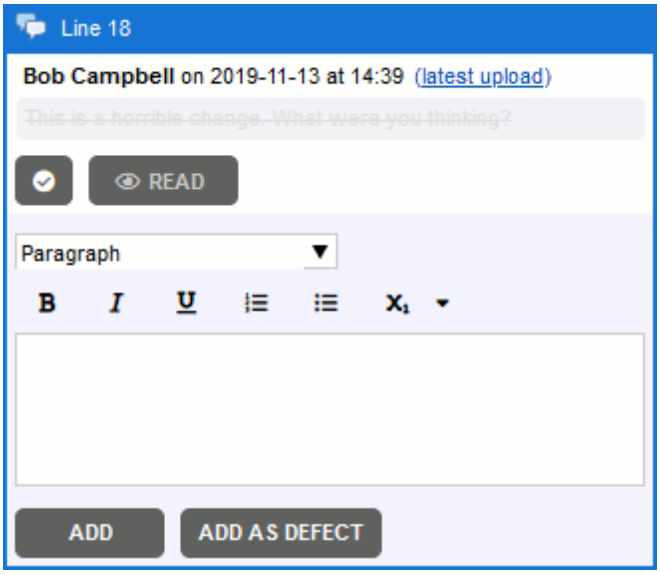
People have a knack for typing things that for various reasons they should not. Afterwards, they may want to discard their harsh, erroneous or otherwise inappropriate statements.

To discard a comment:

Mouse-over the specific comment you would like to discard and click **Delete**.




Further behaviour depends on the the Allow to edit/delete comments 186 server option. By default comment deletion is disabled in order to keep all comments intact for auditing purposes. In this case the comment will not be actually deleted, but it will be displayed in a much more difficult to read manner. This styling is meant to deter the casual reader from actually reading the text.

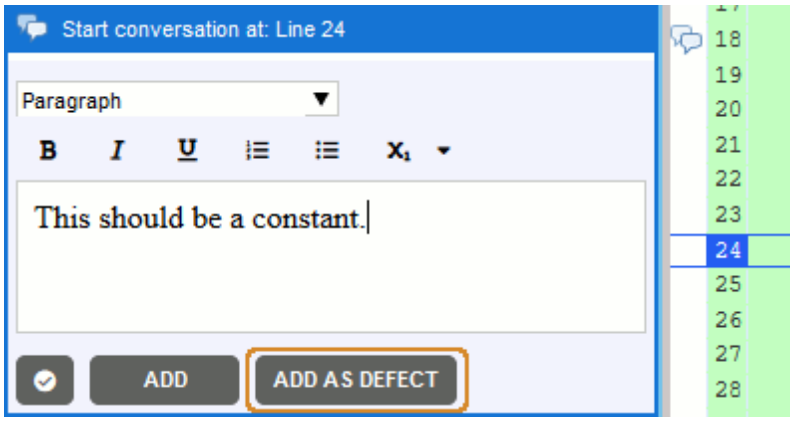



When the Allow to edit/delete comments 186 option is enabled, a confirmation dialog is displayed and on agreement the comment is deleted permanently. Comment could be deleted unless it is followed by another participant's comment.

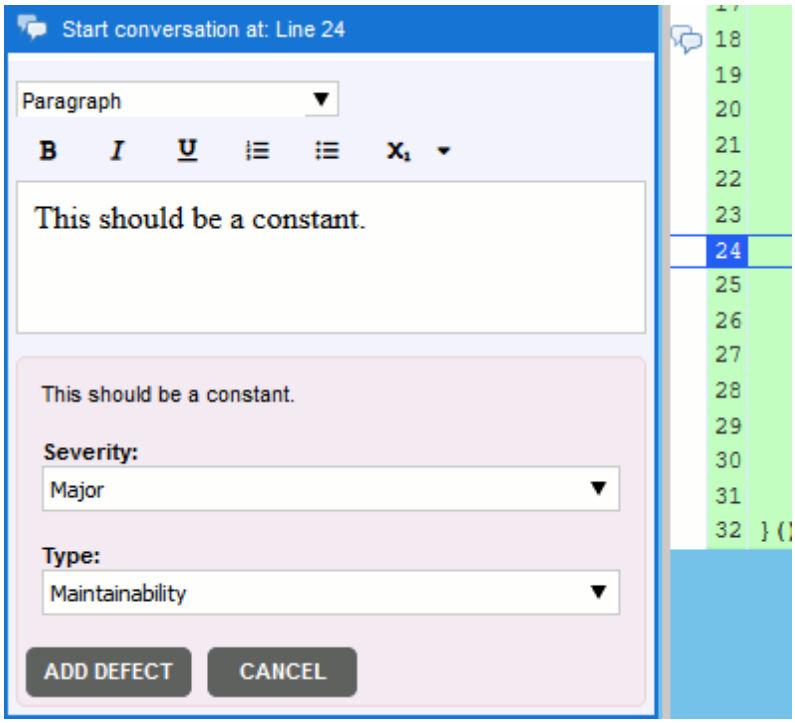
You cannot undo discarding/deleting a comment. System administrators are allowed to discard any comments. Regular users are allowed to discard their own comments.

## Create Defects -- indicate that something is wrong

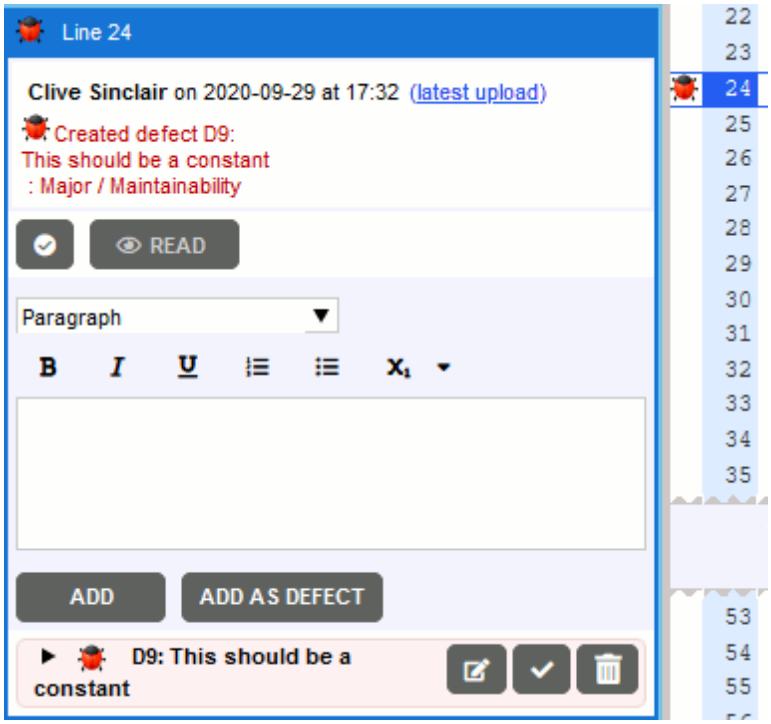
When you want to indicate that there is a problem that needs to be fixed -- not just more chat -- you open a defect. Do this using the **Add as defect** button below the comment field:




Besides the comment text, defects can have any number of additional fields. These are all completely configurable 256 by the system administrator and can also vary depending on the review workflow 338 that was originally selected for the review. Because this is completely configurable, this manual cannot say exactly what the fields will be or what they mean; ask your administrator for details. It is common to see fields like severity, type, checklist item, and phase-injected.




Once the defect is created, it is displayed in the Defect Log[354] and Review Material[357] sections of the Review Summary Screen[345] and in the Chat section of current file. The defect conversation is threaded. Open defects are denoted by red icon :



Every defect is given a unique number, on the server; in the example above the number is D9. There will only ever be a single defect numbered D9, no matter how many defects, documents or reviews exist on the server.

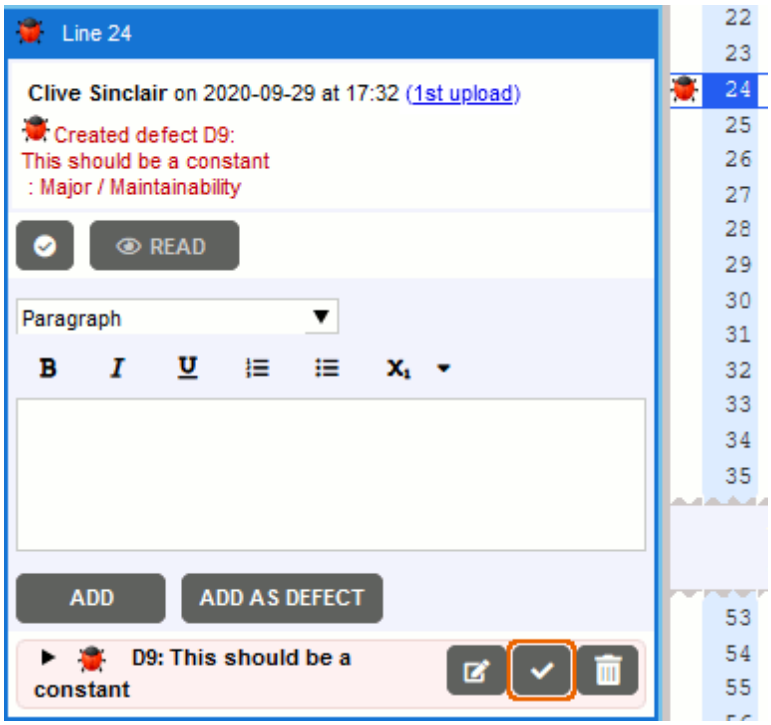
You can have any number of defects on a single line; all will be logged into the list beneath the chat area.

> **Warning:** The word "defect" has many connotations that are inappropriate for peer review. This does not mean the problem will be mirrored in an external issue-tracking system, and it does not necessarily mean it was a bug! Even "bad documentation" can be a defect.
>
> A "defect" is just a way of identifying something that needs to be fixed.
>
> Moreover, if the word "defect" has a negative connotation in your environment, your Collaborator administrator can change it to another term[184].

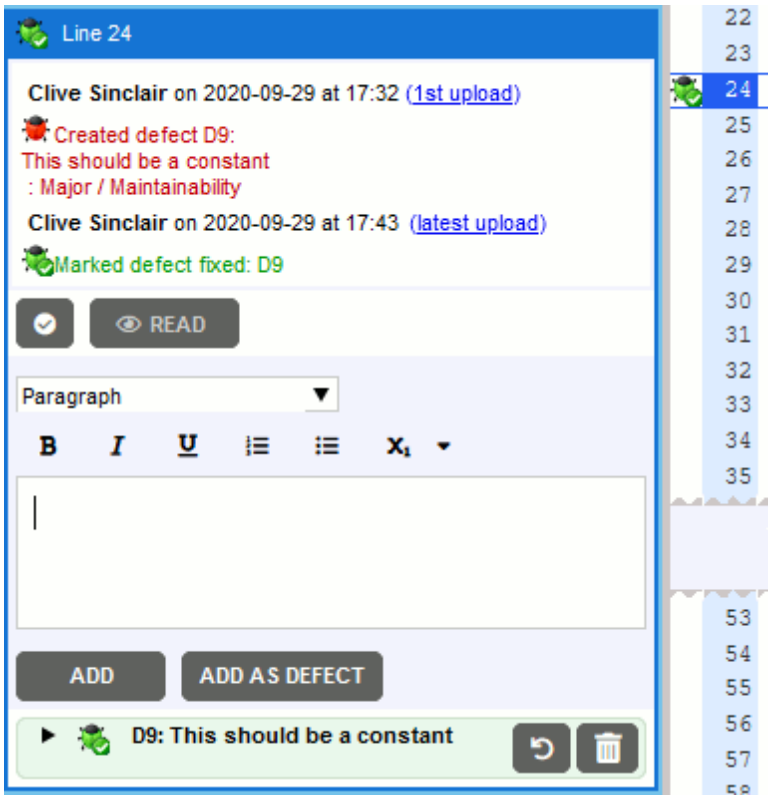
## Verify that defects have been fixed

Later, after the author has attempted to fix the defects and has uploaded the new files to the review, the reviewers will verify that the fixes do fix the defect and do not open more defects in the process.

When the fix is verified, locate the defect and click the **Mark Fixed** button in the defect pane.



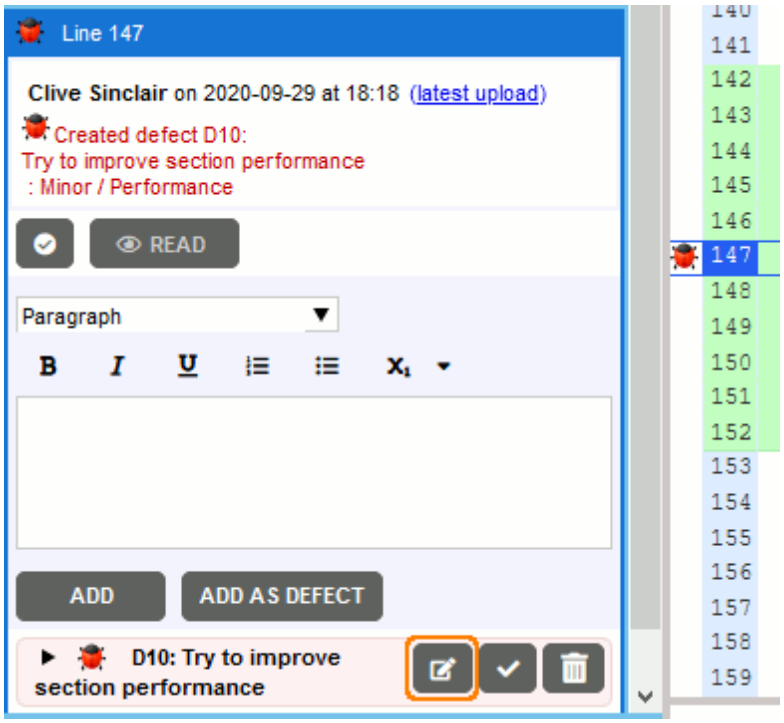

The defect icon will change to green to denote that the defect have been fixed. The new state of the defect will be reflected in the Defect Log[354] and Review Material[357] sections of the Review Summary Screen[345] and in the Chat section of current file.



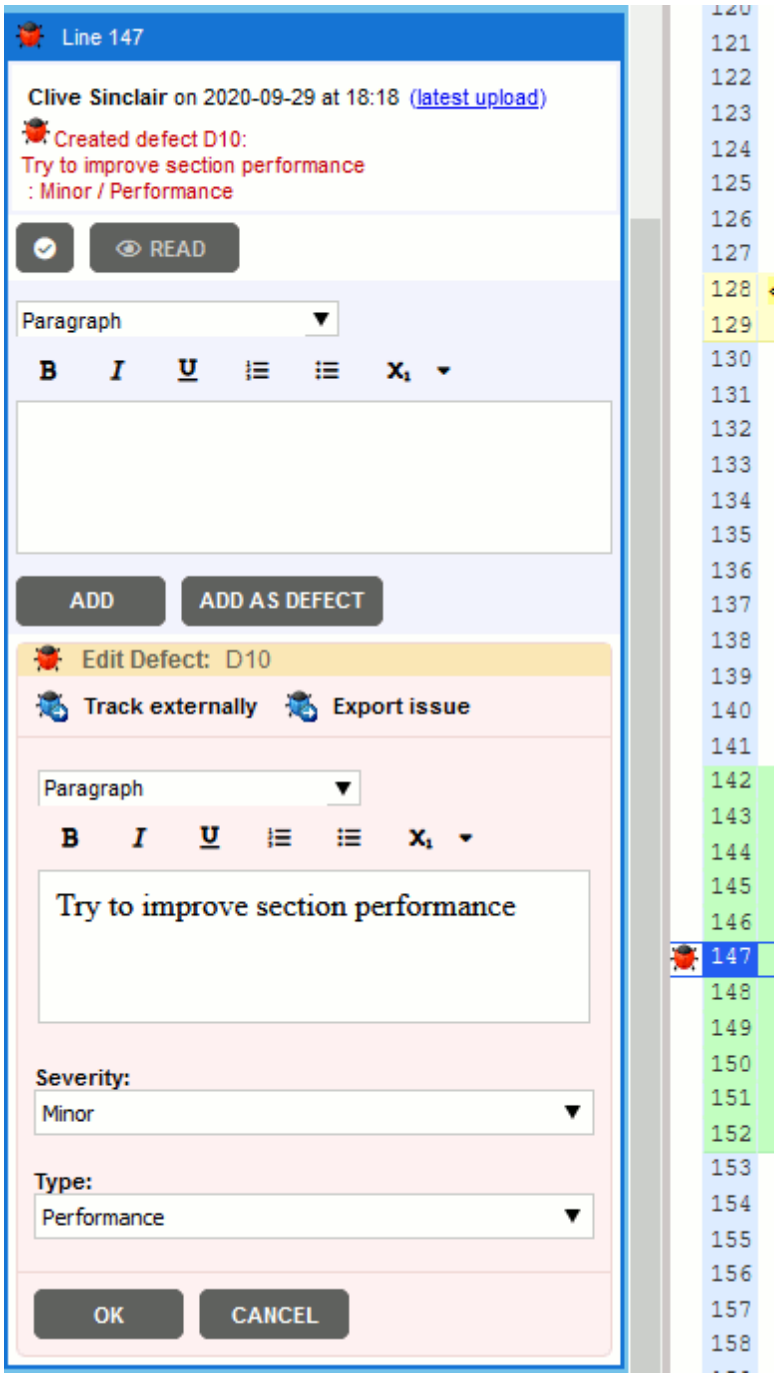

If the defect has re-appeared again in some subsequent revisions, you can reopen it by pressing the **Reopen** button in the defect pane.

## Modify defect information

To modify defect information click the **Edit** button in the defect pane.




In the edit mode, you can change defect description and modify other defect custom fields. Also you can move the defect to external issue tracker (see next section for details on this). All of these activities will be logged into the chat conversation area as an audit trail of what happened.



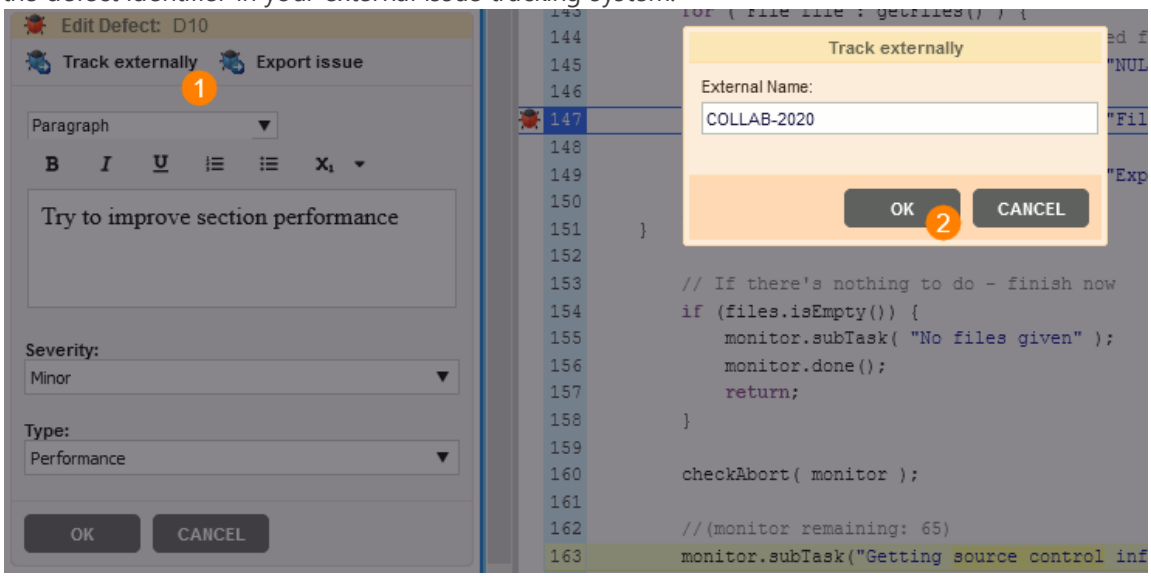

## Externalize defects

Sometimes you decide that a defect should be fixed at a later date. You do not want to mark it fixed because it is not fixed yet. But you do not want to delete it either because it is still a defect.

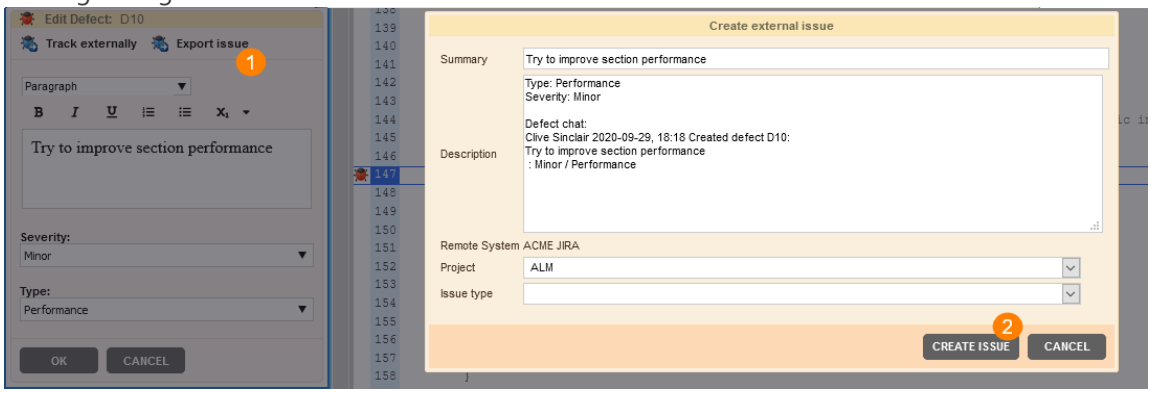
In this case you can move the defect to the external issue-tracking system and mark the defect as external. This special state tells everyone else that (a) the review can complete but (b) the problem was not fixed during the review and (c) how to find the external issue for further auditing.

You may either create the issue (ticket, work item) manually and link it afterwards, or create new item directly from Collaborator and link it automatically. The latter requires, that your Collaborator administrators setup and configure issue-tracking integration 887.

- To link the defect to some existing issue (ticket, work item), press **Track externally** and specify the defect identifier in your external issue tracking system.

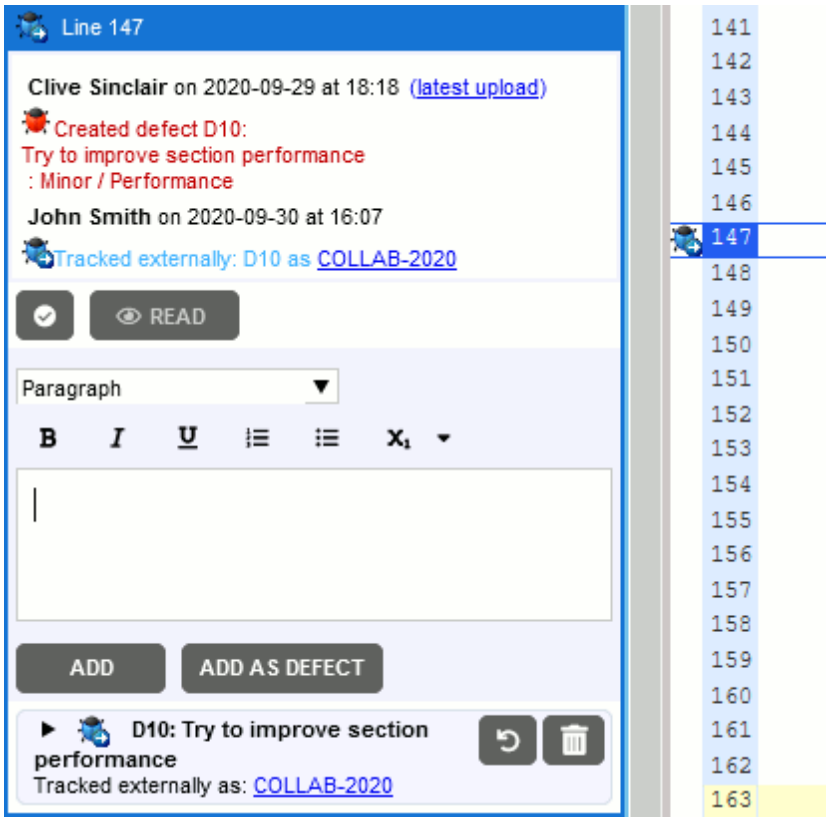



- To create a new issue (ticket, work item), press **Export issue** and specify the defect details in the ensuing dialog.




Any of these actions changes the state of the defect from "Open" to "Tracked Externally".

The defect icon will change to blue to denote that the defect have been tracked externally. The new state of the defect will be reflected in the Defect Log 354 and Review Material 357 sections of the Review Summary Screen 345 and in the Chat section of current file.

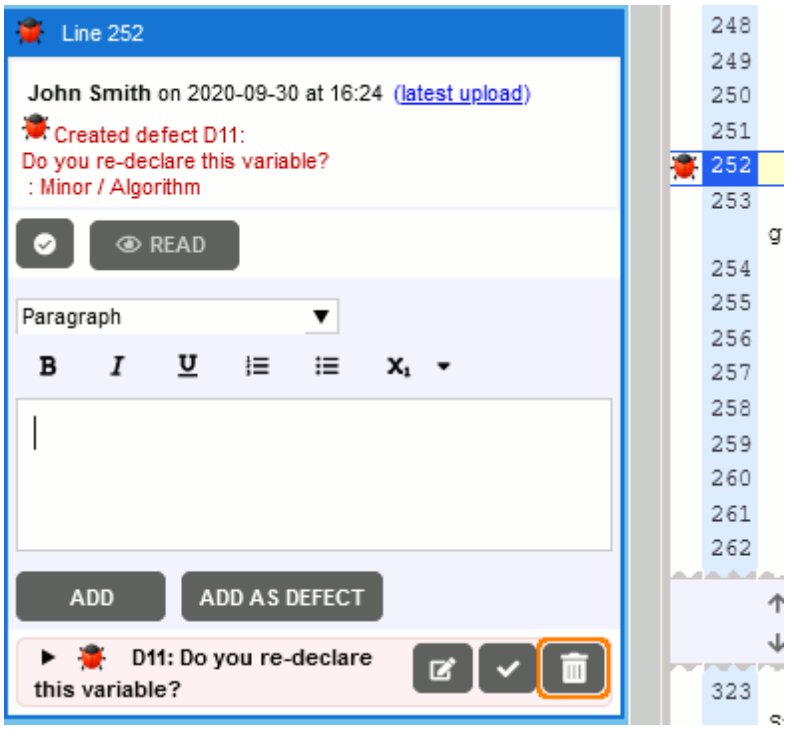



## Delete defects

Sometimes people could create defects by mistake and may want to delete them afterwards.

> **Warning:** It is tempting to delete a defect once the author has corrected the problem, but this is the wrong thing to do. You want to keep the defect record around, just mark it "fixed".
>
> Delete a defect only if it turns out that it really was not a defect at all.

To delete a defect click the **Delete** button in the defect pane.




### 4.3.7 User Mentioning

## About

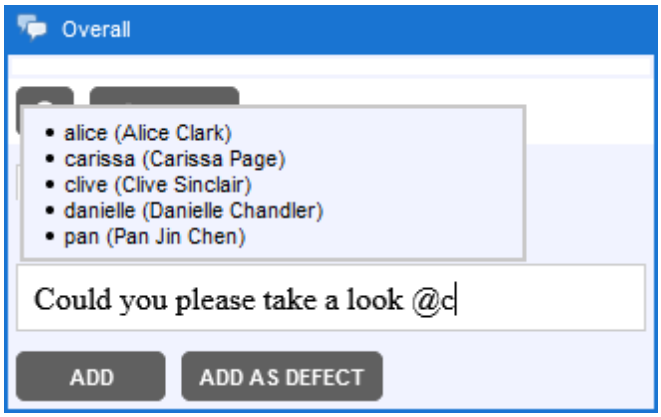
User mentioning is currently supported for the following fields of Web Client:

- comments and defects in the Review Screen 345,
- comments and defects in the Diff Viewer 364,

Other fields of Web Client and other clients do not send any notifications when mentioning users.

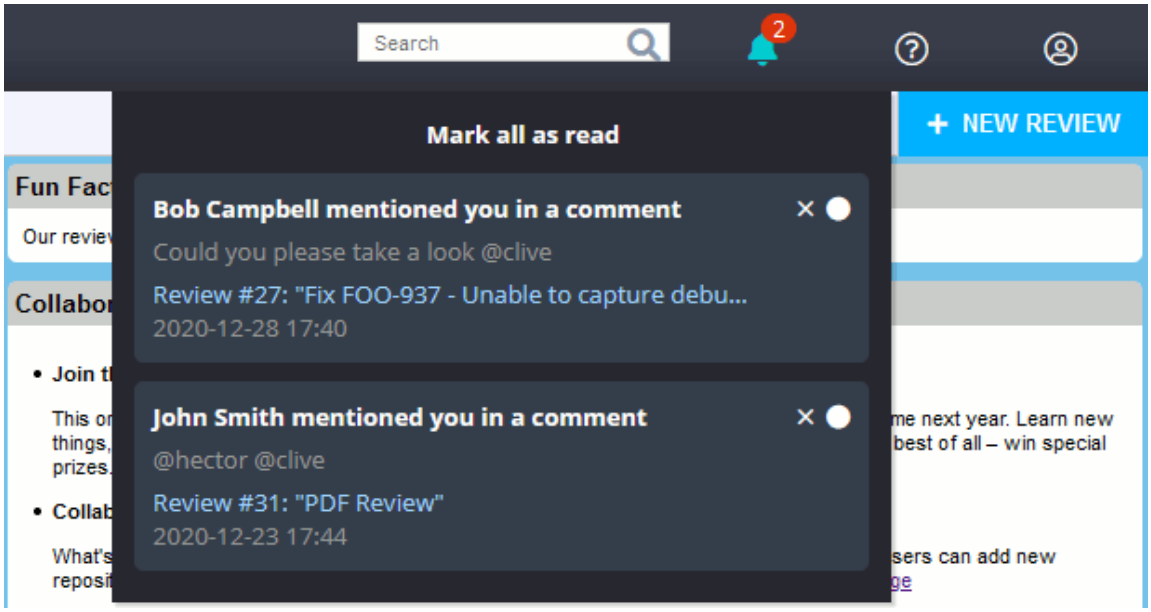
## How it works

To mention a user just type the @ character followed by user login or display name.




The editor will display a drop-down with user suggestions, which filters-out the available users as you type. You can also select the desired user using mouse or using Up, Down and Enter keys.

When you submit your comment or defect, Collaborator sends an e-mail and in-app notifications, so that mentioned users may check the corresponding comment or defect.




## Technical details and limitations

- The list of suggested users contains only those users that are allowed to access the current review according to the Restrict access to review[188] setting.
- Users are identified by their login (as it is unique). User display names are listed to simplify search and improve readability.

- User name and login search is case-insensitive.
- The list of suggested users is refreshing with each input character.
- The list of suggested users is limited by 20 first users that match the input.
- User mentioning is only processed when a blank space character precedes the @ character. For example: *test @username1*. It is not processed when the @ character occurs within the string. For example: *user1@mail.com*

### 4.3.8 Rich-text and Markdown Formatting

## About

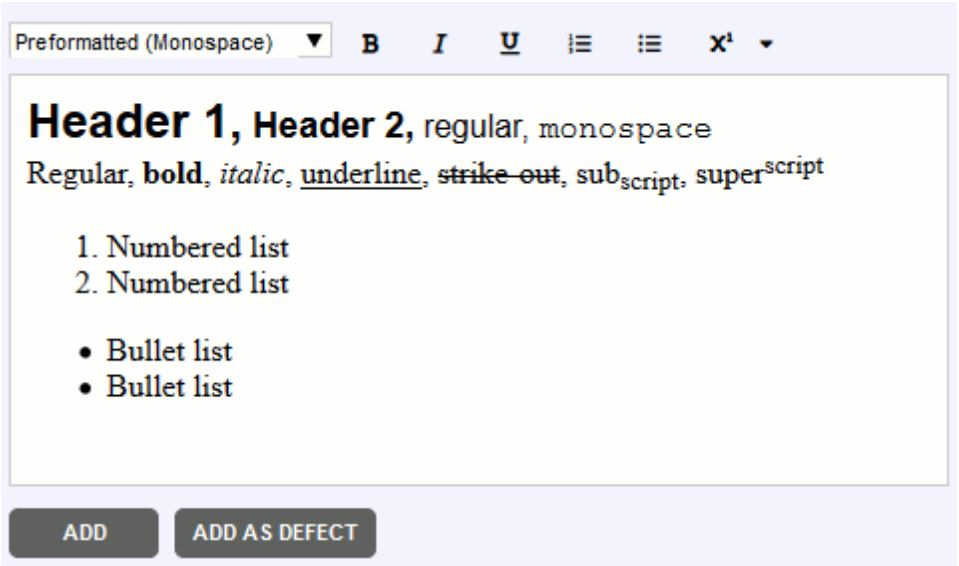
Rich-text and Markdown formatting is currently supported for the following fields of Web Client:

- comments and defects in the Review Screen 345,
- comments and defects in the Diff Viewer 364,
- comments and defects in Reports 467.

Other clients and exported reports display comments and defects using plain-text representation.

## Rich-text formatting

All text fields that support rich-text have a toolbar that allows specifying or changing formatting.



| | |
|---|---|
| Paragraph ▼ | Assigns a pre-defined style to the current paragraph or to selected paragraphs. The following styles are available:<br>• Header 1<br>• Header 2<br>• Paragraph,<br>• Preformatted. |
| **B** | Applies, or clears bold formatting. |
| *I* | Applies, or clears italic formatting. |
| U | Applies, or clears underlined formatting. |
| ⁝≡ | Applies, or clears numbered list formatting. |
| ⁝≡ | Applies, or clears bullet list formatting. |
| $X_1$ | Applies, or clears subscript formatting. |
| $X^1$ | Applies, or clears superscript formatting. |
| ~~S~~ | Applies, or clears strike-out formatting. |

## Markdown formatting

Markdown uses special conventions to create formatted text from plain-text.

To apply Markdown formatting for new comments and defects, your Collaborator administrator should turn on the "Enable Markdown in comments 185" setting. Existing comments and defects will retain their current formatting.

Below is a list of Markdown elements supported by Collaborator.

| | |
|---|---|
| Headers | Start a line with a hash character # to apply header formatting. Up to six levels of headings are supported.<br>`# Header 1`<br>`## Header 2`<br>`### Header 3`<br>`#### Header 4`<br>`##### Header 5` |

| | |
|---|---|
| | `###### Header 6` |
| Bold | Wrap the text with double asterisk * characters.<br>`**This is bold text**` |
| Italic | Wrap the text with a single asterisk * character.<br>`*This is italic text*` |
| Inline code | Wrap the text with a single backquote ` character.<br>`This is regular text `and this is inline code`` |
| Code block | Prefix your code with four space characters.<br>`    main( ) {`<br>`            printf("hello, world\n");`<br>`    }` |
| Link | Specify link text wrapped in square brackets [] and then specify link URL wrapped in round brackets ().<br>`[Our site](http://example.com)` |
| Bullet list | Prefix each line with a single asterisk *, minus - or plus + character.<br>`* Bullet item 1`<br>`* Bullet item 2` |
| Horizontal line | Insert three asterisk *or dash - characters.<br>`***`<br>`---` |

To enter specific or literal characters ignoring Markdown formatting, prefix that character with a backslash \ character.

## Technical details and limitations

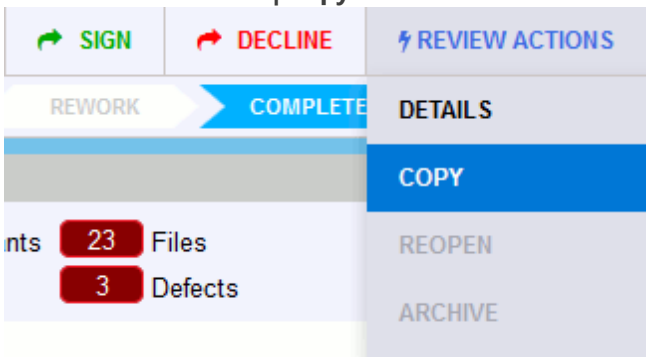
- Message length is limited to 16000 characters (including html tags). Longer messages will be truncated (replaced with "... ");
- Text from clipboard will be pasted as plain text.

### 4.3.9 Copying Previous Review

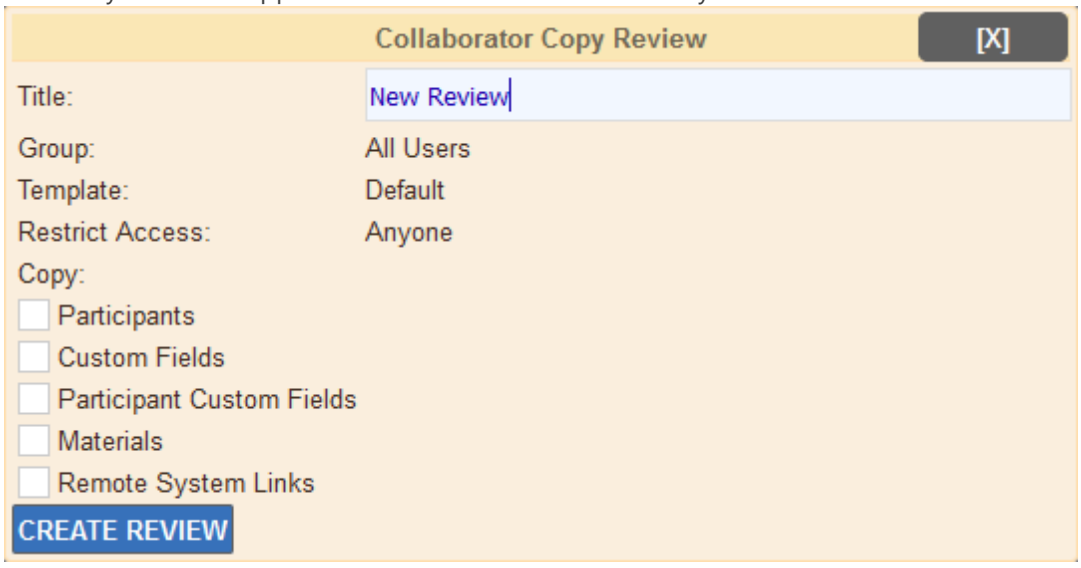
You can create new reviews on the basis of a previous review. This allows to retain some information from the previous review (list of participants, custom field values, review materials and so forth).

To copy a review:

1. Open the review you want to copy in the Review Summary Screen 345.
2. Select **Review Actions | Copy** in review header:




3. In the ensuing pop-up dialog, specify title for the new review and select what information to copy from the previous review: Participants, Custom Fields, Participant Custom Fields, Materials, Remote System Links.
   **Note:** When copying remote system links Collaborator does not copy links to closed items. Also, this functionality acts independently from the template's Automatically Add Remote System Links 246 settings which parse the review's title, custom fields, comments for links to remote systems and append them to the review's Remote System Links section.



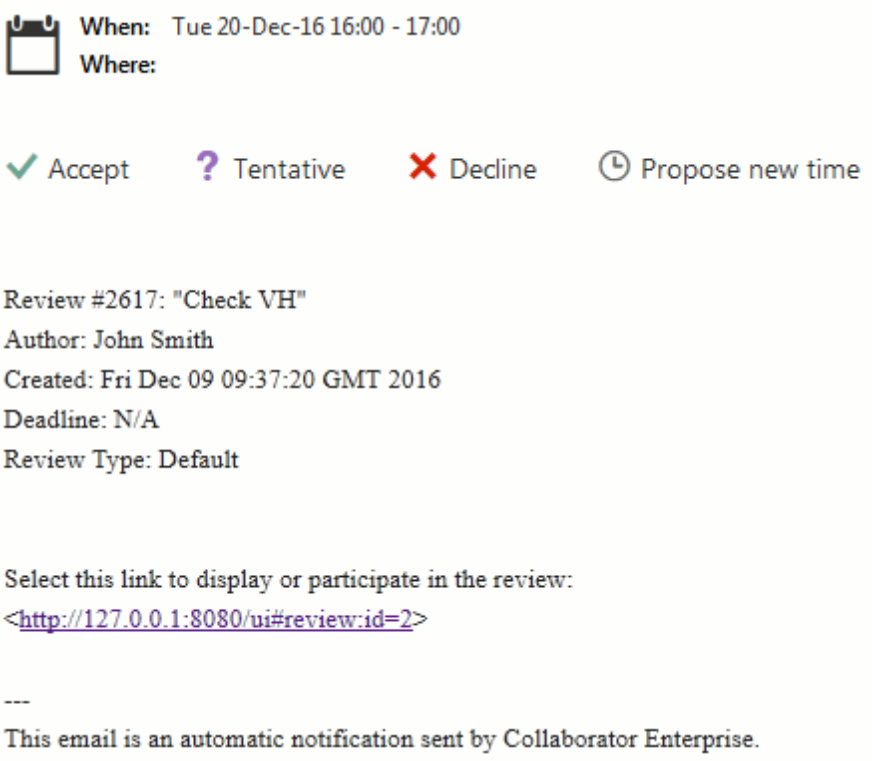

4. Click **Create Review**.

Once created, your existing subscriptions will apply to a new review. This may result in new participants be added to the review which were not part of the original review.

Alternatively, you can copy a previous review using command-line interface (`ccollab admin review copy`), using the JSON API (`ReviewService.copy`) or from the Review Summary Screen[591] of Visual Studio Extension.

### 4.3.10 Sending Calendar Notifications

Calendar notifications may be useful for scheduling formal meetings on a particular review. Anyone of review participants may create an iCalendar invitation from the Review Summary Screen (see below). Once created, all participants of that review will receive email messages with a meeting proposal on specific date and time.




*Calendar invitation in Outlook*

## Technical Details

Calendar invitations are sent via email. Therefore, your Collaborator administrator should configure[199] SMTP server settings and enable email notifications.

Calendar invitations use iCalendar format. To process them correctly, your email client should support this format. Otherwise, invitation will be interpreted as ordinary file attachment.

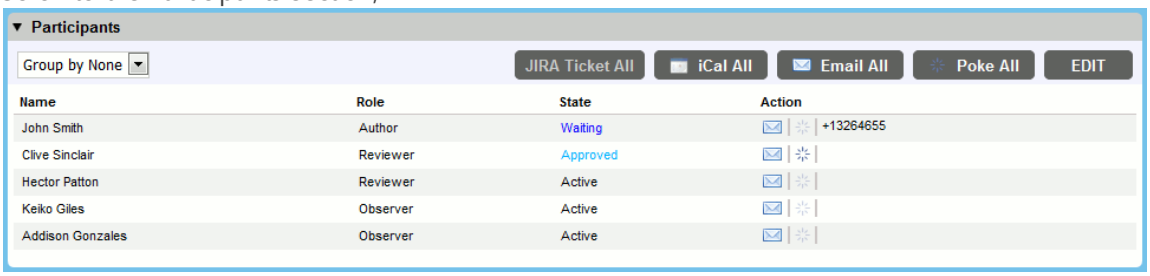
By default, the user who has created the invitation is assigned as the organizer of the meeting. This behaviour can be changed by the *Use iCal organizer* 202 setting.

The text content of Calendar notifications is defined in the List Calendar Invite notification template 288.

## Creating Calendar Invitations

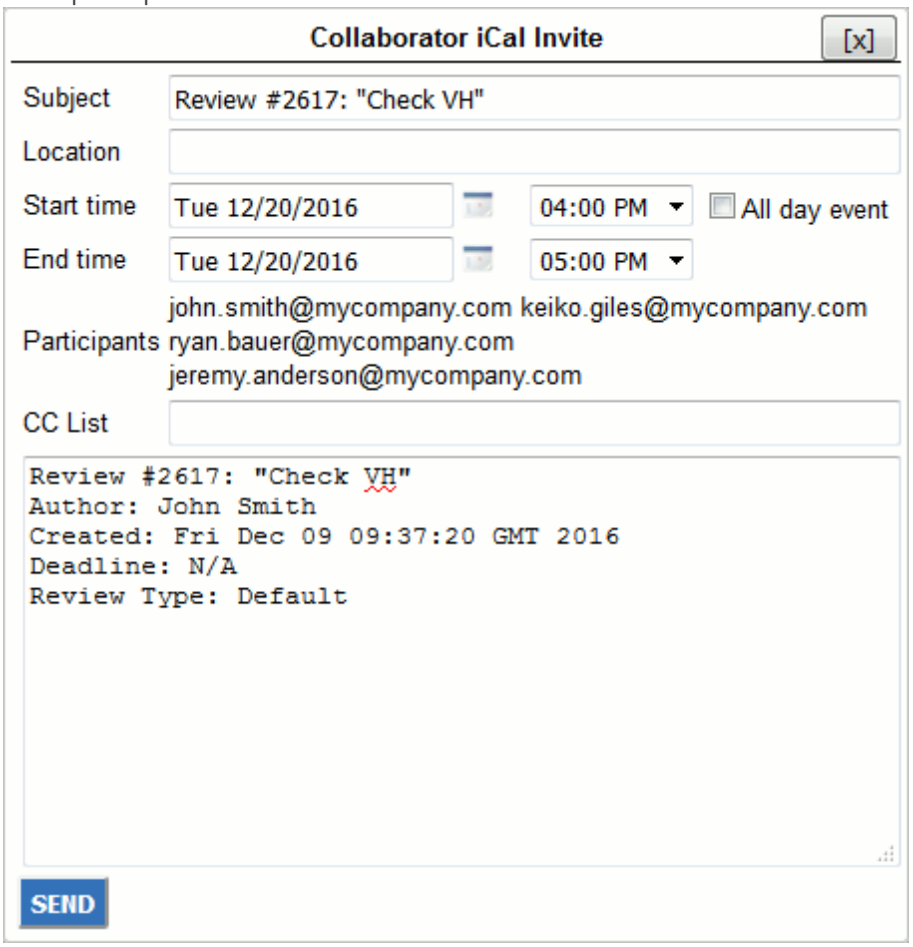
To create Calendar notifications:

1. Open desired review in the Review Summary Screen 352,
2. Scroll to the Participants section,




3. Click the "iCal All" button in the section's upper-left corner,

4. In the ensuing Collaborator iCal Invite dialog:
   Specify date and time of event, (optionally) specify location, modify event description, add other participants and click Send.




## 4.3.11 Hiding Files From Review

Changelists may include files that do not necessarily need to be reviewed. However, since these files are still part of the atomic changelist, these files cannot be deleted from the review. In this case review authors (or all review participants, depending on the *"Who can hide files"*193 setting) can mark unneeded files as hidden.

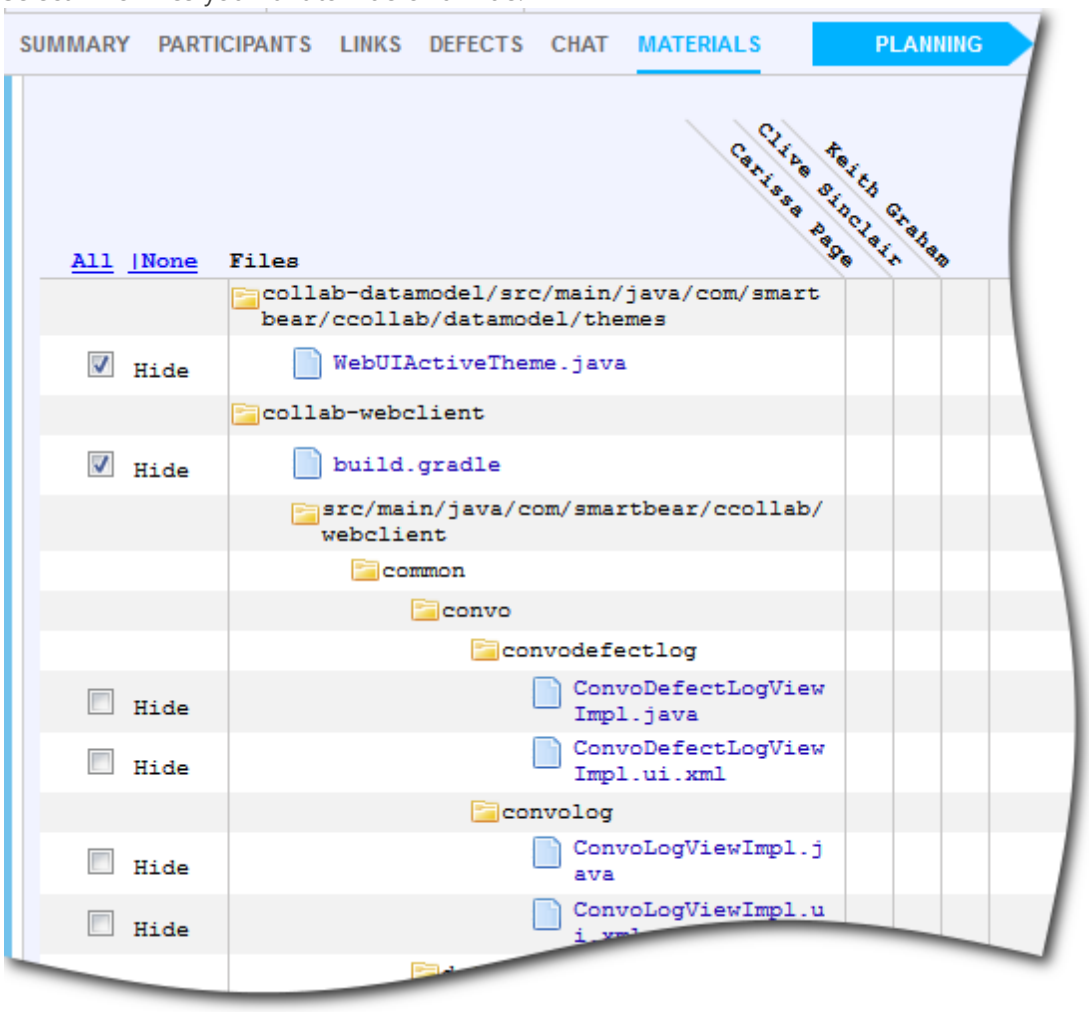
- Hidden files are not displayed in Diff Viewer364. So reviewers will not be able to view their content, create comments of defects to these files.
- Hidden files are not listed in Overlay view mode of Review Summary Screen. In Separate view mode, hidden files are displayed, but will have a special mark.
- Hiding a particular revision of a file, will hide all other revisions of this file as well.

By default, you can hide files only in the Planning phase of the review. However, Collaborator administrators could enable this feature for the Annotating phase or for any other phase via the *"What phases can you hide/unhide files"*193 setting.

If a file has any comment or defect, it cannot be hidden.

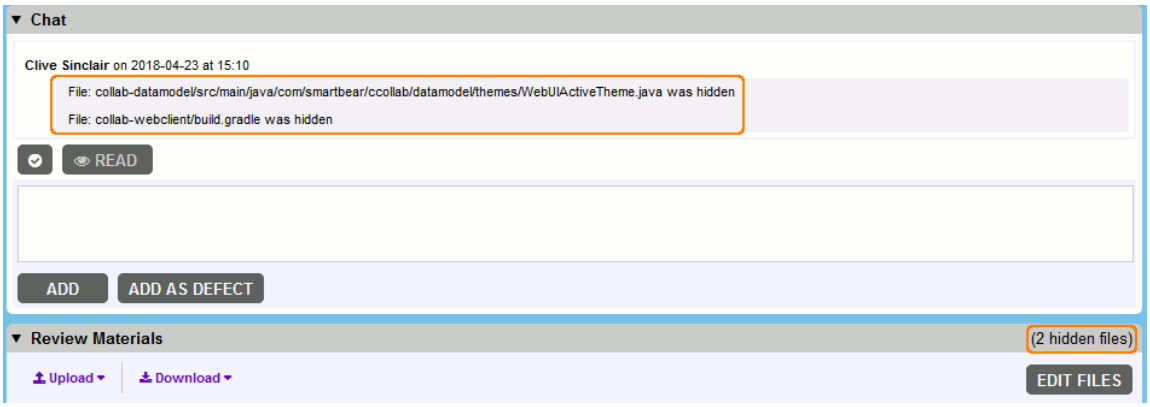
**To hide and unhide files from the review:**

1. Open the review in the Review Summary Screen 345.
2. Scroll to the Review Materials section.
3. Click **Edit Files** button. This will display Hide check boxes in front of file names.
4. Select which files you want to hide or unhide.




5. Click **Done Editing Files** when finished.

Once a file is hidden or unhidden, a comment is made in the Chat section which says that a particular file was excluded or included in the review. Also the counter of hidden files is incremented or decremented, respectively.

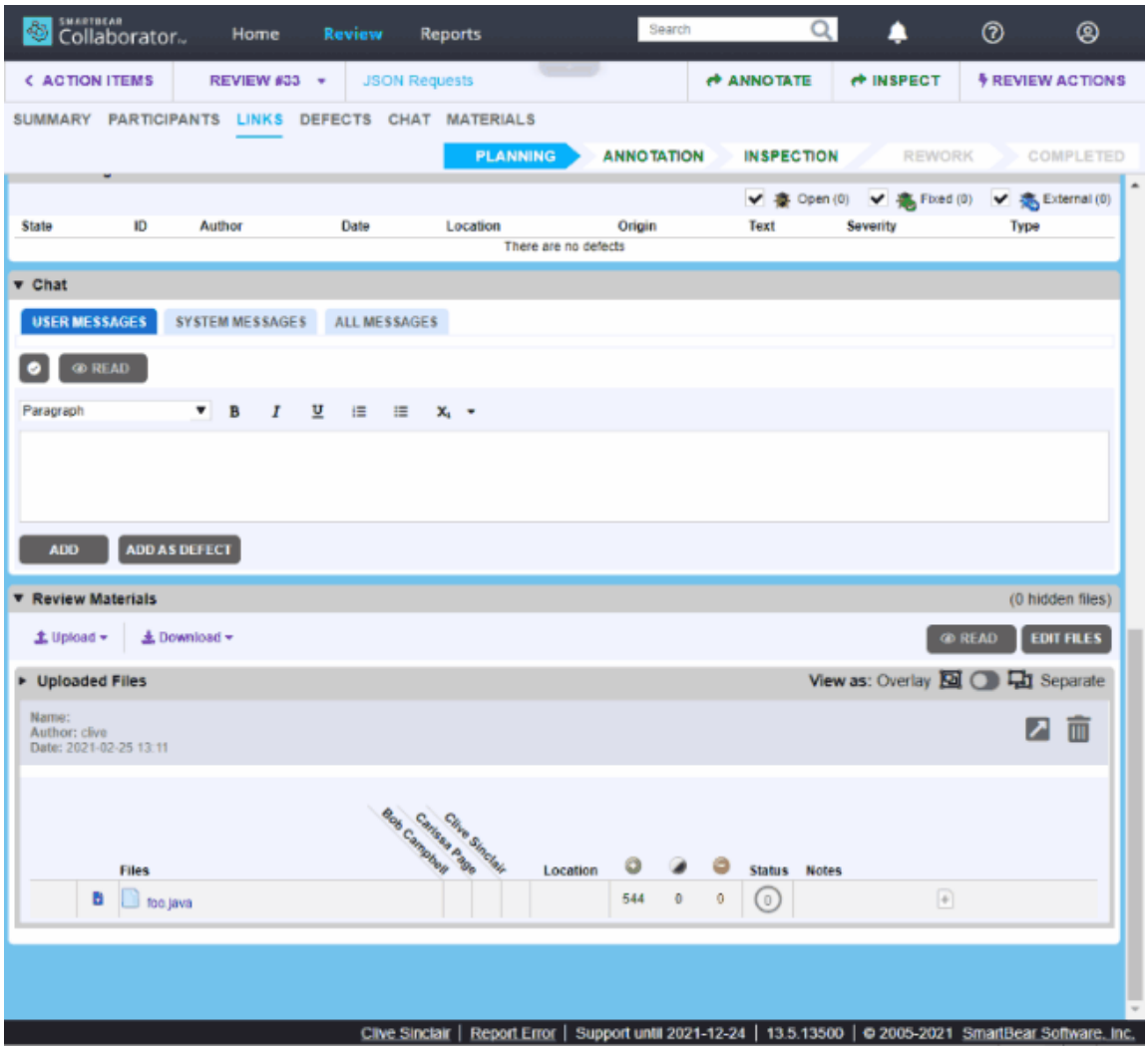



Once a subsequent changelist is uploaded and a hidden file was changed, it may remain hidden (default) or become visible depending on the *"When a hidden file changes"*193 setting.

## 4.3.12 Uploading new revision under a different file name

### Overview

Collaborator can associate a differently named file with a file that is already under review. That is, you can have different file names for the same document. For example, you can upload a file named - *my_file1.doc* and then upload its revised version as *my_file2.doc*, or as *my_file_Johns_version.doc*, or whatever other name.




Collaborator would process such files as subsequent revisions of the same file (will render differences in Diff Viewer, promote comments and defects, calculate metrics and so forth).

This feature is intended for manually uploaded files only, it is not available for files uploaded via source control systems.

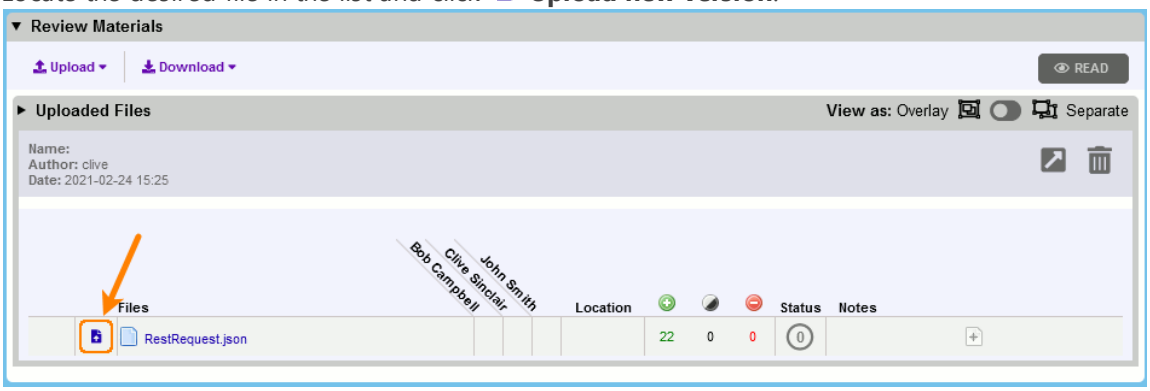
## How to upload a new revision under a different file name

1. Log in to Web Client[313].

2. Open the desired review in the Review Screen[345] and scroll to the Review Materials[357] section.

3. Enable the **Overlay** view option.

4. Locate the desired file in the list and click **Upload new version**.

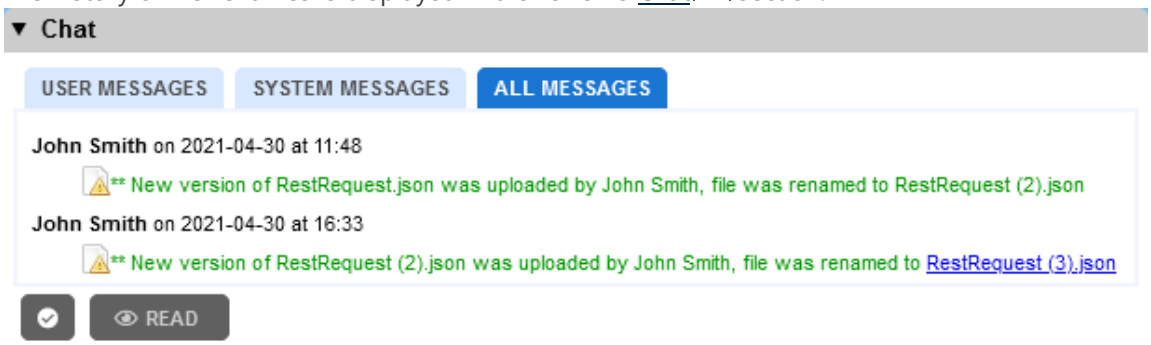



*Uploading new revision*

5. In the subsequent Open File dialog, select the new file revision and click **Open**, then click **Upload**.

## Technical details and limitations

- This feature is intended for manually uploaded files only, it is not available for files uploaded via source control systems.
- New revision can have arbitrary name, but it should be of the same file type (extension) as the original file.
- The **Upload new version** button is only available in **Overlay** view.
- In **Overlay** view, file is listed under its latest file name.
- In **Separate** view, each upload is listed under its original file name.
- The history of file renames is displayed in the review's Chat 356 section.




- Collaborator supports chain renames: that is uploading of several files with different names as subsequent revisions of the same file. For example, `foo-v1`, renamed to `foo-v2`, then renamed to `foo-v3` and so forth.

- Collaborator will not allow uploading a file as new revision if a file with the same file name was already uploaded to the current review (either as current or as previous revision). That is, you cannot upload two *my_file1.doc* files to the same review.

## 4.4 Searching & Reporting

Various reports[467] and a sophisticated search[458] engine help you track reviews completed, reviews in progress, and changes in version control that have not been reviewed.

Collaborator also creates metrics[974] reports for things like time spent in reviews, time between phases, kLOC/hour reviewed, defects per kLOC, time per defect, and many others.

Reports can be viewed on-line or exported in a variety of formats. Reports can also be saved and e-mailed.

### In This Section

- Searching From Web UI[458]
  Describes different ways to find a review from the Web Client.
- Searching Directly From Web Browser[463]
  Describes hot to search for Collaborator reviews directly from the browser's address bar or search box.
- Reporting[467]
  Describes customizable pre-built reports.

### 4.4.1 Searching From Web UI

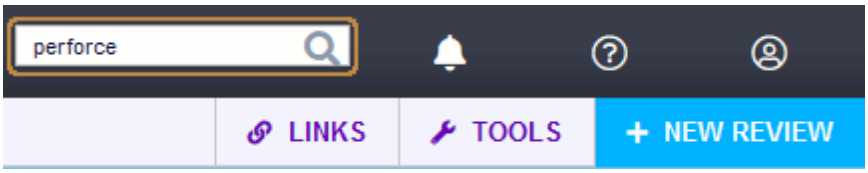
Collaborator has a sophisticated search facility allowing you to find reviews in many ways. Uses include:

- Find reviews by review ID.
- Find reviews with substrings in title and custom fields[256].
- Find reviews with specific participants[338].
- Find reviews of particular changelists by check-in comment or changelist ID.
- Find reviews of certain files by file path.
- Find reviews having a substring in comment or defect text.

### Starting a Search

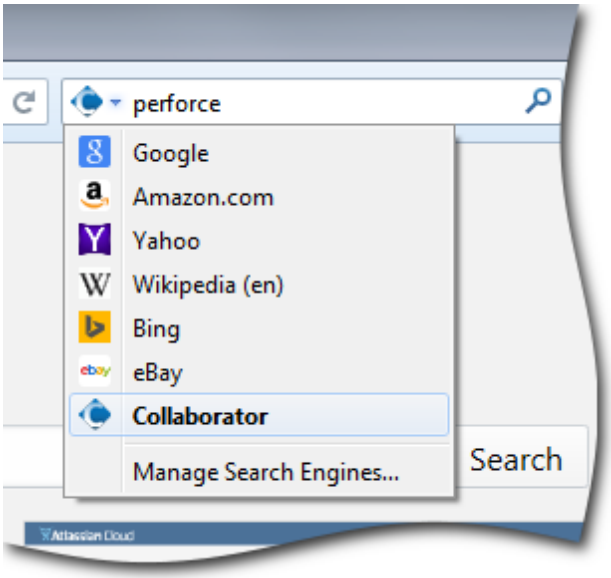
To start a search you can type the search term into any of the following:

- The search box at the left side of the menubar:




*Entering search term in Web User Interface*

- The address bar or search box of your browser (if your Collaborator server was added as browser's search provider[463]):



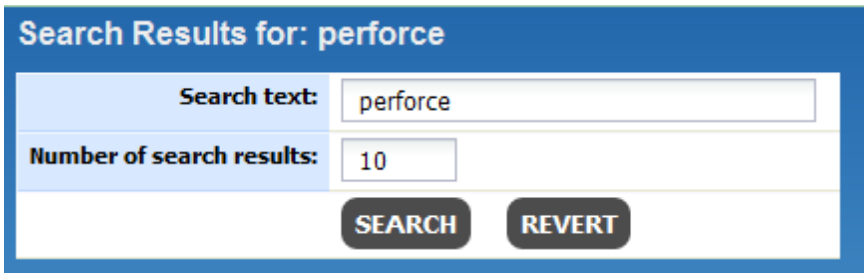

*Entering search term in Mozilla Firefox*

The system will automatically search in all the ways listed above.

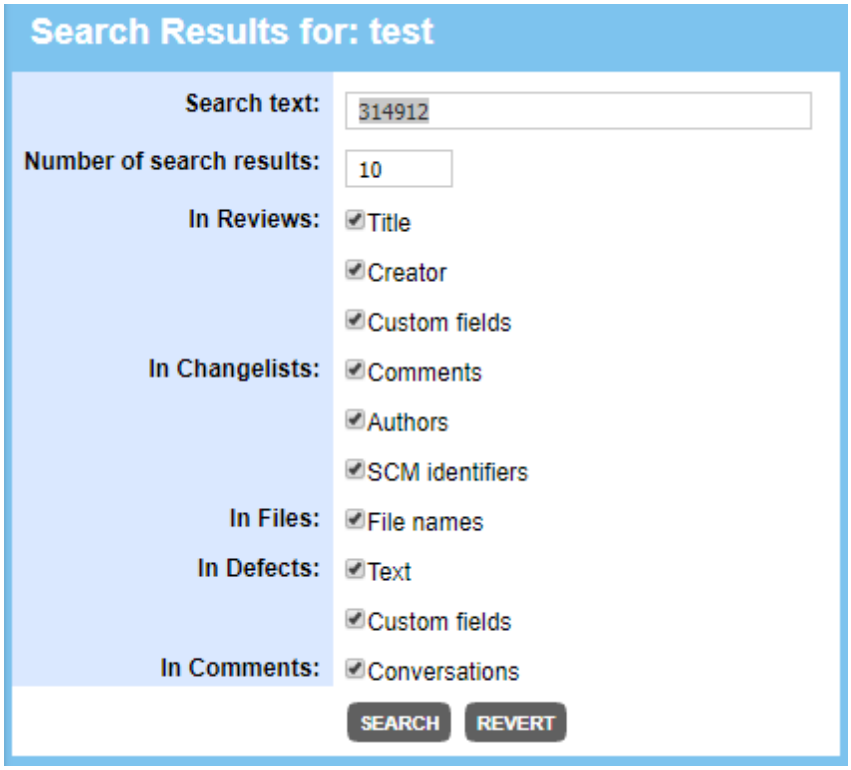
If you have entered the review ID number, you will automatically be taken to review. See Jumping to Reviews[462] section for details.

## Search Results

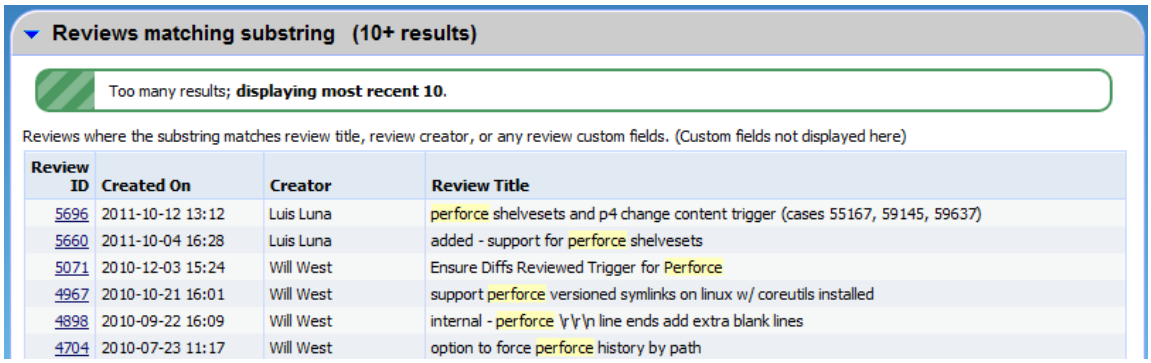
The number of displayed search results will be limited if the search produces too many results. The default number displayed will usually be "10". This can be changed in the form that will appear at the top of the search results. To display more or less than shown, change the value in the **Number of search results** field and click **Search**.




**Important:** On Oracle databases, Collaborator does not search the contents of custom fields by default (since this significantly reduces search performance). Instead, the search results page display additional fields that define in what areas to perform new search. In this panel you can enable searching in custom fields.




Because there are many types of searching, each result block is drawn separately. For example, below are the results of searching inside review title and custom field text:



**Reviews matching substring (10+ results)**

Too many results; **displaying most recent 10.**

Reviews where the substring matches review title, review creator, or any review custom fields. (Custom fields not displayed here)

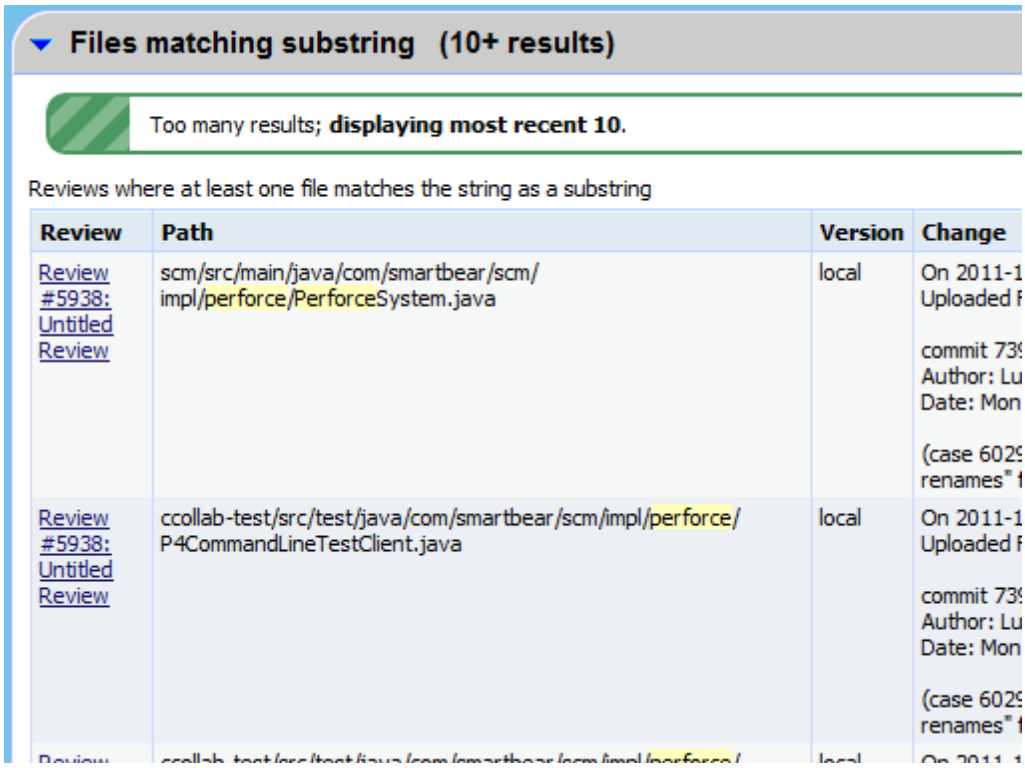
| Review ID | Created On | Creator | Review Title |
|---|---|---|---|
| 5696 | 2011-10-12 13:12 | Luis Luna | perforce shelvesets and p4 change content trigger (cases 55167, 59145, 59637) |
| 5660 | 2011-10-04 16:28 | Luis Luna | added - support for perforce shelvesets |
| 5071 | 2010-12-03 15:24 | Will West | Ensure Diffs Reviewed Trigger for Perforce |
| 4967 | 2010-10-21 16:01 | Will West | support perforce versioned symlinks on linux w/ coreutils installed |
| 4898 | 2010-09-22 16:09 | Will West | internal - perforce \r\r\n line ends add extra blank lines |
| 4704 | 2010-07-23 11:17 | Will West | option to force perforce history by path |

Note several features of the search results:

- Text at top of the block explains exactly what the block is searching on.
- Reviews are displayed in creation-date order, most recent first.
- Specific search term is highlighted in yellow.
- Jump to a review by clicking the ID link.
- When there are many results, a message appears near the top in green.

You may notice that some reviews do *not* have the yellow highlight. Why did these reviews match the search? This search looks not only at the review title and participants but also in all custom fields. Those fields are not, however, displayed in this view, so nothing can be highlighted. (Custom fields are not shown because there can be a large number of them which would make the search results difficult to read.)

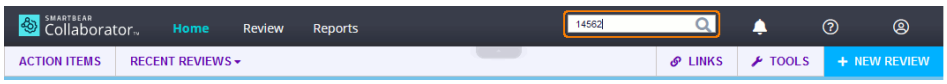
Here is another example where the block is searching over files present in the review:



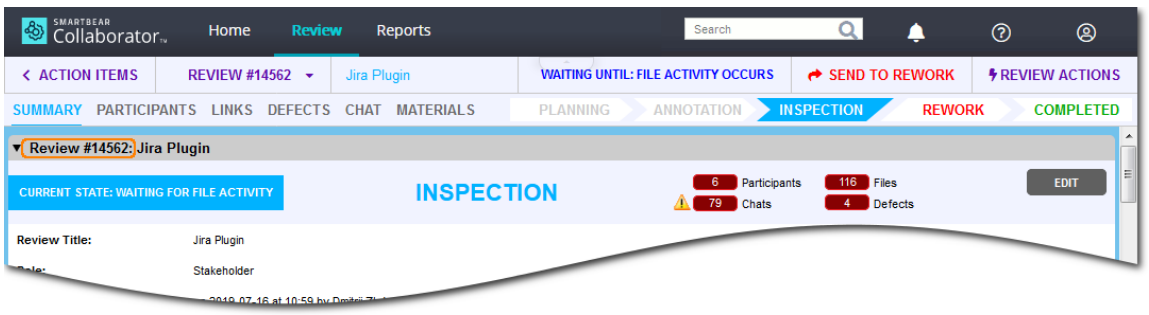

## Jumping to Reviews

You can jump to a review by entering the unique review ID number into the search box. If the number entered matches a review ID number, you will automatically be taken to review. This allows you to bypass search results and quickly find the searched review.

Enter the Review ID number into the search box on the Home page and press Enter:




The specified review will be displayed in the Review screen:



### 4.4.2 Searching Directly From Web Browser

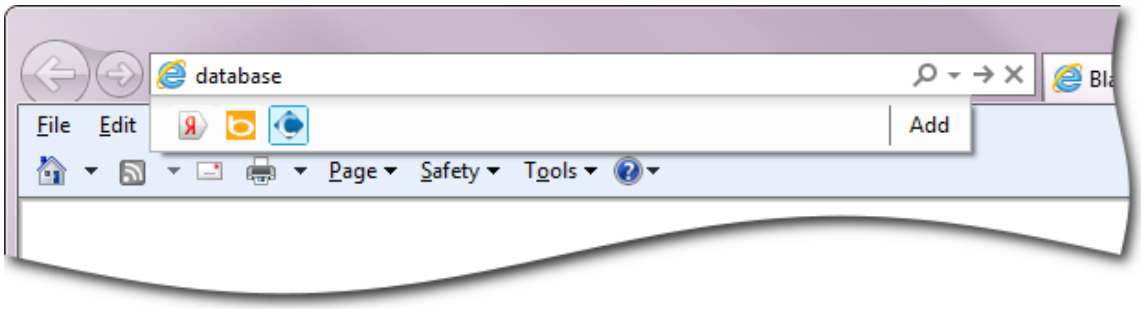
You can add Collaborator to the list of search engines of your browser. In this case you can search for reviews, participants, files and other directly from your browser's address bar or search box.

## How it works

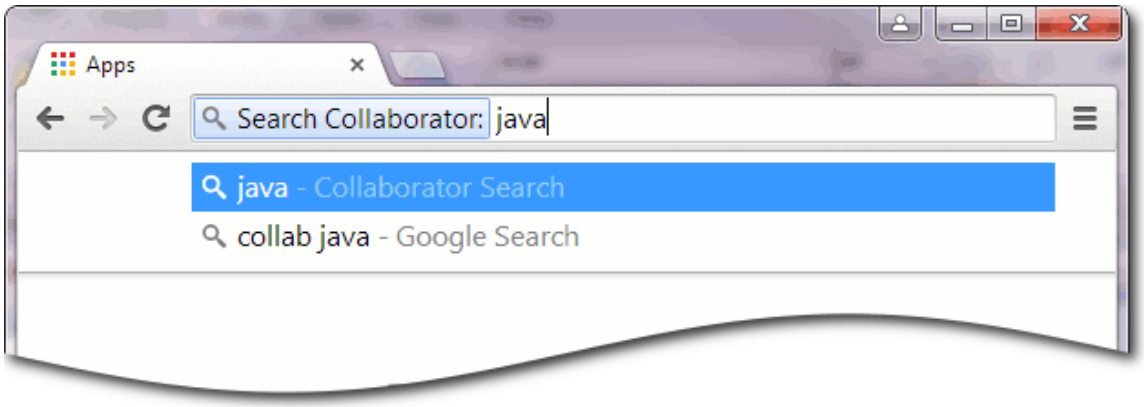
To perform a search in Collaborator from within the web browser, we need to add your Collaborator server as the browser's search provider. (See instructions below 463)

Once this is done, you can open any arbitrary web page (not necessary from Collaborator Web UI) and enter the desired search term in the browser's address bar or search box.

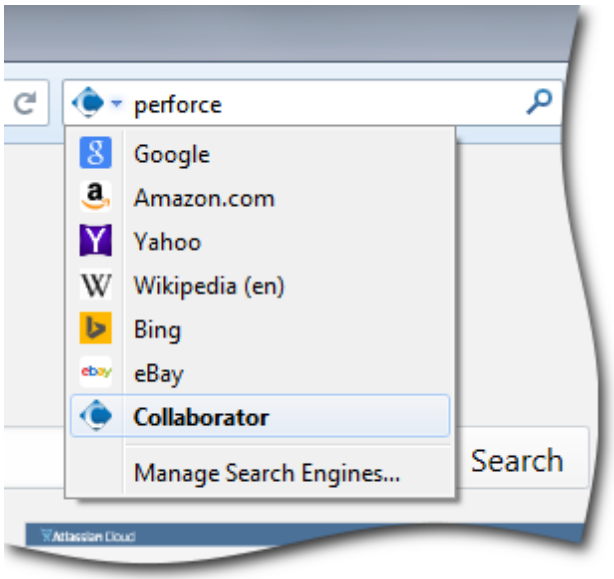
Then you need to select Collaborator as the search provider (if it is not set as the default provider) and press Search.




*Entering search term in Internet Explorer*

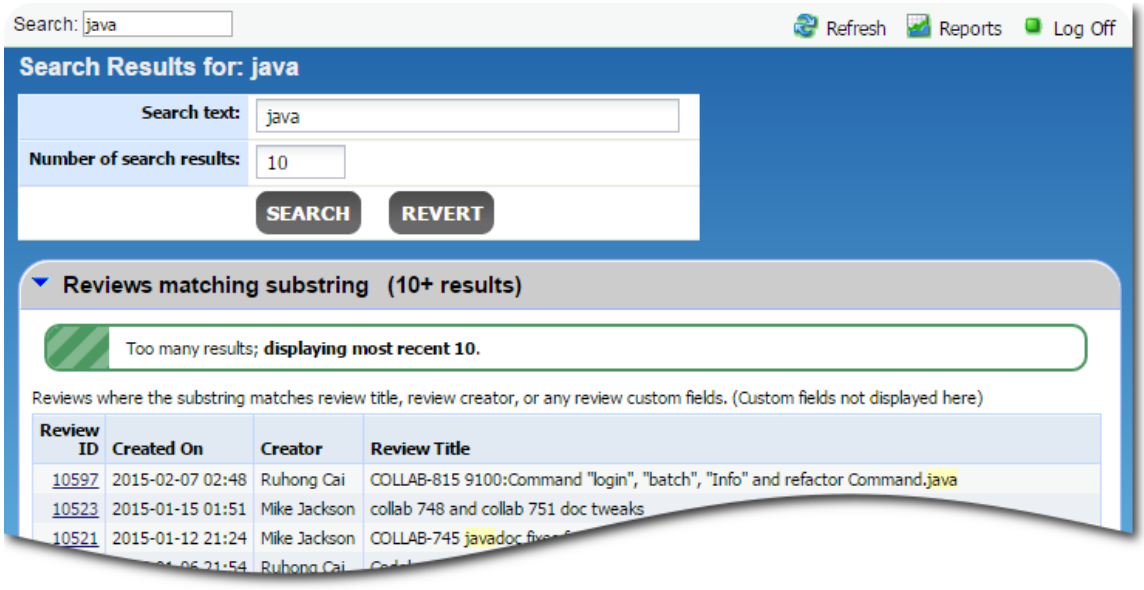



*Entering search term in Google Chrome*




*Entering search term in Mozilla Firefox*

Your Collaborator server will perform a search and display search results for the given term.




*Results of Collaborator search*

## Adding Collaborator as Browser's search engine

Collaborator uses the OpenSearch technology to act as the browser's search provider. The exact instructions on adding custom OpenSearch providers depends on a browser. Below are instructions for the most popular browsers.

### Internet Explorer:

1. Create a *.reg file with the following content:

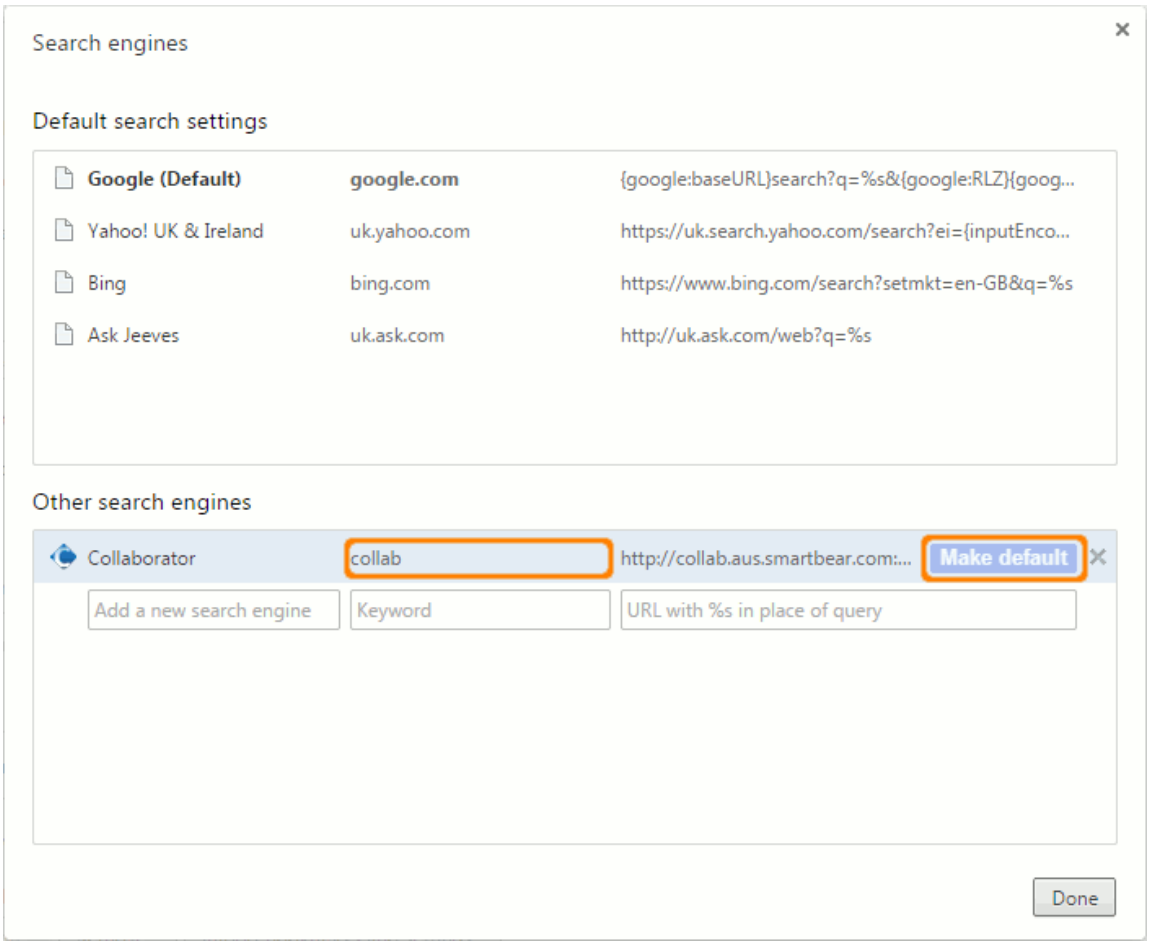
```
Windows Registry Editor Version 5.00


[HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\SearchScopes\Collaborator]
"DisplayName"="Collabortor"
"URL"="http://yourServer.com/go?page=SuperSearch&superSearchField=
{searchTerms}"
"FaviconPath"="C:\\loclPathToFavicon\\CC.ICO"
"FaviconURLFallback"="http://yourServer.com/i/cc.ico"
```

2. Change ***http://yourServer.com*** to the actual URL of your Collaborator server (including the protocol name: http or https).
3. Save the file and run it.
4. Confirm to make changes to the registry.
5. (Optional) Open Internet Explorer, go to **Tools | Manage Add-ons | Search Providers**, and set Collaborator as the Default Search Provider.

### Google Chrome:

1. Navigate to navigate to the URL of the Collaborator Web Client.
2. Click Chrome menu.
3. Select Settings.
4. Go to the "Search" section.
5. Press "Manage Search Engines".
6. Under "Other search engines" select "Collaborator"




7. Specify a more memorable text shortcut for the Keyword field. That is, instead of the full URL of your Collaborator server enter, for example, "collab". Keywords are entered in the address bar to denote which of search engines to use.

8. Press "Make default". This will add your Collaborator server as the default search engine.

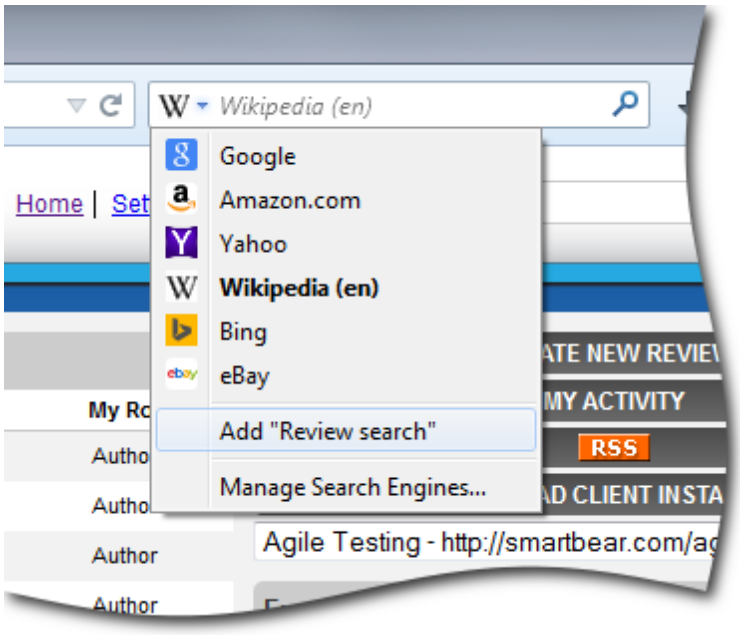
9. Press Done and close Chrome settings.

Later on you may restore your default search engine and use the specified keyword to search for the term on your Collaborator server.

## Mozilla Firefox:

1. Navigate to navigate to the URL of the Collaborator Web Client.

2. Open the search engine list and select "Add Review search". Firefox will add it to your list and make it the active search engine.

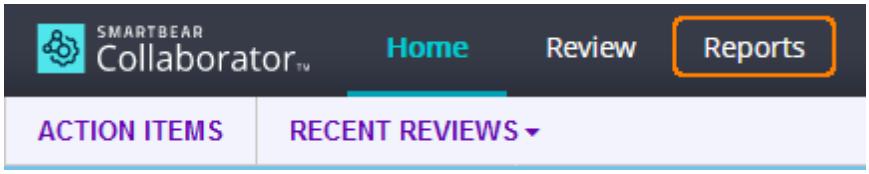



### 4.4.3 Reporting

Collaborator has a number of customizable pre-built reports.

***Note:*** *Support of pre-built reports varies depending on Collaborator edition. Collaborator Community has only basic reports, while Collaborator Team and Collaborator Enterprise have all types of pre-built reports. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

To access the customizable pre-built reports, click the Reports menubar option:




> If you are not an administrator, you may or may not have access to view reports depending on the Reports Access 187 setting.

The pre-built reports are divided into the following groups:

- Customizable Review Reports 468 - generate a list of reviews that meet certain requirements (reviews having the specified phase, status, duration and so forth).
- Review Detail Reports 473 - display detailed information about a particular review.

- Customizable Defect Reports 475 - generate a list of defects that meet certain requirements.
- Customizable User Reports 478 - generate a list of users that meet certain requirements.
- List Reports 480 - generate lists of users and SCM changes.

Additionally, you can retrieve data from Collaborator database and create your own custom reports 908 using an external reporting tool such as Excel, Access, Crystal Reports, or Business Objects. For further information on this approach, see Creating Custom Reports 908.

#### 4.4.3.1 Customizable Review Reports

This group of pre-built reports generate a list of reviews that meet certain requirements. Currently it includes the following review list reports:

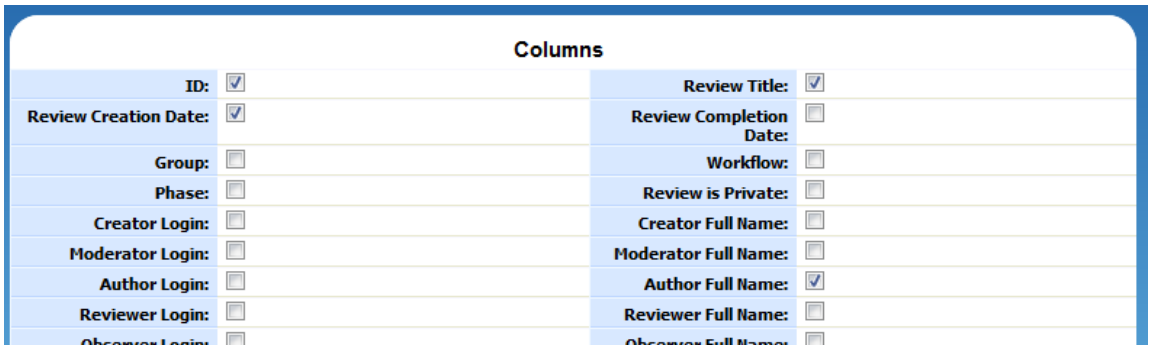
- **Reviews Currently In Progress** - Reviews in Planning, Inspection, or Rework phases.
- **My Reviews, Currently In Progress** - Reviews you're participating in which are currently in Planning, Inspection, or Rework phases.
- **Recently Completed Reviews** - Reviews completed normally (that is, not cancelled or rejected).
- **Stalled Reviews** - Reviews where no communication has happened for a while.
- **Trivial Reviews** - Reviews completed normally but too quickly to have been done properly.

Once you click on a report name, you will be directed to a page where you can tailor the report to your preferences.

> This documentation uses the default names of review fields, columns and options. On your Collaborator server this information may vary, depending on server configuration: display labels, role names, review custom fields and so on.

The options are divided into several groups: Columns, Filters and Options.

## Columns




In this section you can select which of the fields to include in the generated report. Thus you can choose how much or how little you view in the report.

Typically the following column names are displayed:

| | |
|---|---|
| ID | The unique identifier of the review. |
| Review Title | The title of the review. |
| Review Creation Date | Date and time, when the review was created. |
| Review Completion Date | Date and time, when the review was finished. Empty for in progress reviews. |
| Group | Name of a group of users associated with the review. |
| Template | Name of a workflow template associated with the review. |
| Phase | Current phase of the review. |
| Restrict Access | Who can access this review. |
| Creator Login, Creator Full Name | The login and full name of a person who created review. Creator may be a different person that a review author. |
| Author Login, Author Full Name | The login and full name(s) of review authors. |
| Reviewer Login, Reviewer Full Name | The login and full name(s) of reviewers. |

| | |
|---|---|
| Observer Login, Observer Full Name | The login and full name(s) of review observers. |
| Moderator Login, Moderator Full Name | The login and full name(s) of review moderators. |
| Defect Count | Total number of found defects. |
| Defects Per Hour | How much defects were found by reviewers in one hour. |
| Open Defect Count | Total number of open defects. |
| Comment Count | Total number of comments made. |
| Last Comment | Date and time, when the last comment was made. |
| Idle For | Time since any activity has occurred in a review. |
| File Count | Total number of files uploaded to the review. |
| LOC (Uploaded) | Total number of lines in all files. |
| LOC Added | Total number of lines added |
| LOC Removed | Total number of lines deleted |
| LOC Changed | Total number of lines with modifications |
| LOC Delta | Change in line count, that is, lines added minus lines removed. |
| LOC Reworked | Total number of all reworked lines (added + removed + changed). |
| Review Wall-Clock Duration | How much time has passed since the review was created and till the review was completed (or now, if the review is still in progress). |
| Total Person-Time | The total of all recorded time that all participants were looking at review. |
| Reviewer Time | The total of all recorded time that all reviewers were looking at review. |
| Author Time | The total of all recorded time that all authors were looking at review. |
| Number of Participants | Total number of review participants. |
| Average Participant Time | An average time each participant was looking at review. |

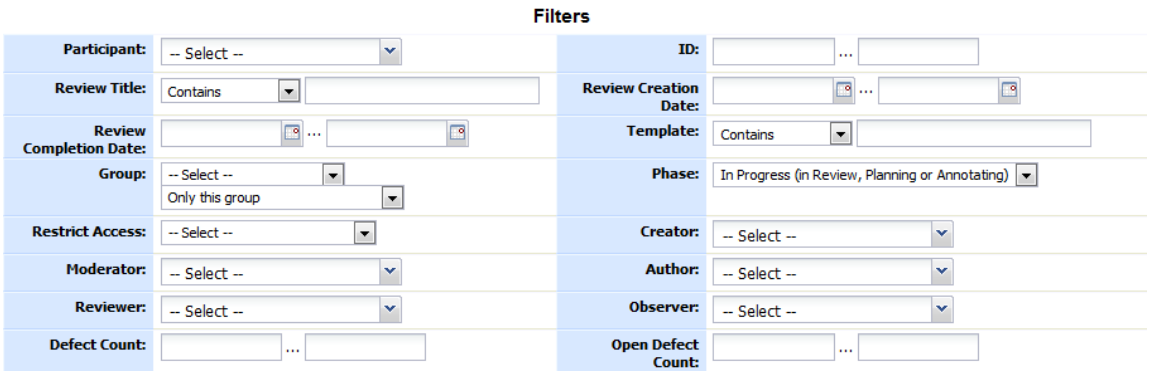
| | |
|---|---|
| Overview | A detailed description of the review. |
| <Review Custom Field> | Value of any other review custom field. |

## Filters

In the Filters section you can define which reviews to include or exclude from the report. Most of the fields that Collaborator can display (see Columns section), can also act as filter fields.

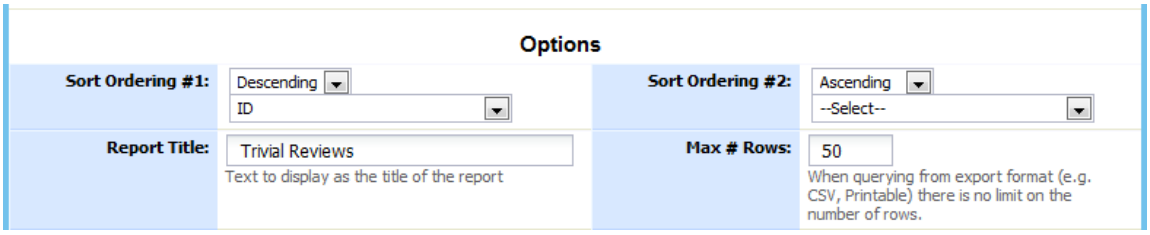
You can specify one or more conditions and Collaborator will show only those reviews that match these criteria. For instance, show reviews of some particular user, reviews created in certain time period, reviews whose title contains particular word and so on.

When several conditions are given, the report displays data that matches all of the specified conditions.




## Options

In the Options section you can specify how the found data is sorted, entitled and divided into pages. A report has a limit on the maximum number of rows that can be displayed: 50 rows by default. You can change the number in the "Max # Rows" field. When you export report data to CSV, SQL or print the report all rows are exported/printed regardless of this option value.

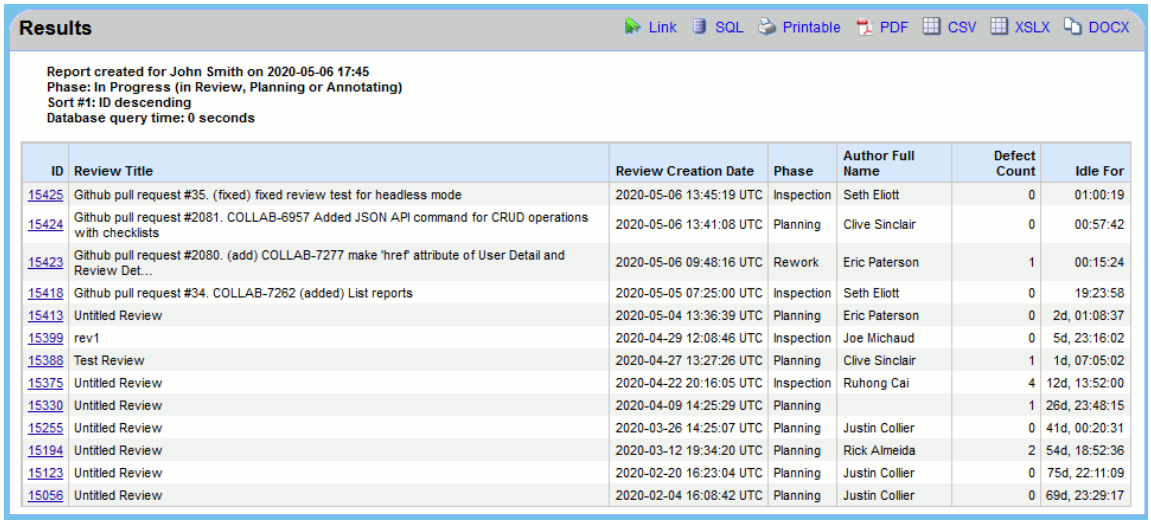


## Generating Reports

To generate any report from this group:

1. Choose the type of a report that you need and click its name.
2. Specify the desired options for the report: column set, filters, sorting and so on. To reset the report options and filters, press **Revert** button.
3. Click **Run** button to generate the report with the specified criteria and parameters.

The report will be displayed in the Results pane in the bottom of the page.

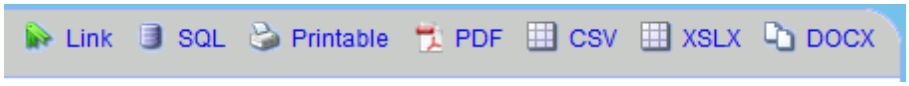



| ID | Review Title | Review Creation Date | Phase | Author Full Name | Defect Count | Idle For |
|---|---|---|---|---|---|---|
| 15425 | Github pull request #35. (fixed) fixed review test for headless mode | 2020-05-06 13:45:19 UTC | Inspection | Seth Eliott | 0 | 01:00:19 |
| 15424 | Github pull request #2081. COLLAB-6957 Added JSON API command for CRUD operations with checklists | 2020-05-06 13:41:08 UTC | Planning | Clive Sinclair | 0 | 00:57:42 |
| 15423 | Github pull request #2080. (add) COLLAB-7277 make 'href' attribute of User Detail and Review Det... | 2020-05-06 09:48:16 UTC | Rework | Eric Paterson | 1 | 00:15:24 |
| 15418 | Github pull request #34. COLLAB-7262 (added) List reports | 2020-05-05 07:25:00 UTC | Inspection | Seth Eliott | 0 | 19:23:58 |
| 15413 | Untitled Review | 2020-05-04 13:36:39 UTC | Planning | Eric Paterson | 0 | 2d, 01:08:37 |
| 15399 | rev1 | 2020-04-29 12:08:46 UTC | Inspection | Joe Michaud | 0 | 5d, 23:16:02 |
| 15388 | Test Review | 2020-04-27 13:27:26 UTC | Planning | Clive Sinclair | 1 | 1d, 07:05:02 |
| 15375 | Untitled Review | 2020-04-22 20:16:05 UTC | Inspection | Ruhong Cai | 4 | 12d, 13:52:00 |
| 15330 | Untitled Review | 2020-04-09 14:25:29 UTC | Planning | | 1 | 26d, 23:48:15 |
| 15255 | Untitled Review | 2020-03-26 14:25:07 UTC | Planning | Justin Collier | 0 | 41d, 00:20:31 |
| 15194 | Untitled Review | 2020-03-12 19:34:20 UTC | Planning | Rick Almeida | 2 | 54d, 18:52:36 |
| 15123 | Untitled Review | 2020-02-20 16:23:04 UTC | Planning | Justin Collier | 0 | 75d, 22:11:09 |
| 15056 | Untitled Review | 2020-02-04 16:08:42 UTC | Planning | Justin Collier | 0 | 69d, 23:29:17 |

## Exporting Reports

To export any report from this group:

1. Generate the desired report as described above.
2. Click one of the toolbar buttons in the top right of the **Results** box:




- The **Link** button will bring up a pop-up window with a direct link to the current customized report (with all the current options and filters preserved). You can bookmark this link to use whenever you want to revert back to the review configuration.

- The **SQL** button will display an SQL query that was used to generate the current report. You can use this SQL query when creating custom reporting tool[908]. This button is not available to regular users, unless the *Reports access*[187] setting is set to *Enabled for everyone*.
- The **Printable** button will prepare the report in a printable format.
- The **PDF** button will export the report in PDF format.
- The **CSV** button will export the report in comma-separated values format. Note the **CSV** output format does not consume a license.
- The **XLSX** button will export the report as Excel spreadsheet.
- The **DOCX** button will export the report as Word document. (Due to report engine implementation, generated documents are compatible with Microsoft Word and Google docs, but could be incompatible with OpenOffice and LibreOffice.)

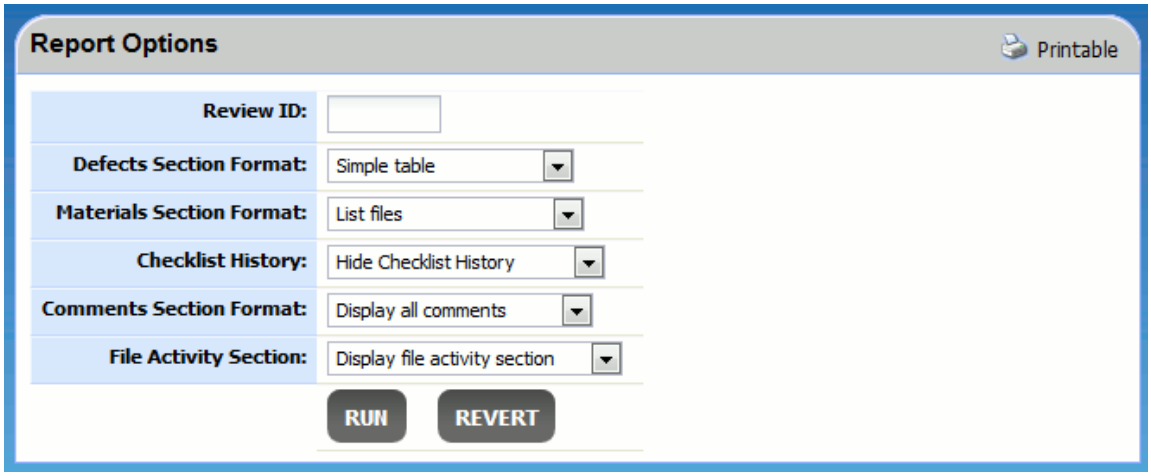
#### 4.4.3.2 Review Detail Reports

This group of pre-built reports display information about a particular review. Currently it includes the following reports:

- **Brief Summary** - One-page summarization of the review.
- **Detailed Report** - Complete log of the review.

  **Tip:** Another way to open the Brief Summary report for a specific review is to click the Details button in the Review Summary Screen[348].

Once you click on a report name, you will be directed to a page where you can tailor the report to your preferences.




You will need to specify the following options:

| | |
|---|---|
| Review ID | The unique identifier of the review. |
| Defects Section Format | Specifies whether to display the Defects section and its format. |
| Materials Section Format | Specifies whether to display the Materials section and its format. |
| Checklist History | Specifies whether to display the Checklist History section. |
| Comments Section Format | Specifies whether to display the Comments section and its format. |
| File Activity Section | Specifies whether to display the File Activity section and its format. |

Of these options, the Review ID number is required, other options have pre-defined values.

## Generating Reports

To generate any report from this group:

1. Choose the type of a report that you need and click its name.
2. Specify the desired options for the report. To reset the report options and filters, press **Revert** button.
3. Click **Run** button to generate the report with the specified criteria and parameters.

The report will be displayed in the Results pane in the bottom of the page.

## Exporting Reports in Printable Format

To export any report from this group:

1. Generate the desired report as described above.
2. Click one of the toolbar buttons in the top right of the **Results** box:

- The **Printable** button will prepare the report in a printable format.
- The **PDF** button will export the report in PDF format.
- The **DOCX** button will export the report as Word document. (Due to report engine implementation, generated documents are compatible with Microsoft Word and Google docs, but could be incompatible with OpenOffice and LibreOffice.)

#### 4.4.3.3 Customizable Defect Reports

This group of pre-built reports generate a list of defects that meet certain requirements. Currently it includes the following review list reports:

- **All Defects** - List of all defects from all reviews.
- **My Recent Defects** - List of defects you created recently.
- **My Open Defects** - List of defects you created and that are still open.

Once you click on a report name, you will be directed to a page where you can configure the criteria for the defect reports.

> This documentation uses the default names of defect fields, columns and options. On your Collaborator server this information may vary, depending on server configuration: display labels, role names, review custom fields and so on.

The options are divided into several groups: Columns, Filters and Options.

## Columns

In this section you can select which of the fields to include in the generated report. Thus you can choose how much or how little you view in the report.

Typically the following column names are displayed:

| | |
|---|---|
| Defect ID | The unique identifier of the defect. |
| State | The state of the defect: Open, Fixed or Tracked Externally. |
| Review ID | The unique identifier of the review to which the defect belongs. |
| Review Title | The title of the review to which the defect belongs. |
| Review Group | Name of a group of users associated with the review. |
| Review Creation Date | Date and time, when the review was created. |
| Review Completion Date | Date and time, when the review was finished. Empty for in progress reviews. |
| Created | Date and time, when the defect was created. |

| | |
|---|---|
| Creator Login, Creator Full Name | The login and full name of a person who created the defect. |
| File | Full name of a file to which the defect belongs. |
| File Version | The ID of file revision where the defect was found, or "unversioned" if the file is not under SCM. |
| Changelist ID | The ID of a changelist that contains the file where the defect was created. Empty if the file is not under SCM. |
| Changelist Date | Date and time, when the changelist was created. |
| Changelist Author | Name of a person who uploaded the changelist. |
| Changelist Comment | The comment of a changelist. |
| SCM Type | Type of a version control system. |
| Location | Location for context specific defects. |
| LocatorType | Type of the defect: Global, Annotation, Overall for File Revision or URL, Line, Coordinate, Label, Cell. See Types of Review Comments and Defects 421 |
| Comment | Comment that relates to the defect. |
| External Name | Name assigned to the defect in the external issue tracker system. |
| Severity | Indicates how critical the defect is. |
| Type | Indicates the type of found defect. |
| <Defect Custom Field> | Value of any other defect custom field. |

## Filters

In this section you can define which defects to include or exclude from the report. Most of the fields that Collaborator can display (see Columns section), can also act as filter fields.

You can specify one or more conditions and Collaborator will show only those reviews that match these criteria. For instance, show defects in some particular state, defects created in certain time period, defects whose comment contains particular word and so on.

When several conditions are given, the report displays data that matches all of the specified conditions.

## Options

In the Options section you can specify how the found data is sorted, entitled and divided into pages. A report has a limit on the maximum number of rows that can be displayed: 50 rows by default. You can change the number in the "Max # Rows" field. When you export report data to CSV, SQL or print the report all rows are exported/printed regardless of this option value.

## Generating Reports

To generate any report from this group:

1. Choose the type of a report that you need and click its name.
2. Specify the desired options for the report: column set, filters, sorting and so on. To reset the report options and filters, press **Revert** button.
3. Click **Run** button to generate the report with the specified criteria and parameters.

The report will be displayed in the Results pane in the bottom of the page.

## Exporting Reports

To export any report from this group:

1. Generate the desired report as described above.
2. Click one of the toolbar buttons in the top right of the **Results** box:


- The **Link** button will bring up a pop-up window with a direct link to the current customized report (with all the current options and filters preserved). You can bookmark this link to use whenever you want to revert back to the review configuration.
- The **SQL** button will display an SQL query that was used to generate the current report. You can use this SQL query when creating custom reporting tool 908. This button is not available to regular users, unless the *Reports access* 187 setting is set to *Enabled for everyone*.
- The **Printable** button will prepare the report in a printable format.
- The **PDF** button will export the report in PDF format.
- The **CSV** button will export the report in comma-separated values format. Note the **CSV** output format does not consume a license.
- The **XLSX** button will export the report as Excel spreadsheet.

- The **DOCX** button will export the report as Word document. (Due to report engine implementation, generated documents are compatible with Microsoft Word and Google docs, but could be incompatible with OpenOffice and LibreOffice.)

#### 4.4.3.4 Customizable User Reports

This group of pre-built reports generate a list of users that meet certain requirements. Currently it includes the following review list reports:

- **Enabled Users** - List of users who are allowed to log into Collaborator server
- **Disabled Users** - List of users who are no longer allowed to log into Collaborator server
- **Administrators** - List of active administrator accounts
- **All Users** - List of users that have been created in the system
- **Activity This Week** - List of users who were active in reviews this week
- **Activity Last Week** - List of users who were active in reviews last week
- **User Detail Report** - Detailed report of user information

Once you click on a report name, you will be directed to a page where you can configure the criteria for the user reports.

All reports from this group, except for the User Detail Report, have the options divided into several groups: Columns, Filters and Options. Whereas, the User Detail Report has only one group of options.

## Columns

In this section you can select which of the fields to include in the generated report. Thus you can choose how much or how little you view in the report.

Typically the following column names are displayed:

| | |
|---|---|
| ID | The unique identifier of the user. |
| Login | User login |
| Name | User full name |
| E-mail | User e-mail address. |
| Phone | User phone number. |
| Last Login | Date and time, when the user has logged in last time. |

| | |
|---|---|
| Last Activity | Date and time, when the user has performed any actions on Collaborator server. |
| Last Logout | Date and time, when the user has logged out last time. |
| Admin | Indicates whether the user is an administrator. |
| Enabled | Indicates whether the user is enabled. |
| User Activity | Duration of user activity in the given time period. |

## Filters

In this section you can define which users to include or exclude from the report. Most of the fields that Collaborator can display (see Columns section), can also act as filter fields.

You can specify one or more conditions and Collaborator will show only those reviews that match these criteria. For instance, show users that belong to some particular group, users who were active in last week and so on.

When several conditions are given, the report displays data that matches all of the specified conditions.

## Options

This section allows you to edit the sorting, title, and number of rows of the displayed report. A report has a limit on the maximum number of rows that can be displayed: 50 rows by default. You can change the number in the "Max # Rows" field. When you export report data to CSV, SQL or print the report all rows are exported/printed regardless of this option value.

## User Detail Report Options

The User Detail report has only two options:

| | |
|---|---|
| Participant | The full name and login of the desired user. |
| Activity Period | Time period for which to generate the user activity report. |

## Generating Reports

To generate any report from this group:

1. Choose the type of a report that you need and click its name.

2. Specify the desired options for the report: column set, filters, sorting and so on. To reset the report options and filters, press **Revert** button.

3. Click **Run** button to generate the report with the specified criteria and parameters.

The report will be displayed in the Results pane in the bottom of the page.

## Exporting Reports

To export any report from this group:

1. Generate the desired report as described above.

2. Click one of the toolbar buttons in the top right of the **Results** box:


- The **Link** button will bring up a pop-up window with a direct link to the current customized report (with all the current options and filters preserved). You can bookmark this link to use whenever you want to revert back to the review configuration.
- The **SQL** button will display an SQL query that was used to generate the current report. You can use this SQL query when creating custom reporting tool[908]. This button is not available to regular users, unless the *Reports access*[187] setting is set to *Enabled for everyone*.
- The **Printable** button will prepare the report in a printable format.
- The **PDF** button will export the report in PDF format.
- The **CSV** button will export the report in comma-separated values format. Note the **CSV** output format does not consume a license.
- The **XLSX** button will export the report as Excel spreadsheet.
- The **DOCX** button will export the report as Word document. (Due to report engine implementation, generated documents are compatible with Microsoft Word and Google docs, but could be incompatible with OpenOffice and LibreOffice.)

### 4.4.3.5 List Reports

This group of pre-built reports generate various types of data about Collaborator users and source control management integration. They include the following reports:

- **User List** - List of users in the system and when they last accessed the server
- **Reviews by Changelist** - List of reviews in the system, organized by SCM changelists present in the review

- **Changes List** - List of SCM changes in the system

Once you click on a report name, you will be directed to a page where you can configure the criteria for the user reports.

Each list report has the same basic structure. In the upper part it contains several filter fields and in the lower part it displays a table that lists the users, changes or reviews that match the specified criteria.

## User List Report

This report lists the users registered on the current Collaborator server and indicates whether the user is active and when they last accessed the server. By default the report lists all the users. To filter the user list, enter a value into the User Login or User Full Name fields. In this case the report will list only the users whose login or full name contain the specified value.

## Reviews by Changelist Report

This report lists the reviews having SCM changelists linked to the review. By default the report lists the reviews changed during the current week. To specify another time period for the changes, enter the desired values into the Start Date and End Date fields. To filter the review list, enter a value into the Change List Author or Changelist ID fields. In this case the report will list only the changelists whose author or ID contain the specified value.

## Changes List Report

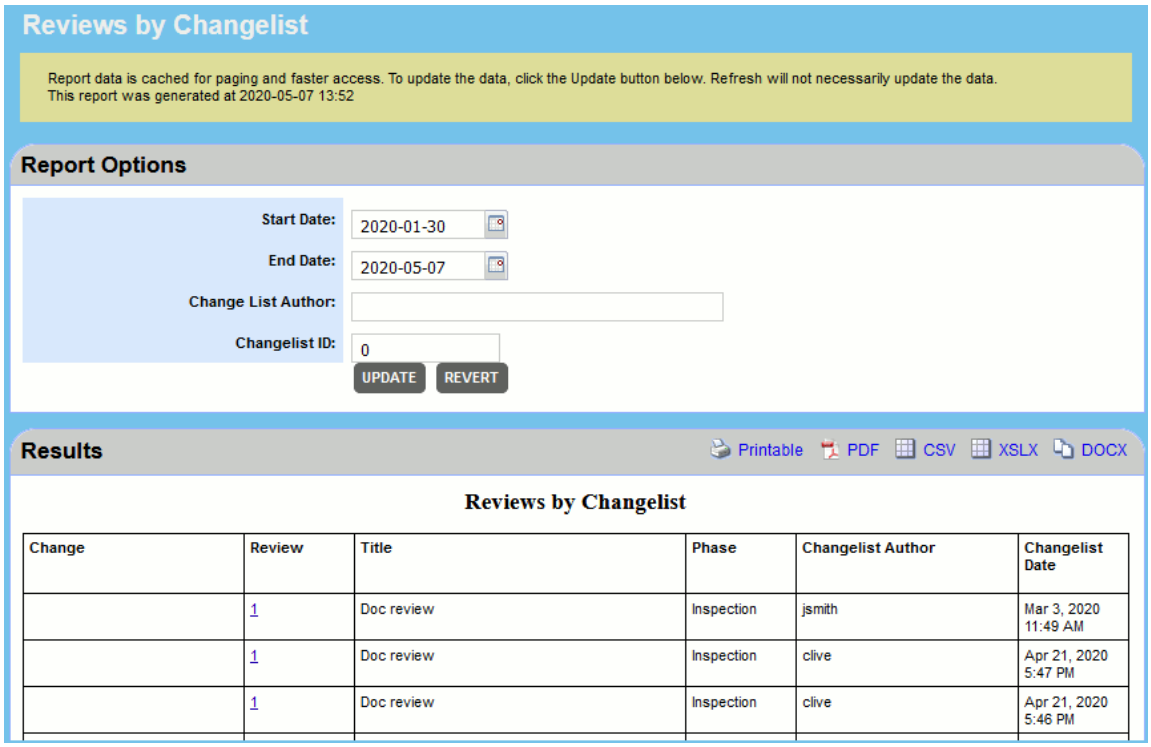
This report lists the SCM changes linked to the review. By default the report lists the changes made during the current week. To specify another time period for the changes, enter the desired values into the Start Date and End Date fields.

## Generating Reports

To generate any report from this group:

1. Choose the type of a report that you need and click its name.
2. (Optional.) Specify the desired filter conditions. To reset the report options and filters, press **Revert** button.
3. Click **Update** button to generate the report.

The report will be displayed in the bottom of the page.




**Warning:** Report data is cached by the server automatically. This makes it fast to page through data, but can be confusing since data is not updated automatically when you just refresh the page.

To actually refresh data, click the `Update` button under the list of filters.

## Exporting Reports

To export any report from this group:

1. Generate the desired report as described above.

2. Click one of the toolbar buttons in the top right of the **Results** box:

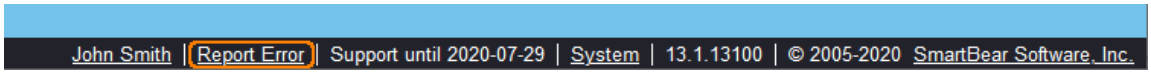
    - The **Printable** button will prepare the report in a printable format.
    - The **PDF** button will export the report in PDF format.
    - The **CSV** button will export the report in comma-separated values format. Note the **CSV** output format does not consume a license.
    - The **XLSX** button will export the report as Excel spreadsheet.

- The **DOCX** button will export the report as Word document. (Due to report engine implementation, generated documents are compatible with Microsoft Word and Google docs, but could be incompatible with OpenOffice and LibreOffice.)

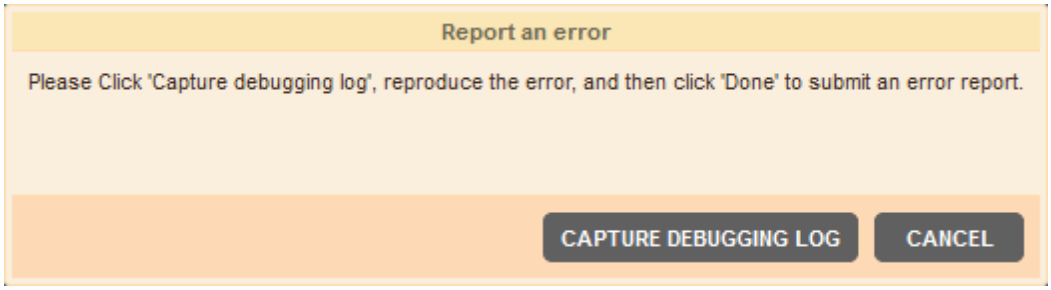
## 4.5 Troubleshooting

Collaborator has a debugging feature which makes gathering support logs easy.

The first step is to click the 'Report Error' link at the bottom of the web UI.

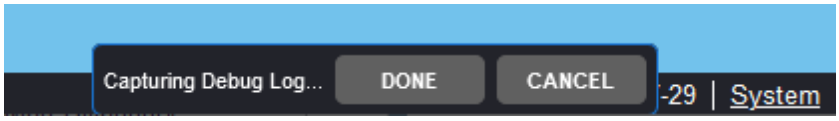



A **Report an error** dialog box will automatically launch once the link is clicked.

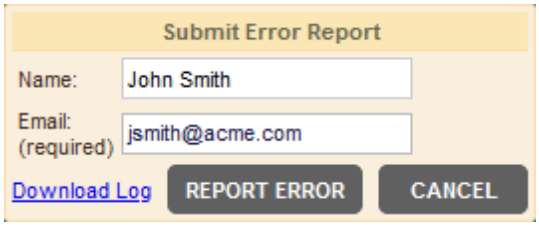



Click **Capture Debugging Log** and then go through the steps that originally caused the error. The log will automatically capture the error information.

Once the error has been reproduced, click **Done** at the bottom of the screen in the flashing Capturing Debug Log dialog box.




A **Submit Error Report** dialog box will be displayed. Fill in your contact information and click **Report Error**.



SmartBear Support Portal will be opened in web browser, where you can specify the details of your support case. The captured debugging log will be attached to your support case.

The Support team will answer you via email. All further communication will be made via email.

For information on our support policies, please visit our web site http://support.smartbear.com/about.

**Note:** Log files may accumulate in the temp directory if users enable debugging and do not disable it. These logs are capped at 50MB but if many users leave debugging on, the number of files can impact performance. These files can be deleted if they are not needed and administrators could set up a system task to clean that temp folder of anything older than 2 weeks (for example).

# 5 Desktop Clients

This chapter describes various desktop client applications and plugins, that simplify and expedite your daily work with Collaborator.

- In order to use any of Collaborator clients, your administrator must first install and configure 56 the server part of Collaborator.
- Clients version number should be less or equal to server's version number. Old clients (versions 11.x and earlier) are incompatible with Collaborator server 12.0 and later. To work with it, you will need to upgrade your clients to version 12.0 or later. To find out what is the minimal version of clients supported by your Collaborator server, check the Minimum client build 195 admin option.
- To use desktop clients when single sign-on authentication 130 is enabled, users should generate login tickets 318 and specify them in client connection settings instead of password.

## In This Section

- Configuration 493
  Describes how to configure client's connection to Collaborator server and to version control system.
- GUI Client 496
  Describes a cross-platform GUI client.
- Command-Line Client 506
  Describes a cross-platform command-line client.
- IDE Clients 524
  Describes plug-ins for Eclipse and Visual Studio that allow creating reviews directly from these IDE's
- Simulink Reviewer App 595
  Describes plug-in that allow creating reviews directly from MathWorks Simulink
- Microsoft Office Plug-ins 604
  Describes plug-ins for Microsoft Office that allow creating reviews directly from Word, PowerPoint or Excel
- Tray Notifier 613
  Describes a helper tray notifier client.
- External Diff Viewer Launcher 615
  Describes a helper tool that allows to launch an external difference viewer for Collaborator reviews.

## Related Topics of Interest

- Collaborator Server 56
  Describes the server component of Collaborator.
- Web Client 312
  Describes the web user interface of Collaborator.

# 5.1 Desktop Clients - Overview

Collaborator has a variety of clients for creating reviews and uploading files for review. While the Collaborator web server user interface provides the capability for uploading documents and, for some SCM integrations, the capability for creating reviews from committed changes, a client simplifies and expedites this process. Also, importantly, a client allows reviews to be created from local modifications to SCM files that are not accessible on the SCM server (that is, Collaborator clients allow for pre-commit reviews).

### Client/Server Version Compatibility

Clients version number should be less or equal to server's version number. Old clients (versions 11.x and earlier) are incompatible with Collaborator server 12.0 and later. To work with it, you will need to upgrade your clients to version 12.0 or later. To find out what is the minimal version of clients supported by your Collaborator server, check the Minimum client build 195 admin option.

### SCM Integration

All Collaborator clients integrate with SCM systems, for identifying files under source control and ease of selecting modified files to be uploaded to the Collaborator server for a review. Files are uploaded as changelists. For SCM systems that do not implicitly support changelists, the clients have mechanisms for selecting a group of files to be uploaded to a review.

The SCM integration understands adds, changes and deletes but does not currently track or detect file renames or moves.

### Creating Reviews and Uploading Files For Review

Each client also provides a means for creating a review, and for attaching a changelist to an existing review.

### Other Features

Additional features are available depending on the client. The client right for your development environment may be determined by your SCM system, or just a matter of personal choice.

## 5.2 Client Installation

The Client Installer includes the following components:

- GUI Client 496
- Command-Line Client 506
- Tray Notifier 613
- External Diff Viewer Launcher 615
- Perforce P4V/P4Win plug-ins 786
- Version Control System triggers

The client installer is cross-platform and can be run in an interactive GUI mode or silently from the command-line.

## Getting the Installer

Go to our website to get the latest installer for your platform:

https://support.smartbear.com/downloads/collaborator/

Installers are provided for Windows, Linux/Solaris and Mac OS. We also supply the raw files without a managing installer in the form of a compressed tar archive.

**Requirements:**

- Java 8, 9, or 11. (OpenJDK is recommended, Oracle JRE/JDK might work as well.) See Java Compatibility Matrix 1243 for more details.

**Notes:**

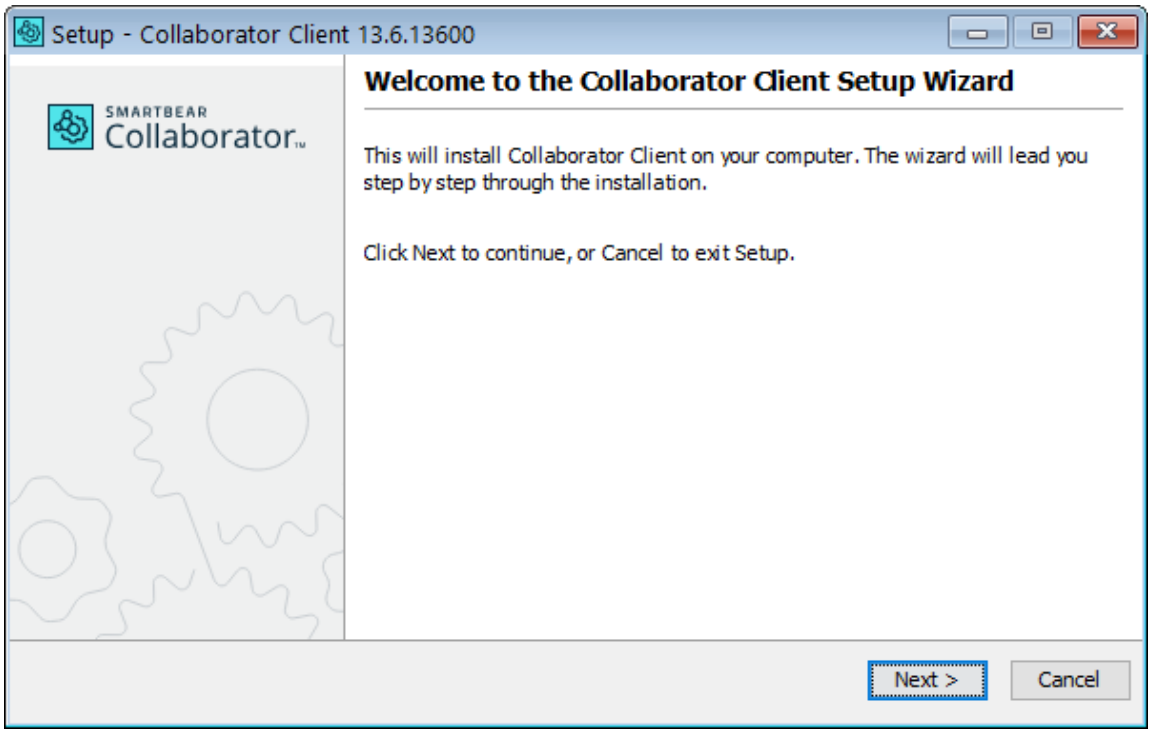
- If you have previously installed the client with the 32 bit installer and are now wanting to run the 64 bit installer, you must first uninstall the 32 bit client. This will not remove any existing configuration data, like preferences, SCM configurations, default browsers and so on.
- In order to enable integration with ClearCase Remote Client, you should launch client installer with administrator permissions on Windows. Otherwise, you can launch the installer as a regular user.

## Graphical Installation

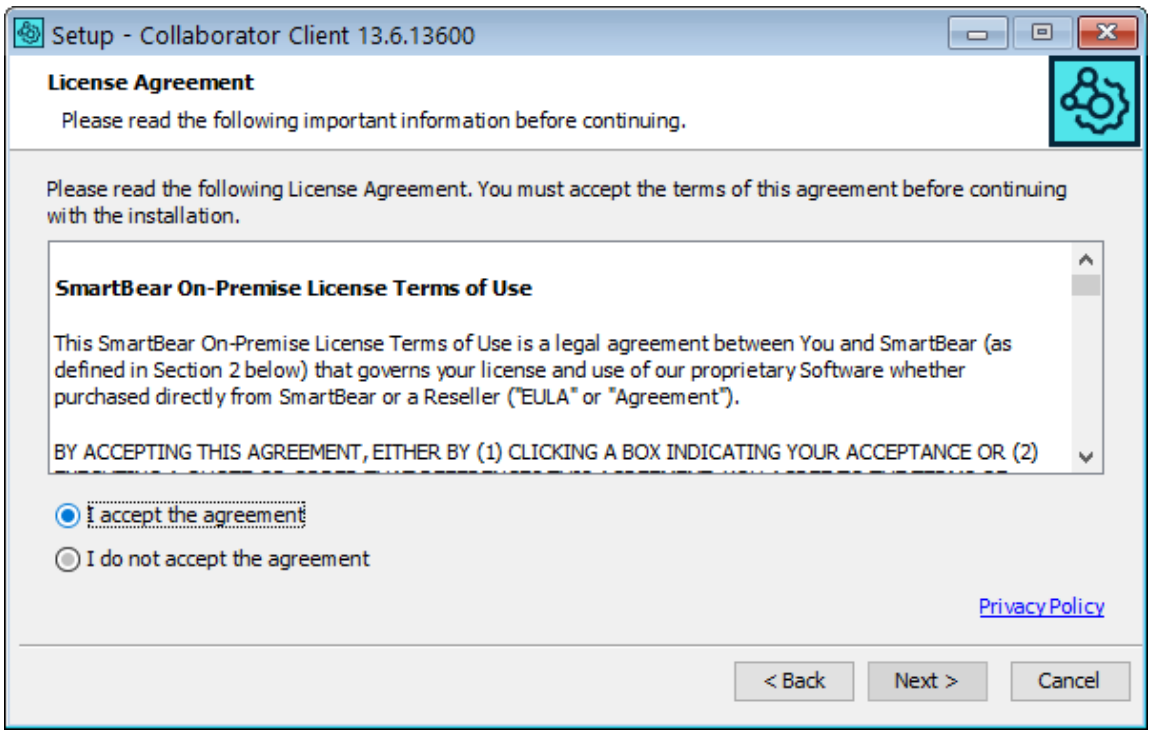
This is the recommended way to install the client software because you get a chance to see all the configuration options and allow the installer to validate your configuration.

On *nix platforms, you should grant execute permissions to the installer script and then launch it in shell:

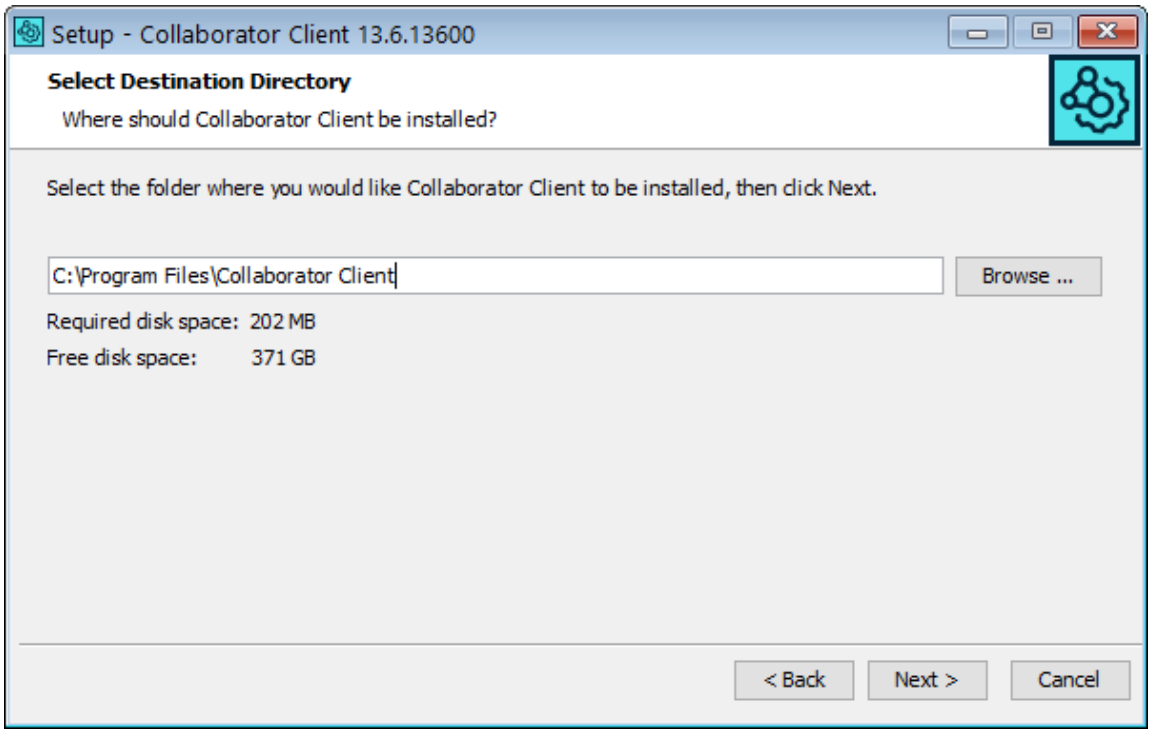
```
chmod +x ccollab_client_13_6_13601_unix.sh
sh ccollab_client_13_6_13601_unix.sh
```



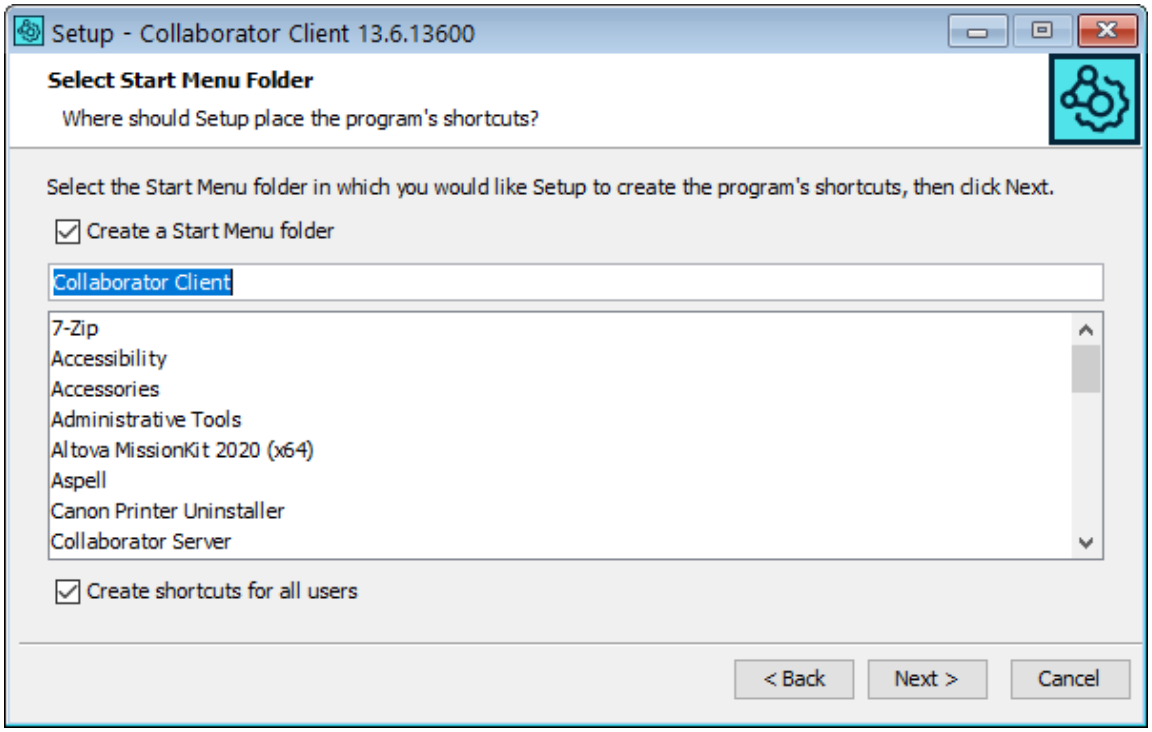

After the welcome screen, you are prompted with the EULA (End User License Agreement):



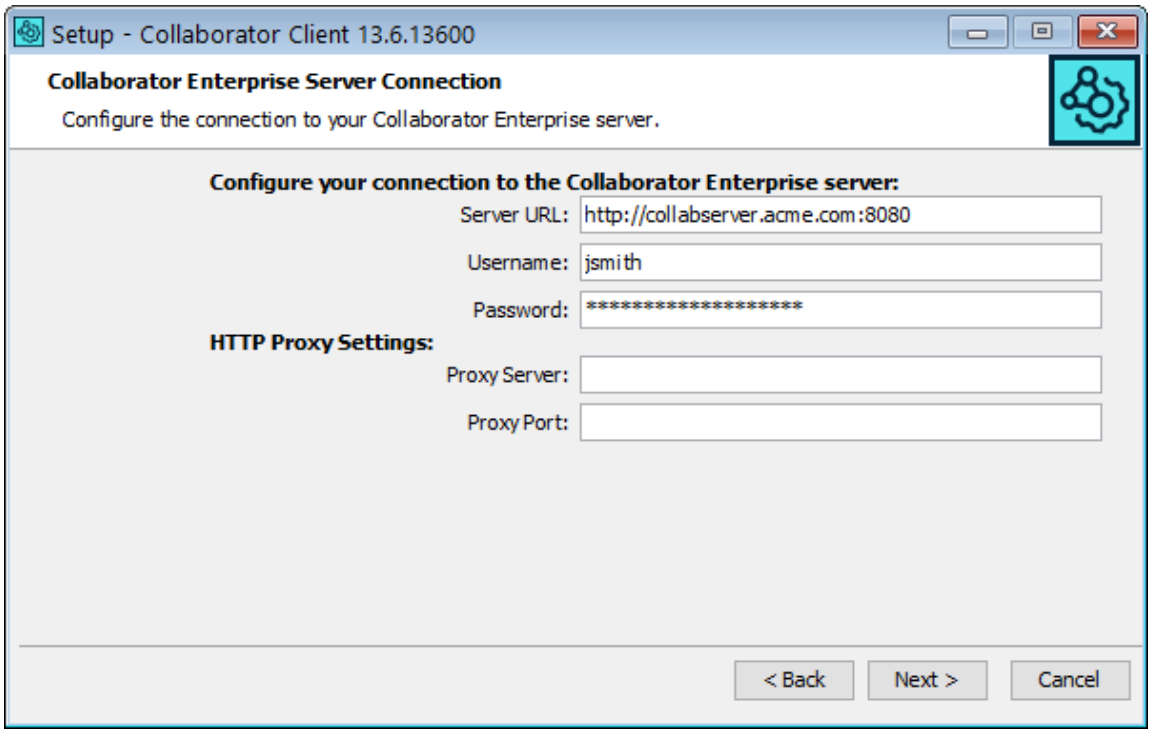

The next screen lets you select where to lay down the files for the installation:




The next screen lets you configure how and if your Start Menu (Windows only) is modified. This does not affect the behavior of the command-line client itself:



Next, you configure your connection to the Collaborator server:



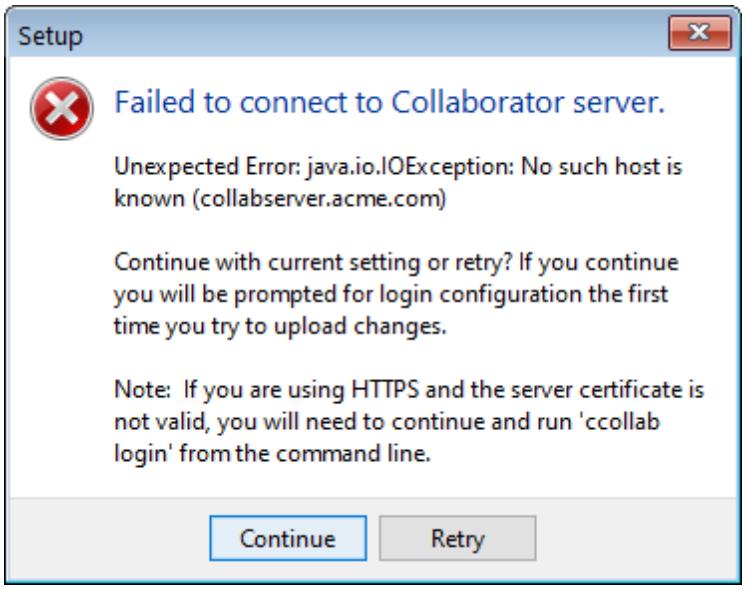

When you configure the URL, be sure to remember to specify the protocol (for example, `http://` or `https://`) and port number, and path. Leaving out one of those things is the most common mistake.

Your username and password or login ticket[318] must already exist[220] on the server. Log into[315] the server first from a web browser to test your account information.

When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead.

If you use a proxy, supply your proxy information as well.

If the connection to the server cannot be established, you will get an error dialog and an explanation of the problem:



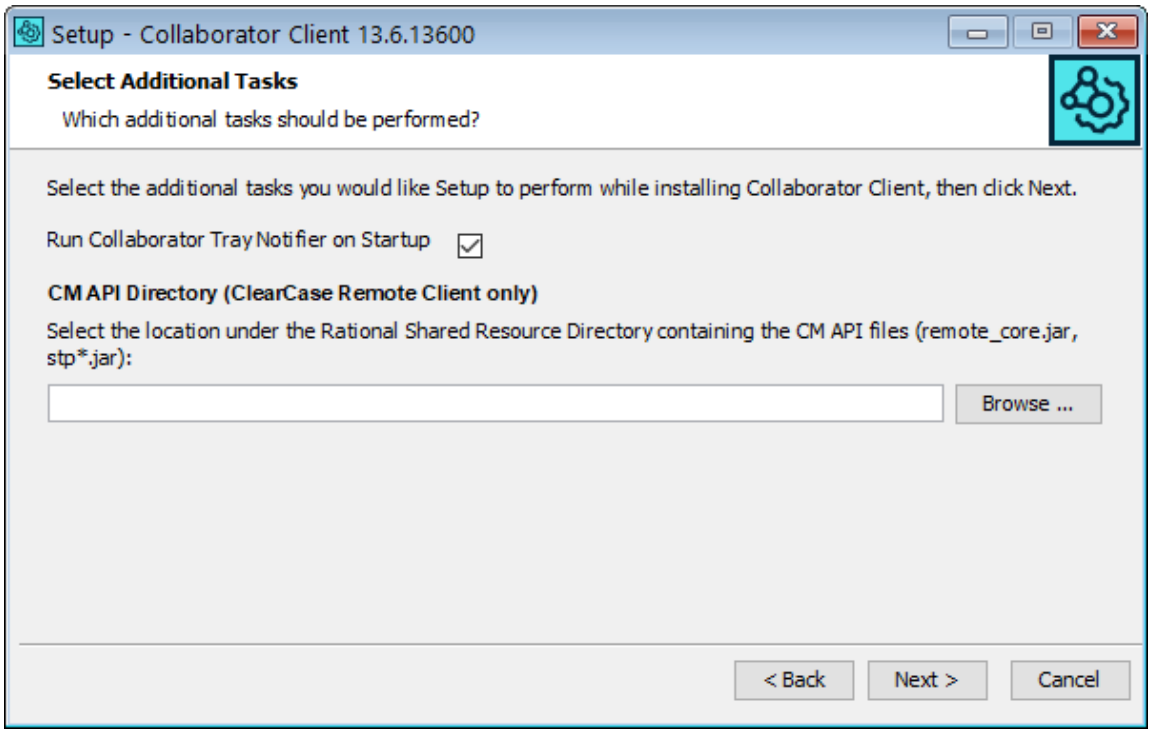

If you select "Retry", you can supply different configuration information and try again. Otherwise, if you select "Continue", the installation will continue but you will still need to set up your configuration 506.

After this step, you will be prompted for additional tasks.

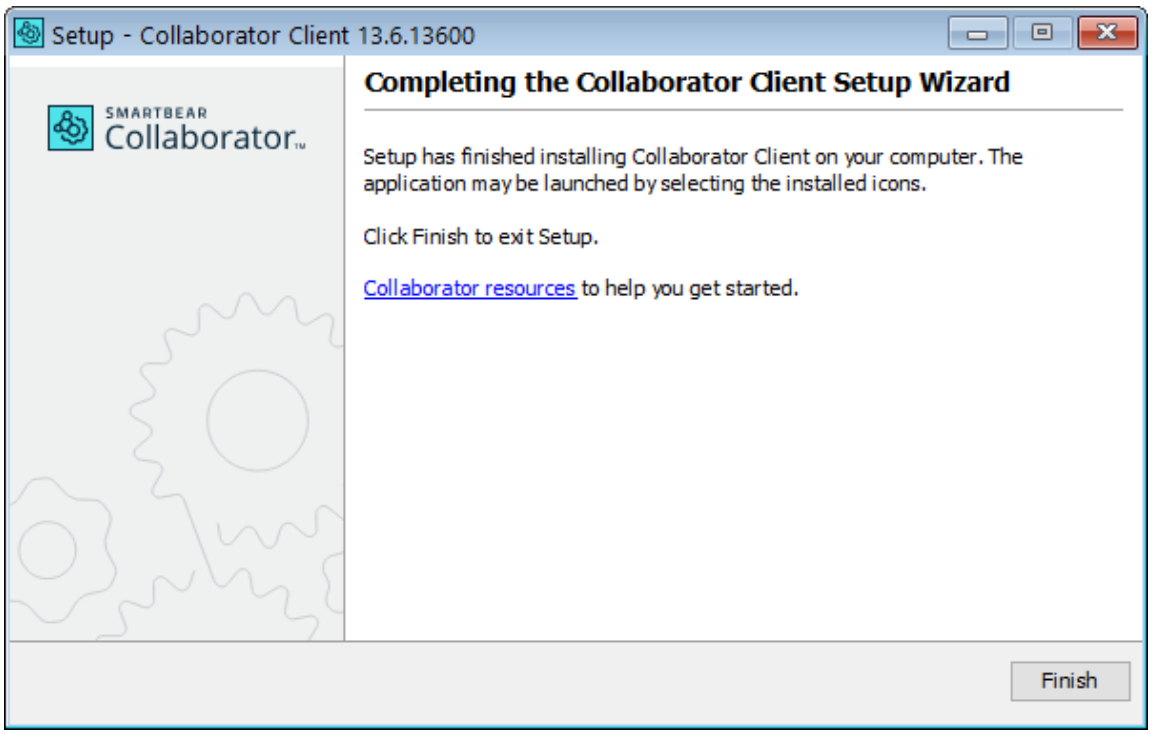
On Windows you will have the option to configure the Collaborator Tray Notifier to run on Startup. On all platforms, there will be options to configure the integrations with ClearCase Remote Client and Perforce Visual Tools.

In order to enable integration with ClearCase Remote Client, you should launch client installer with administrator permissions on Windows. You need to specify the location of the Rational CM API jar files on your system. See details in Support for ClearCase Remote Client 667.

The "Configure Addons To Perforce Visual Tools" option is shown only if Perforce Visual Tools is installed. If this option is selected, the installer will attempt to configure the P4V and P4Win integrations 786 for the current user. Other users will still need to configure the integrations manually.




Once you click "Next," the installer will complete:



### Unattended Installation

To run the installer without a GUI, run it from the command-line using the `-q` switch. In this case, you can also use the `-dir [directory]` switch to specify the target install directory. The -q switch gives you a silent install and will not prompt you for any installation instructions.

This installation technique can also be used to install clients remotely on many workstations.

If you would like to be prompted for installation instructions without using the graphical installer, you can run the installer from the console using the `-c` command. Please note that in this case you should answer to prompt messages in console input, that follows the prompt message (that is, **not** in the prompt message itself).

```
C:\install>ccollab_client_13_6_13601_windows_x64 -c
C:\install>This will install Collaborator Client on your computer.
OK [o, Enter], Cancel [c]
C:\install>o
```

**Warning:** When run in unattended mode, the installer cannot verify the client's connection with the Collaborator server. You might have to manually configure the connection with the server.

### Installing on a system with multiple JRE installations

See Known Issues with the Collaborator 978 for instructions on selecting a specific JRE to be used by Collaborator.

## 5.3 Configuration

Clients will need to be configured to both the Collaborator server and the SCM server.

Your client may have been successfully configured to the Collaborator server by the installer during the installation 486 process. If it has not been properly configured, please visit the Server Connection Configuration section 494 for detailed instructions.

Please make sure that your client is also configured to your version control or SCM. For instructions on how to do so, please visit the configuration sections under the appropriate client:

- GUI Client SCM Configuration 498
- Command-Line Client SCM Configuration 506

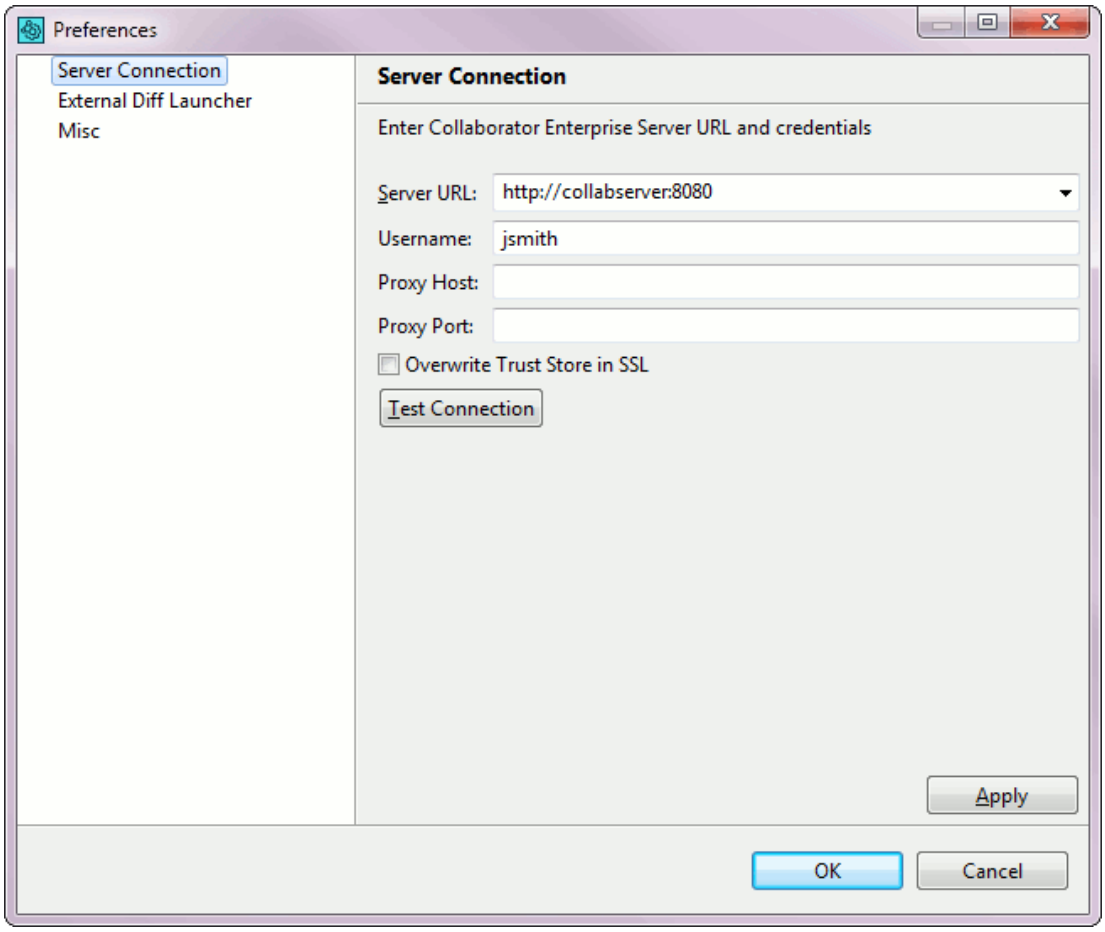
### 5.3.1 Server Connection Configuration

If you used the graphical installer[487], your connection to the Collaborator Server should be configured already. Otherwise, you will be prompted when you try to connect.

The GUI Client, Command-Line Client, Tray Notifier, and SCM Triggers share a common server connection configuration file[495]. The server connection configuration can be set with a graphical interface[494] using GUI Client or the Tray Notifier, or on the command-line[495] using the Command-Line Client.

## Graphical Server Connection Preferences

Selecting *File -> Preferences* on the GUI Client main screen[496] or selecting *Preferences* in the Tray Notifier context menu[614] open the graphical Server Connection preferences page.




The *Server URL* must include the correct port number and path if applicable. The user name and password are the same as you use when logging into the web server[315] (the password or login ticket[318] is requested when you log in for the first time). The proxy settings should be used if you have a client proxy between your workstation and the server.

Please note that if you are connecting to a server using https and the server uses a self-signed certificate, you *must* run "ccollab login" using the command line client first to import the certificate information to your keystore. Also, you may specify whether client should override the `java.net.ssl.truststore` VM option with hard coded default values ($JAVA_HOME\lib\security\cacerts). The latter option requires restarting the client in order to apply.

Use the *Test Connection* button to make sure the connection is working. If it fails, the error message will be helpful.

## Command-line Server Connection Global Options

The Command-Line Client uses the following global options[516] to specify the connection to the Collaborator Server:

- --url[517] - The server's fully-qualified URL.
- --user[517] - The user name for the login (same as web user login[315]). The password or login ticket is requested and stored the first time you log in.
- --server-proxy-host[517] - The proxy's URL (specify if you connect to the Internet via proxy).
- --server-proxy-port[517] - The port number to use on the proxy server (if connect via proxy).
- --overwrite-trust-store-in-ssl[516] - Overrides the `java.net.ssl.truststore` VM option with hard coded default values ($JAVA_HOME\lib\security\cacerts). Restart client to apply this option.

Use the `ccollab login` command to connect to the Collaborator Server and save your server connection options to the configuration file.

You can try testing your configuration[507] to verify the configuration is working.

## Configuration Files

Collaborator uses several configuration files to store configuration. When a user sets server connection configuration using the graphical interface[494] or the command-line[495], the settings are stored inside a directory called `.smartbear` inside the user's home directory. (Under Windows, the "home directory" is your "`Documents and Settings`" Profile directory.)

The `.smartbear` directory and the configuration files therein can be placed in other locations to establish default behavior. Each of the locations is loaded in a particular order of precedence, with each successive location overriding the settings (if any) in the previous locations:

1. `Custom location specified via the --pref-dir global option`
   If the `--pref-dir` command-line option is specified, Collaborator will use the configuration files from the specified location.

2. `INSTALLDIR/.smartbear`
   Here `INSTALLDIR` refers to the Collaborator installation directory. This is typically `/opt/ccollab-cmdline` under Unix or `C:\Program Files\Collaborator Client` under Windows.
   This is most useful for system-wide default settings.

3. `PROFILEDIR/.smartbear`
   Here `PROFILEDIR` refers to a directory specified using the Java property `smartbear.profile` on one of the Collaborator Client executables. With this property defined, the client will look in this directory for more configuration.
   This is most useful for executable-specific default settings, such as settings just for SCM server-side triggers.

4. `USERDIR/.smartbear`
   Here `USERDIR` refers to the user's home directory (under Windows, your "`Documents and Settings`" Profile directory).
   This is the default location for configuration settings to be stored, and is useful for user-specific settings.

# 5.4 GUI Client

The GUI Client is a cross-platform client interface to the Collaborator server. It is used for uploading files, either before or after check-in.

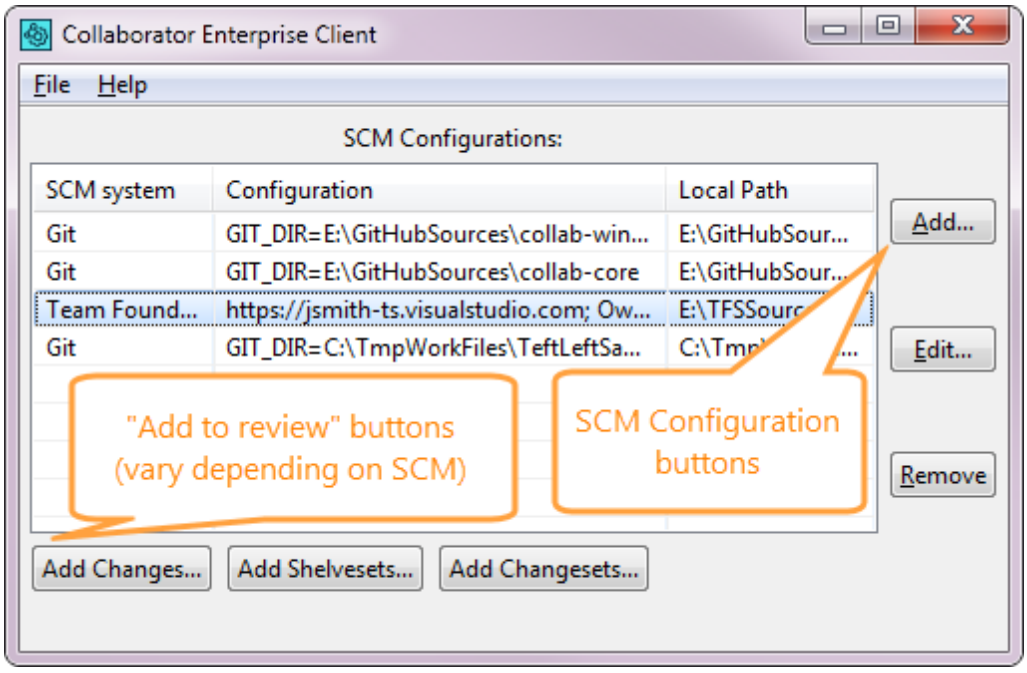
This chapter includes:



## 5.4.1 Main Screen

### Main Screen

The GUI Client's main screen contains a list of SCM configurations. Each SCM Configuration entry shows its SCM system, Configuration and Local Path (if any). Use the SCM Configuration buttons 497 on the right side of the screen to add or modify an SCM Configuration. Use the Add to Review buttons 497 on the bottom of the screen to upload review materials to the Collaborator Server.




*GUI Client Main Screen*

## Server Connection configuration

If you used the graphical installer 487, your connection to the Collaborator Server should be configured already. Otherwise, you will be prompted when you try to connect. You can also select *File -> Preferences* in the system menu to open the Server Connection preferences 494 page.

## SCM Configuration buttons

The buttons on the right side of the main screen are used to add or modify an SCM Configuration in the SCM Configurations list.

- *Add...* adds a new SCM Configuration
- *Edit...* edits the selected SCM Configuration
- *Remove* removes the selected SCM Configuration

## Add to Review buttons

The buttons on the bottom of the main screen are used to upload materials to a new or existing Review on the Collaborator Server. The buttons available will depend upon the selected SCM Configuration. Different buttons are available for different SCM systems:

- AccuRev 624

- CVS 636
- Git 649
- IBM Rational ClearCase 671
- IBM Rational Synergy 692
- Mercurial 731
- Microsoft Team Foundation Server 742
- PTC Integrity 755
- Perforce 771
- Subversion 805

## Keyboard Shortcuts

The GUI Client also offers keyboard shortcuts as follows:

```
File Menu
Alt + F = File menu
Alt + P = Preferences
Alt + X = Exit
Help Menu
Alt + H = Help menu
Alt + L = Capture Debugging Log
Alt + E = Email Support...
Alt + A = About Collaborator
```

* For Mac users, 'Alt' should be replaced with 'Cmd.'

## Known Issues

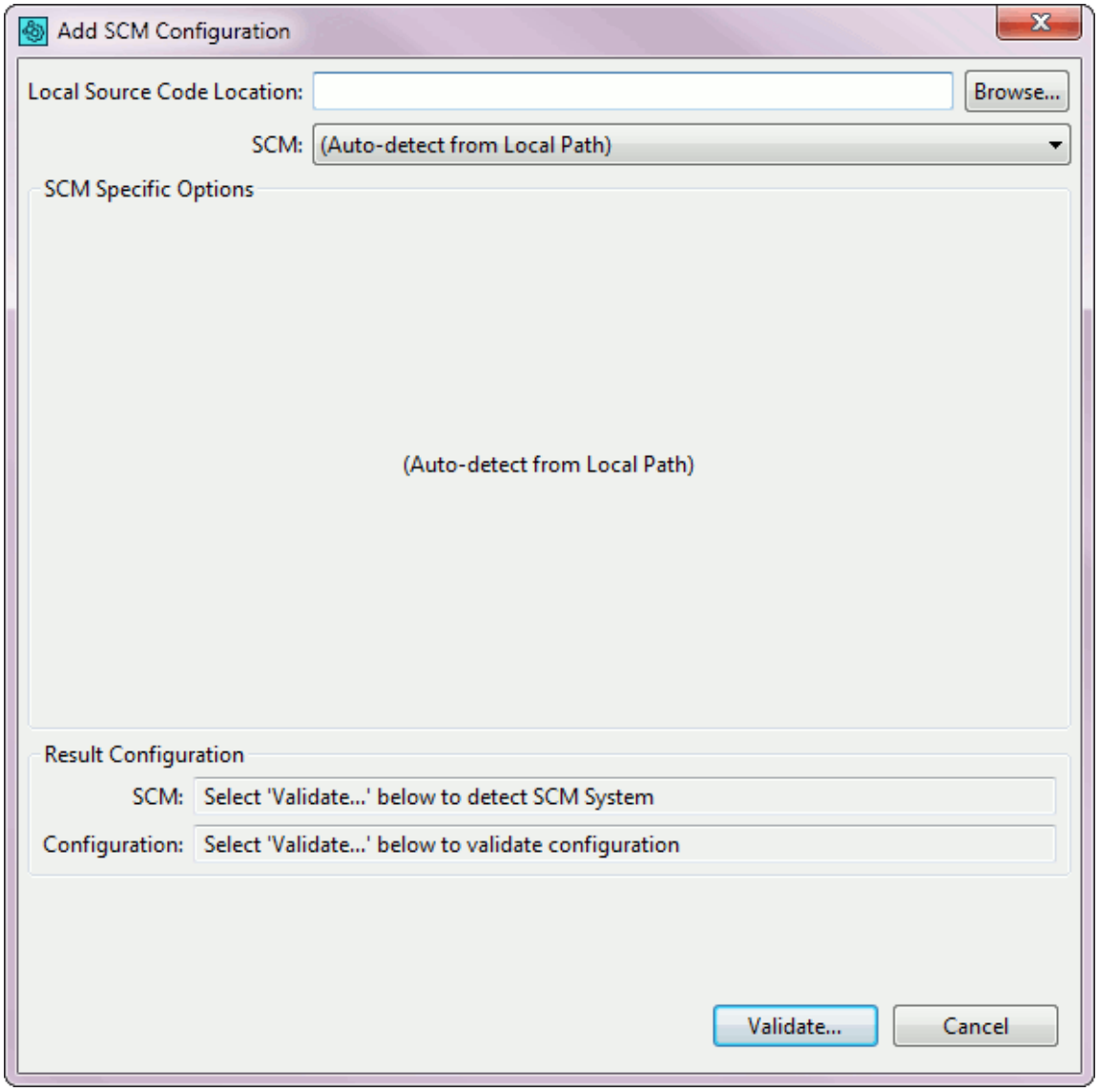
Because of technical issue with GTK3, GUI Client on Unix/Linux operating systems may fail to display table data on Add Changelists, Add Commits, Add Transactions and similar pages.

### 5.4.2 SCM Configuration

This section will describe how to configure the GUI Client to your SCM or version control. You will also have to configure 494 the client to the Collaborator server.

## Configuring Version Control

Pressing the *Add...* or *Edit...* SCM Configuration buttons[497] on the right side of the main screen[496] opens the SCM Configuration dialog.




*SCM Configuration dialog*

## Local Path

If you have a copy of your source on your local machine, enter its location into the optional *Local Path* field using the *Browse* button. Press the *Validate...* button to detect your SCM and SCM Configuration.

## Scm Specific Options

If there are additional settings necessary to connect to your SCM system, select your SCM system using the *SCM* drop-down to show the available options. The Collaborator integration requires that a command-line client be installed. There are different options available for each SCM system:

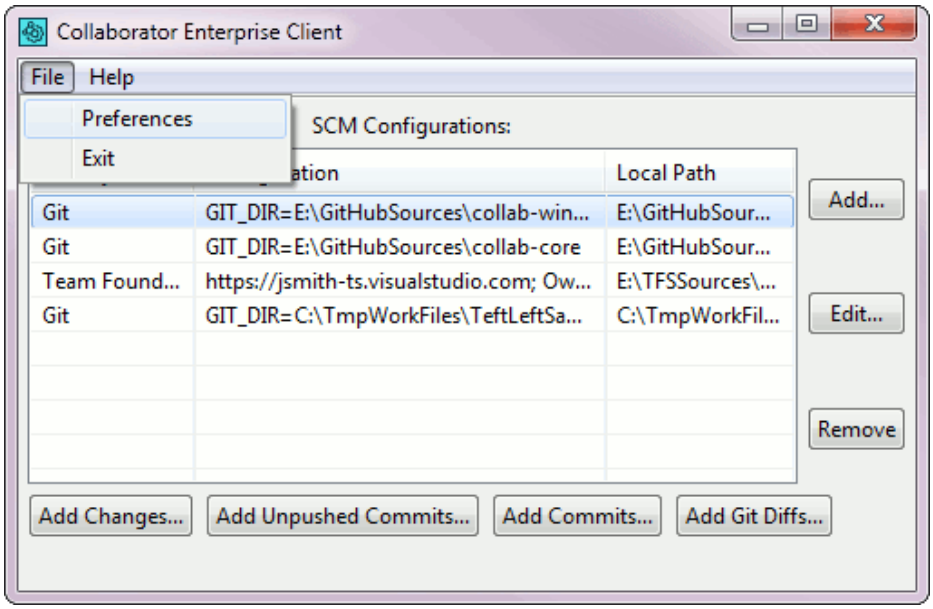
- AccuRev 623
- CVS 636
- Git 648
- IBM Rational ClearCase 670
- IBM Rational Synergy 691
- Mercurial 731
- Microsoft Team Foundation Server 740
- PTC Integrity 754
- Perforce 769
- Subversion 804

### 5.4.3 Preferences

The Preferences menu will allow you to change the settings for:

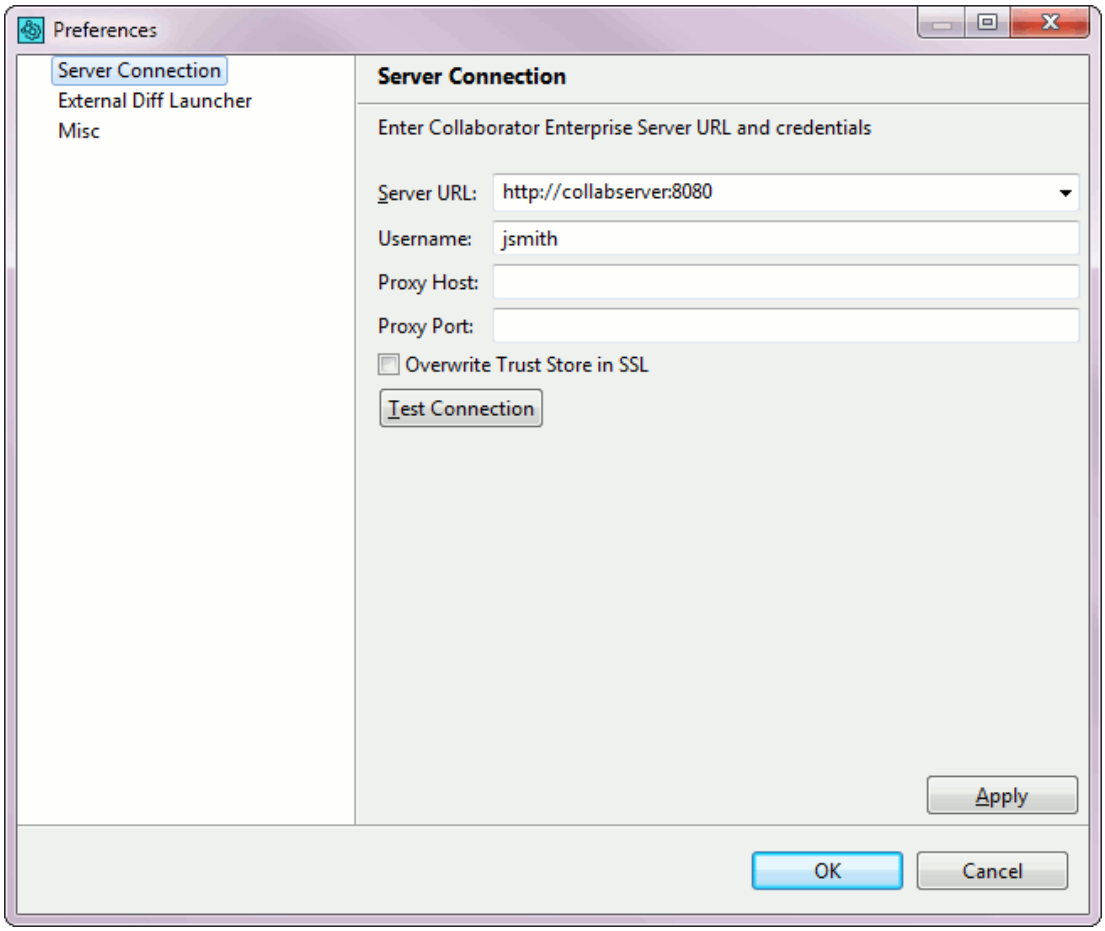
- Server Connection
- External Diff Viewer Launcher
- Miscellaneous

To view preferences, click on *File -> Preferences* in the main screen 613 of the GUI Client:



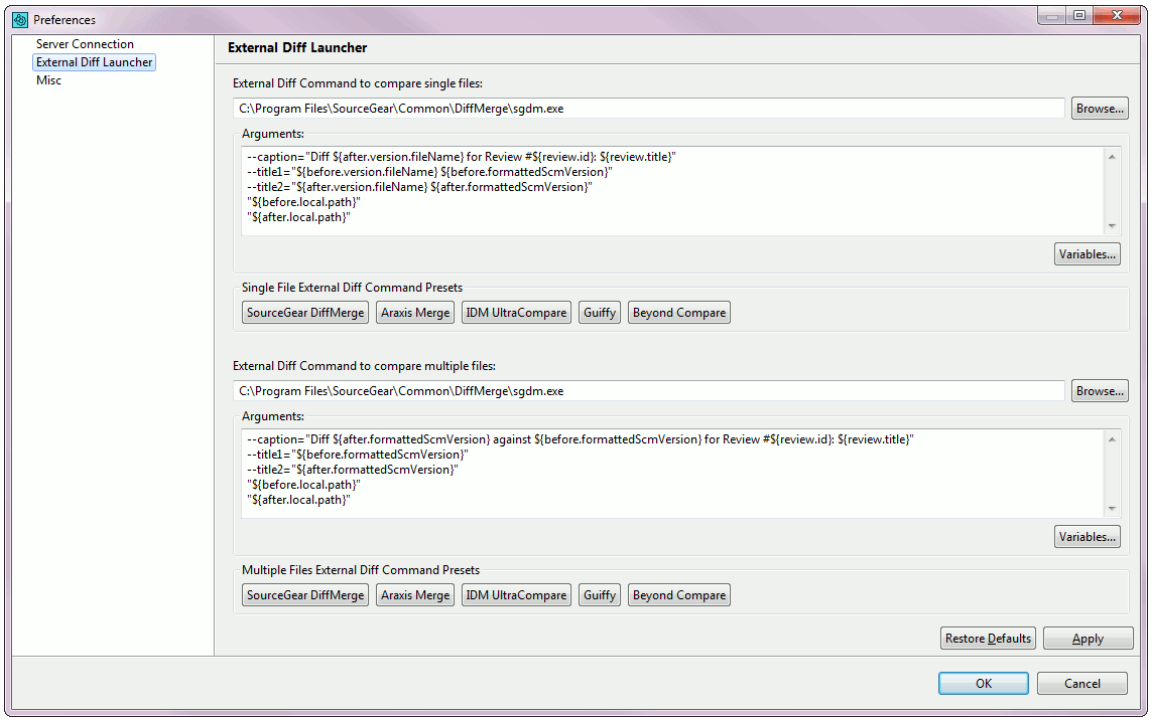

You can also view the Preferences menu via the Context Menu[614] of the Tray Notifier.

# Server Connection




Here you can configure the client connection to the server. See Server Connection Configuration 494.

# External Diff Viewer Launcher




Here you can configure settings for the External Diff Viewer Launcher[615]. This is a helper utility that allows comparing review materials in any third-party diff viewer instead of Web Client's diff viewer.

There are two separate **External Diff Command** settings, one for comparing two versions of a **single file**, and one for comparing **multiple files** (two directory trees).

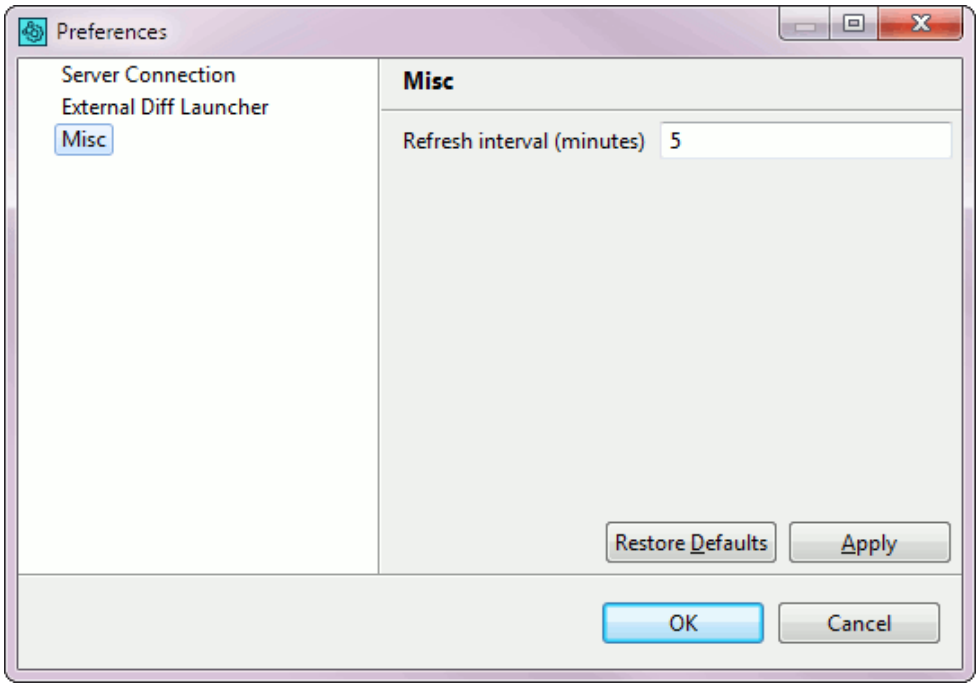
The **External Diff Command** field contains the absolute path to the executable the External Diff Viewer Launcher will launch.

The **Arguments** field contains arguments passed to the executable. Certain variables specified in the **Argument** field are substituted when the external diff command is invoked. Click the **Variables** button to view the list of available variables and their descriptions.

External Diff Viewer Launcher can work with almost any third-party diff viewer, and has presets for 5 most popular diff viewers: SourceGear DiffMerge, Araxis Merge, IDM UltraCompare, Guiffy, and Beyond Compare.

To use one of the presets, just click the appropriate button (for example, Guiffy) under **External Diff Command Presets**.

## Miscellaneous




In the Miscellaneous tab, you can change the refresh interval - that is how often to check Collaborator server for updates.

### 5.4.4 Troubleshooting

## Anti-Virus Software

If you have any anti-virus software[980] running, first determine whether it is interfering with the Collaborator client.

## Known Issues

Check the Known Issues Appendix[978] to see whether SmartBear already knows about this issue.

#### Linux packages

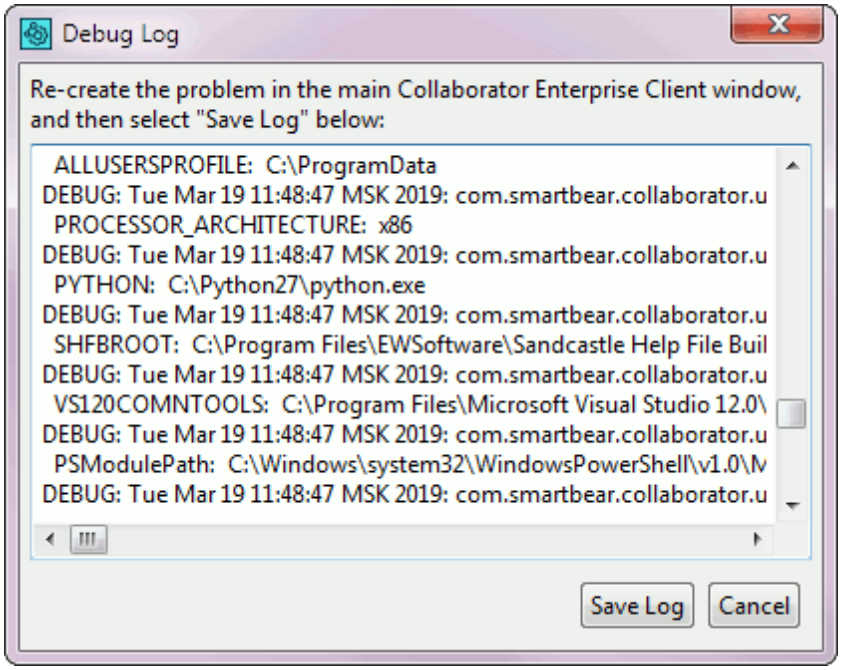
On RedHat Enterprise Linux or Fedora Core, you may need to install the following package names to use our GUI client. If the GUI client will not start successfully, please be sure that these packages are installed.

```
glib2
```

```
glibc
gtk2
libX11
libXau
libxcb
libXext
libXi
libXtst
```

## Capturing a debugging log

If you are experiencing problems with the GUI Client, especially if the problem is related to your SCM system, it will help SmartBear technical support if you send in a debugging log. To capture a debugging log, select *Help -> Capture Debugging Log* from the system menu on the Main Screen 496.




*Debug Log*

After the Debug Log window has opened, go back to the Main Screen 496 window and recreate your problem. You should see log messages being created in the Debug Log window. After you have recreated the problem, go back to the Debug Log window and select the Save Log button. This will prompt you for a convenient place on your machine to save the log file.

Please send the log file along with a full description of what you were trying to do when the error occurred to support@smartbear.com.

# 5.5 Command-Line Client

The Command-Line Client is a cross-platform client interface to the Collaborator server. It can be used by a human for uploading files, integrating with version control, and querying the server, or as a part of an automated script in a sophisticated ALM / build system.

This chapter includes:



## 5.5.1 Configuration

The Command-Line Client needs to be able to connect with your Collaborator Server, connect with your SCM system, and may need to launch a graphical editor for selecting files to be uploaded to reviews. The following sections describe configuring 494 the Command-Line Client for these actions.

### Global Options

The Command-Line Client supports many global options 516 and sub-command options. Global options may be specified on the command-line before the sub-command for each invocation, or they can be saved and applied to all future invocations using the set sub-command.

Default settings for global options can also be set using one or more configuration files 508.

### Configuring Collaborator Server Connection

If you used the graphical installer 487, your connection to the Collaborator Server should be configured already. Otherwise, you will be prompted when you try to connect. Try testing your configuration 507 to verify the configuration is detected correctly. You can also manually configure your connection 495 using the `login` sub-command.

## Configuring Version Control

In most cases, the Command-Line Client can automatically detect your SCM system configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your SCM system, or if you want to override the detected settings, you can specify SCM configuration settings using global options 516.

- AccuRev 629
- ClearCase 680
- CVS 640
- GIT 656
- PTC Integrity 758
- Perforce 777
- Subversion 812
- Team Foundation 748

## Testing Configuration Settings

To test the current configuration settings, go to working copy of your repository (the place on your local machine where files from an SCM system are checked-out) and execute:

```
ccollab info
```

This prints the current effective Collaborator and version control configuration settings. If you have not specified the scm 517 global option, the Command-Line Client will attempt to automatically detect your SCM configuration.

You will see some error messages if the configuration is not valid. If all goes well you should see something like this:

```
Connecting to Collaborator server http://myserver:8080
Connected as: John Doe (jdoe)
Auto-detecting SCM System for 'C:\mycode'
Detected Subversion
SCM Username: jdoe
```

```
SCM Config: repo=http://mysvnserver/repos/myrepo
```

## Graphical Editor Configuration

The Command-Line Client uses the default system text editor to display files to be uploaded for review. You can override this to point to any other text editor using the editor 516 global option.

Some editors on some platforms are launched as detached processes. This will cause the Command-Line Client to continue before the editor is closed, losing any changes made to the file list. The editor-prompt 516 global option can be used to pause the Command-Line Client after the editor is launched and wait for user keyboard input before continuing.

## Configuration Files

Collaborator uses several configuration files 495 to store default global options. When a user saves a global option 516 setting (using `ccollab set`), the setting is stored inside a directory called .smartbear inside the user's home directory. (Under Windows, the "home directory" is your "Documents and Settings" Profile directory.)

Global options specified by command-line switches 516 override settings stored in the configuration files.

### 5.5.2 Commands

## Basic Commands

The Command-Line Client provides a few basic commands which are unrelated to uploading files to the Collaborator Server:

- `help` - Display help on using the Command-Line Client
- `login` - Verify and/or change connection to the Collaborator Server
- `info` - Validates server connection and SCM configuration
- `set` - Save a global option 516 setting

## Upload Commands

The primary function of the Command-Line Client is to upload review materials (files) to the Collaborator Server.

The recommended commands to use vary by Version Control System:

- AccuRev Commands 622
- ClearCase Commands 666
- CVS Commands 635
- Git Commands 645
- Mercurial Integration 730
- PTC Integrity Commands 752
- Perforce Commands 765
- Rational Synergy Commands 691
- Subversion Commands 799
- Team Foundation Commands 738

In addition to the Version Control specific commands, the Command-Line Client provides a few basic upload commands that work for any (or no) Version Control system:

- `addfiles` - Upload local files without diffs
- `addchanges` - Uploads locally modified files controlled by an SCM
- `adddiffs` - Upload local or arbitrary diffs (see Uploading Diffs 510)
- `addsimulinkarchives` - Uploads Simulink model archive (see Reviewing Simulink Models 404)

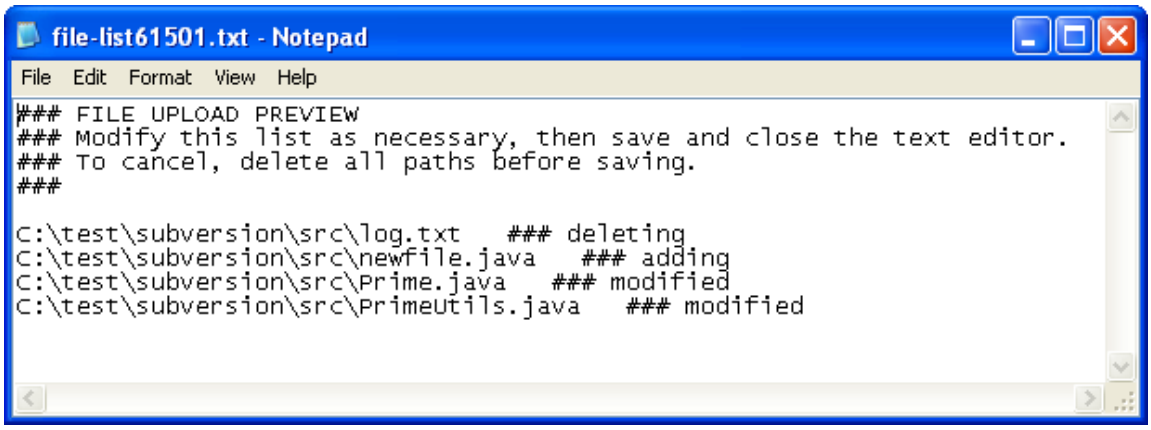
Upload commands that deal with files managed by Version Control systems should be called from any sub-folder within the working copy of your repository (the place on your local machine where files from an SCM system are checked-out). Otherwise, the commands may fail to detect the SCM configuration.

## Scripting Commands

The Command-Line Client also includes many commands meant for constructing custom scripts, usually used for integrating with external systems or implementing custom behaviors. For more information see Scripting 926.

### 5.5.3 Graphical Editor

The addchanges and addactivity 688 commands can launch a graphical editor with a file list so you can review the files about to be uploaded and make any corrections you wish:




The ### sequence is a comment. If all lines with files are deleted the operation is aborted. In this example there are two modified files, one file being added, and one being deleted.

If not otherwise configured, the command-line uses the default system editor to display the file list. See Graphical Editor Configuration 508 for information on specifying a different text editor to be used.

### 5.5.4 Uploading Diffs

You can upload arbitrary file differences using the `ccollab adddiffs` command. This subject is complex enough to warrant its own section in the manual.

## Uploading Local File Diffs

You can upload local file differences with the `adddiffs` command:

```
ccollab adddiffs <review> path-from path-to
```

The `path-from` and `path-to` fields are paths to two files or directories to compare, and *<review>* is an integer review-id, 'new', 'ask', or 'last'. For example, this command would upload any differences found as a change-set as a new review:

```
ccollab adddiffs new /old/directory /new/directory
```

## Uploading Arbitrary Diffs

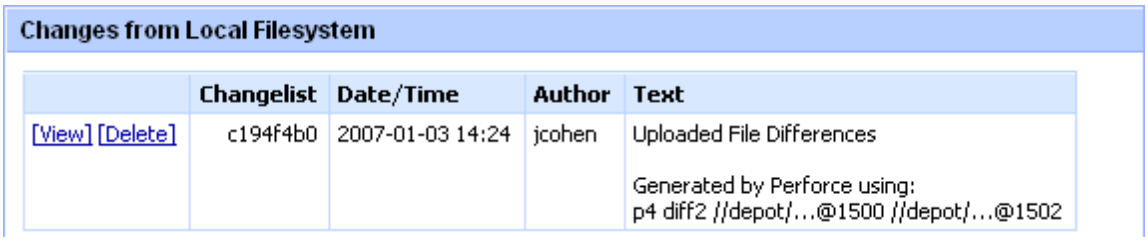
You can also upload arbitrary differences with the `adddiffs` command. Assuming textual diffs have already been generated in a file called `diffs.txt`, use this:

```
ccollab adddiffs review diffs.txt
```

Or you can pipe the results of another diff command right into the `ccollab` process using a single dash instead of a file path. For example, this command uses the GNU `diff` command to generate a unified diff format, then uploads it directly into review #2534:

```
diff -u100 dir1 dir2 | ccollab adddiffs 2534 -
```

After you upload diffs to the server you will see the "changelist" appear similar to this:



**Changes from Local Filesystem**

| | Changelist | Date/Time | Author | Text |
|---|---|---|---|---|
| [View] [Delete] | c194f4b0 | 2007-01-03 14:24 | jcohen | Uploaded File Differences<br><br>Generated by Perforce using:<br>p4 diff2 //depot/...@1500 //depot/...@1502 |

## Uploading Differences from Version Control

If you want to upload differences generated by a version control command-line tool, you should use the `ccollab add*diffs` command specific to your version control system (if available) instead of `ccollab adddiffs`.

Version Control specific `add*diffs` commands:

- `addardiffs` 632 (AccuRev 622)
- `addcvsdiffs` 643 (CVS 635)
- `addgitdiffs` 660 (Git 645)
- `addhgdiffs` 736 (Mercurial 730)
- `addp4diffs` 779 (Perforce 764)
- `addsvndiffs` 816 (Subversion 799)

These commands integrate with version control servers and supply exactly the right command-line switches to the underlying version control command-line to minimize server transmissions and maximize features like fully reproducing both previous and current versions of the document.

Many of the pitfalls with diffs described below are avoided when using `ccollab add*diffs`.

# Supported Diff Formats

There are many kinds of diff formats out there. Our software supports many of them; if you find a format that is not working, please let us know 32 so we can get your diff format into our unit tests.

- **Unified**
  The unified format starts with a two-line header naming the previous and name versions of a file, plus optional file information. Then it follows with diff-chunks starting with "@@" and then having both unmodified lines and added/removed lines. An exact definition of the format can be found here.
  Any number of these file-diffs can be strung together. Our software parses all of the files and uploads each as a "before" and "after" version. Context lines are preserved; lines skipped are reconstructed (more about this later 513).

- **UltraCompare Text Format**
  We support the proprietary text format emitted by the 3rd party commercial command-line tool UltraCompare.
  This is a Unicode diff with full file context making it an excellent format that preserves all file content.
  Although this utility emits only one file difference at a time, you can concatenate any number of these diffs together and upload them in one shot through the `ccollab` command-line. This can be especially handy if you are writing a script to automate this process.

- **Subversion Style**
  Subversion outputs a two-line header for each file; one starts with "Index:" and names the file in question, the next is a row of equal signs. Following the header is any of three types of diff output; we support all three but you should use the unified format for best results 513. Any number of these file-diffs can be strung together.
  Note that with Subversion you should use `addsvndiffs` 816 instead of `adddiffs`.

- **Perforce Style**
  Perforce has a variety of output formats; we support all of them. One is just the Unified format specified above. The other uses a one-line file header that shows the "before" and "after" file paths and version numbers with other file data, and then uses any of three types of diff output to encode the diffs. We support all three but you should use the unified format for best results 513. Any number of these file-diffs can be strung together.
  Note that with Perforce you should use `addp4diffs` 779 instead of `adddiffs`.

- **CVS Style**
  Similar to Subversion Style (see above) but has additional lines of output besides just those two file header markers.
  Note that with CVS you should use `addcvsdiffs` 643 instead of `adddiffs`.

- **GNU Format**
  The GNU format shows just additions, deletions, and modifications with no file context. An exact definition of the format can be found here.
  We do not recommend you use GNU format because (1) it does not support file names or multiple files and (2) it does not support context lines which causes problems when reconstructing files 513.

- **RCS Format**
  The RCS format shows just additions and deletions with no file context.
  We do not recommend you use RCS format because (1) it does not support file names or multiple files and (2) it does not support context lines which causes problems when reconstructing files 513.

- **Araxis Merge, BeyondCompare, GuiffyMerge, GNU diff, xdiff, and other tools**
  With these diff utilities, use an export option that creates unified diffs with lots of context. See the guidelines below 513.

When the command-line utility is uploading diffs it will print out the format that was detected. If it complains that the format is unknown, attach a copy of your diff data to an email to our tech support department 32 so we can add support for your format.

## Context Lines and Reconstructing Files

Most diff formats skip over lines that were unchanged, possibly leaving a few lines of "context" around each modification. This might be OK for "patch" utilities but it is not good when you want to do a review. Why not?

When you do the review, if you do not have many lines of context you cannot see much beyond the diffs themselves. Our software is smart enough to reconstruct the two files as well as possible, retaining all line numbers, but of course there will be gaps.

Here is how the side-by-side view 379 looks with some differences that only had three lines of context:

```
69  *** unspecified line placeholder ***
70  *** unspecified line placeholder ***
71  *** unspecified line placeholder ***
72      }
73  }
74
75  function preloadImage (object, url) {
76    eval (object + " = new Image ()");
77    eval (object + ".src = '" + url + "'");
78  }
79
80  preloadImage ("updatingimage", "../images/AIMSAnim.gif");
81
82  /*****************************\
83  |*** Show Updating window ***|
84  called from DCImageButton
85  *** unspecified line placeholder ***
86  *** unspecified line placeholder ***
87  *** unspecified line placeholder ***
```

You can see the "`*** unspecified line placeholder ***`" text where the diff skipped over lines.

How to prevent this? It depends on how you are generating diffs. Here are some guidelines:

- Always use Unified diff format[512], not RCS or GNU. Those others do not support lines of context *at all*, nor do they support filenames so your uploads will be "anonymous".
- Always include a lot of context with your Unified diff format. The way to specify this varies depending on your diff utility. For example, using GNU `diff` the command-line option is `-u1000` where "`1000`" is the number of lines of context. Selecting a big number like 1000 is not unreasonable! Reviewers will thank you for it.
- Use the version-control specific `ccollab add*diffs`[511] commands whenever you can. This automatically selects the right command-line switches to maximize context while minimizing data transmissions.

## 5.5.5 Troubleshooting

Check here for common problems and their solutions.

## Anti-Virus Software

If you have any anti-virus software[980] running, first determine whether it is interfering with the Collaborator client.

## Known Issues

Check the Known Issues Appendix[978] to see whether SmartBear already knows about this issue.

## Version History

Check the Version History on the SmartBear website to see if this issue has been resolved in a later release.

## I try to run `ccollab` but it says File Not Found.

Under Windows, sometimes the installer is not able to update the PATH variable due to system configuration or permissions. Or the PATH variable is updated but the system requires a log-out or log-in before it is in the PATH.

Under Unix systems, the PATH variable needs to be updated if you did not install in a standard location such as `/bin` or `/usr/bin.`

## I try to upload files managed by an SCM system but `ccollab` fails to detect a valid SCM configuration

You should call the upload commands from any sub-folder within the working copy of your repository (the place on your local machine where files from an SCM system are checked-out).

## Running with debug logging enabled

Running the command-line client with the --debug[516] option will create a log file at the following location: `~/.smartbear/log/ccollab.log` on Unix platforms and `%USERPROFILE%/.smartbear/log/ccollab.log` on Windows platforms. If you contact technical support with a problem and they cannot reproduce it, they will typically ask you to do this and send them that debug log.

## 5.5.6 Global Options Reference

### Description

The following topic lists the available command-line global options.

Global Options can be specified on the command-line, or they can be saved and applied to every command using the `ccollab set` command. Global options specified explicitly for a particular command-line call override the values saved by the `ccollab set` command.

### General Options

| Option | Description |
|---|---|
| `--debug` | Run in debug mode, create a log file at the following location: `~/.smartbear/log/ccollab.log` on Unix platforms and `%USERPROFILE%/.smartbear/log/ccollab.log` on Windows platforms. |
| `--editor <value>` | External editor to use for editing file list |
| `--editor-prompt` | Prompt user for input to signal editing of file list complete |
| `--no-browser` | Do not pop up a web browser when the command completes |
| `--non-interactive` | Do not interact with user |
| `--overwrite-trust-store-in-ssl` | Overrides the `java.net.ssl.truststore` VM option with hard coded default values ($JAVA_HOME\lib\security\cacerts).<br><br>Restart client to apply this option. |
| `--password <value>` | Password or login ticket 318 to use when connecting to the Collaborator server.<br><br>When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead. |
| `--pref-dir <value>` | Overrides the location of Collaborator configuration files 495.<br><br>The path to preference folder must be specified either as relative path (a subfolder of the SmartBear profile folder), or as absolute path.<br><br>The folder should allow read/write access to files. |

| Option | Description |
| --- | --- |
| `--quiet` | Do not display progress messages |
| `--scm <value>` | Manually sets the SCM system type, for example, "perforce" or "none" for no SCM System. Clearing this option causes the SCM system to be auto-detected |
| `--server-proxy-host <value>` | Proxy server URL to use to connect to the Collaborator server |
| `--server-proxy-port <value>` | Proxy server port number to use to connect to the Collaborator server |
| `--url <value>` | URL to use when connecting to the Collaborator server |
| `--user <value>` | Username to use when connecting to the Collaborator server |

## AccuRev-specific Options

| Option | Description |
| --- | --- |
| `--accurev-depot <value>` | AccuRev depot name |
| `--accurev-exe <value>` | Full path to the `accurev` command-line client |
| `--accurev-anc-algorithm <value>` | Which algorithm to use when calculating the predecessor, either predecessor, previous-occupant, or basis-version |

## ClearCase-specific Options

| Option | Description |
| --- | --- |
| `--ccrc-server-url <value>` | CCRC server URL (for ClearCase Remote Client integration only) |

| Option | Description |
| --- | --- |
| `--ccrc-user <value>` | CCRC username (for ClearCase Remote Client integration only) |
| `--ccrc-passwd <value>` | CCRC password (for ClearCase Remote Client integration only) |
| `--clearcase-exe <value>` | Full path to the `cleartool` command-line client |
| `--clearcase-update-snapshot` | Whether to update ClearCase snapshot views prior to uploading files for review |

## CVS-specific Options

| Option | Description |
| --- | --- |
| `--cvs-exe <value>` | Full path to the `cvs` command-line client |
| `--cvsroot <value>` | Connection to the CVS repository |

## Git-specific Options

| Option | Description |
| --- | --- |
| `--git-exe <value>` | Full path to the 'git' command line client |
| `--git-skip-index` | Defines whether the changes you upload via the `ccollab addchanges` command should be taken from the working tree (including not-yet-committed changes) instead of from the index. This is an analogue of the `git commit -a` command. |

## Mercurial-specific Options

| Option | Description |
|---|---|
| `--mercurial-exe <value>` | Full path to the 'mercurial (hg)' command line client |

## Perforce-specific Options

| Option | Description |
|---|---|
| `--p4 <value>` | Full path to the P4 executable |
| `--p4port <value>` | How to connect to the Perforce server |
| `--p4user <value>` | Perforce user name |
| `--p4passwd <value>` | Perforce password or ticket |
| `--p4client <value>` | Mapping of Perforce server data to the local machine |
| `--p4-ignore-integration-history <value>` | Ignore integration history when calculating predecessor |
| `--p4charset <value>` | Perforce character set used for translation of Unicode files |
| `--p4-require-empty-default-changelist` | If true, do not allow uploads if the default changelist contains files |
| `--p4-specify-command-charset <value>` | Should a character set be specified for communication with Perforce |

## PTC Integrity-Specific Options

| Option | Description |
|---|---|
| `--mks-host <value>` | PTC Integrity server name. |
| `--mks-port <value>` | PTC Integrity server port. |
| `--mks-user <value>` | PTC Integrity user name. |
| `--mks-passwd <value>` | PTC Integrity user password. |
| `--mks-expand-keywords` | Specifies whether to expand keywords in source files. |

The arguments use the *--mks* prefix as MKS Integrity is a former name of PTC Integrity.

## Rational Synergy-specific Options

| Option | Description |
|---|---|
| `--ccm-exe <value>` | Full path to the `ccm` command-line executable |
| `--ccm-user <value>` | User Name to use when starting a Rational Synergy session |
| `--ccm-passwd <value>` | Password to use when starting a Rational Synergy session |
| `--ccm-engine-host <value>` | Host the Rational Synergy Engine will run on |
| `--ccm-database-path <value>` | Path of the Rational Synergy database to connect with |

| Option | Description |
| --- | --- |
| `--ccm-local-database-path <value>` | Path of the local Rational Synergy database, typically c:/temp/ccm or /tmp/ccm |
| `--ccm-remote-client` | Start Rational Synergy sessions as a Remote Client |
| `--ccm-server-url <value>` | Server URL for Web-Mode Rational Synergy servers |

## Rational Team Concert-specific Options

In order to use Rational Team Concert integration from command-line client, you need to configure both Collaborator server and clients and RTC server. See configuration instructions 713.

| Option | Description |
| --- | --- |
| `--rtc-repository-uri <value>` | The URI of the repository to work with. |
| `--rtc-username <value>` | The user name to use for the chosen RTC repository. |
| `--rtc-password <value>` | The password of the user. |

## Subversion-specific Options

| Option | Description |
| --- | --- |
| `--svn-exe <value>` | Full path to the `svn` command-line executable |
| `--svn-look-exe <value>` | Full path to the `svnlook` command-line executable (used by Subversion triggers) |

| Option | Description |
|---|---|
| `--svn-repo-url <value>` | Subversion repository URL |
| `--svn-user <value>` | Subversion user name |
| `--svn-passwd <value>` | Subversion password |
| `--svn-require-client-certificate-password <value>` | Use this if you have a non-empty and unsaved password for your SSL Client Certificate |
| `--svn-auto-add` | Treat unversioned files as if they had been added to Subversion |
| `--svn-recurse-externals` | Recurse in the 'svn:externals' directories as if they were part of the same repository |
| `--svn-repo-path <value>` | Full path to the repository (used by Subversion Triggers) |

## Team Foundation Server-specific Options

Client configuration settings depend on whether you use a self-hosted version or a SaaS version of Team Foundation Server known as Visual Studio Team Services. If you use the latter, then you will need to set-up alternate authentication credentials at first and then use these credentials for Collaborator clients. See configuration instructions 741.

| Option | Description |
|---|---|
| `--tfs-collection <value>` | For self-hosted version of Team Foundation Server, specify the URL of Team Foundation Project Collection to work with.<br><br>For SaaS version of Team Foundation Server, specify the URL of your Visual Studio Team Services account (without project or collection names). |

| Option | Description |
|---|---|
| `--tfs-user <value>` | The name of Team Foundation user. For SaaS version of Team Foundation Server, specify alternate primary user name. |
| `--tfs-passwd <value>` | The password of the user. For SaaS version of Team Foundation Server, specify alternate password. |

### 5.5.7 Command-line Reference

## Description

Use the `ccollab` command to perform a number of Collaborator tasks from the command-line.

## Command Line Syntax:

```
ccollab [global-options] command [command-options]
```

## Sub-Commands

| Sub-Command | Description |
|---|---|
| `help` | Display help on using the Command Line Client |
| `info` | Validates server connection and SCM configuration |
| `login` | Verify and/or change connection to the server |
| `logout` | Clears the stored password and/or login ticket id, and invalidates the user's login ticket on the server |
| `set` | Save a global option setting |
| `addchanges` | Attaches locally-modified files to a review |
| `addfiles` | Attaches local files to a review without diffs |
| `addchangelist` | Attaches an atomic changelist to a review |

| Sub-Command | Description |
|---|---|
| adddiffs | Attaches file differences to a Review |
| addsvndiffs 816 | Uploads diffs generated from the svn diff command |
| addardiffs 632 | Uploads diffs generated from accurev diff command |
| addcvsdiffs 643 | Uploads diffs generated from cvs diff command |
| addgitdiffs 660 | Uploads diffs generated by git diff command |
| addhgdiffs 736 | Uploads diffs generated by hg diff command |
| addp4diffs 779 | Uploads diffs generated from p4 diff2 command |
| addversions | Attaches any 2 given versions to a review |
| addactivity 688 | Attaches file versions in a ClearCase activity to a review |
| actionitems | List current action items |
| addsimulinkarchives | Uploads Simulink model archive to a review |
| addstream 634 | Attaches pending differences from an AccuRev stream |
| browse | Launch a browser to the Collaborator Server homepage |
| commit | Commit changes in the review |
| addp4job 785 | Adds all numbered local changes that fix a job to the review |
| addurls | Attaches urls to a review |
| admin | Perform administration tasks |
| gitaddbranch 661 | Uploads diffs between the specified git branch and the remote-tracking branch. |
| pinCoordinates | Returns a list of pushpin coordinates and numbers for the particular file revision of the specified review. |

## 5.6 IDE Clients

Collaborator provides a number of plugins for most popular integrated development

environments (IDEs).

These plugins allow you to create Collaborator reviews directly from the user interface of your favorite IDE.

See the section below for your IDE:

- Eclipse 525 (including, Eclipse-based applications)
- Microsoft Visual Studio 566

**Important:** Collaborator IDE Clients are designed to perform code reviews, rather than document reviews. That is, from within the IDE you can review and compare source code and text files, but may fail to review and compare other types of review materials. To review Word documents, Excel tables, PDF files, images, or URLs you should use the Collaborator Web Client 312.

### 5.6.1 Eclipse Plug-in

The Collaborator Eclipse Plug-in works with many Eclipse-based applications.

## Features

- View with Action Item display 537
- Perform Reviews 538 without leaving Eclipse
- Label Decorations 544 for local files under review
- Markers 544 on local files for Defects in Reviews
- Upload files 545 from local hard drive
- Integrates with existing source control plug-ins including AccuRev 553, ClearCase 553, CVS 553, Git 554, Perforce 560, Rational Team Concert 700, and Subversion 563.
- Update site 526 for automatic plug-in install and upgrade.

## Compatibility

Eclipse base version 4.6 or higher (released versions only).

Eclipse-based applications known to work with the plug-in:

- Eclipse SDK
- IBM WebSphere

## Known Issues

- Conversations View in Eclipse Oxygen does not split long line of text into multiple lines, because of this a horizontal scroll bar can appear if your conversation of checklist item contains long text.
- Collaborator Eclipse Plug-in is shipped with the Google Guava library version 20. Other Eclipse plug-ins may use another versions of this library and in some rare cases this may cause dependency conflicts.

#### 5.6.1.1 Install & Update

## Update Site

Eclipse can be configured to automatically download the Collaborator plug-in, and even to check periodically for updates. If you already know how to do this, all you will need to know is that our Eclipse update site is located here:

http://eclipse.smartbear.com/13.0

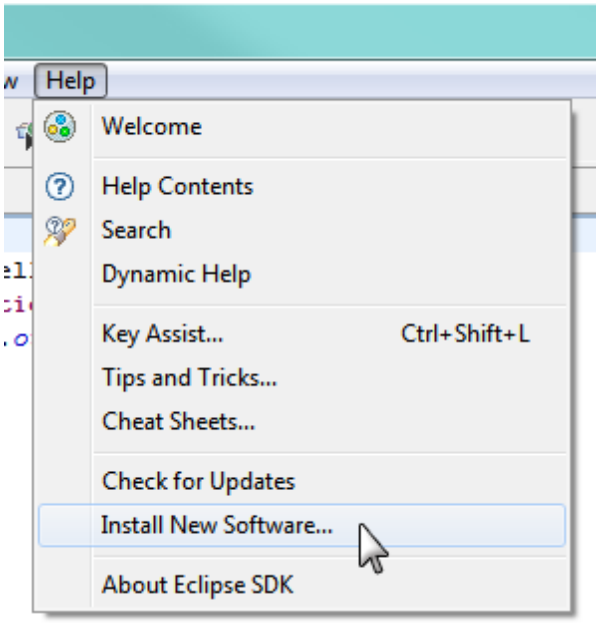
Otherwise, read on for step-by-step installation instructions.

## Installation Instructions

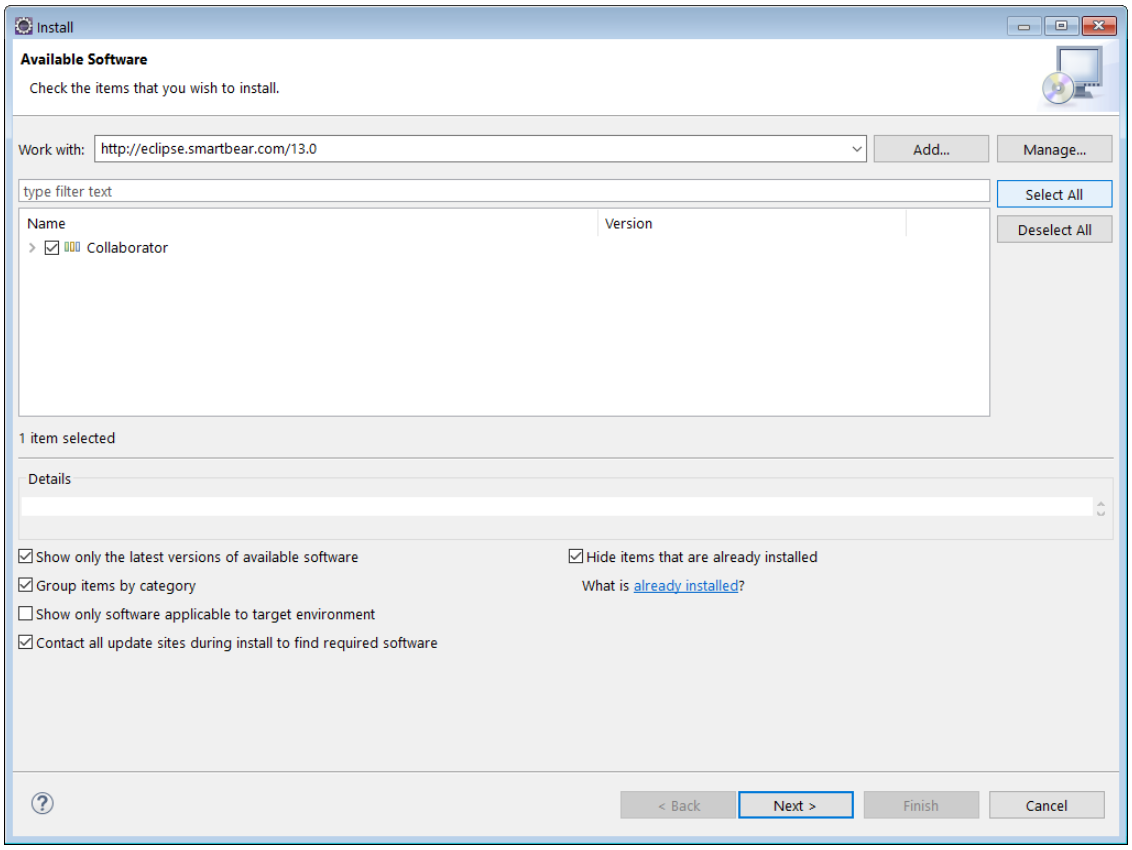
Here are detailed instructions for how to install our plug-in in Eclipse:

1. Open Eclipse IDE.

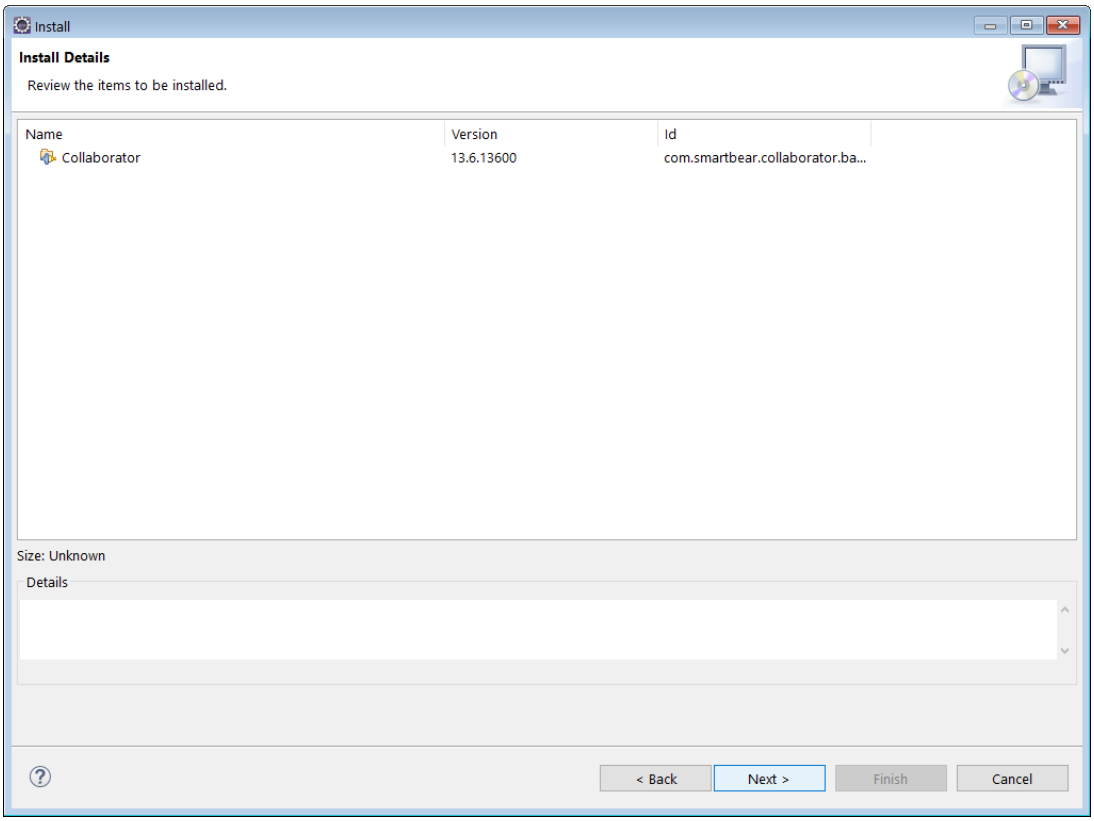
2. Select **Help | Install New Software** from the main menu.



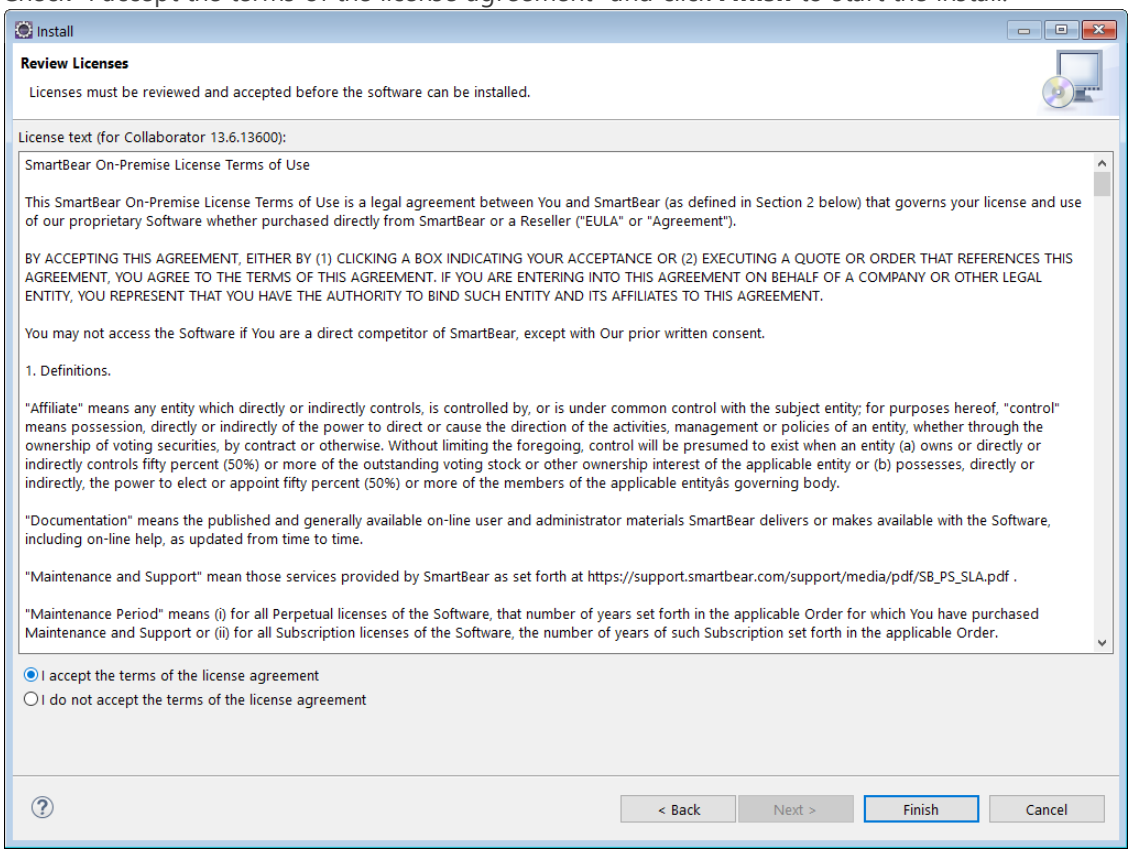
3. Type "http://eclipse.smartbear.com/13.0" in the **Work with** field and press Enter. Then check "Collaborator" and click **Next**.




4. Click **Next** to finish the "Install Details" page.

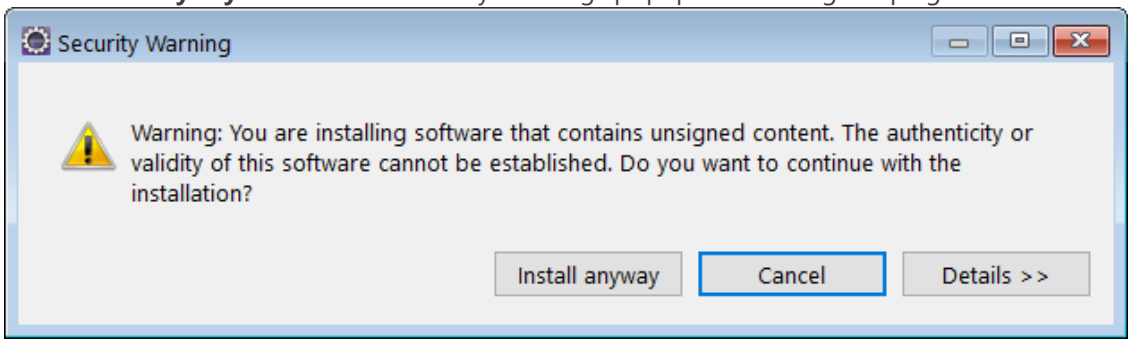


5. Check "I accept the terms of the license agreement" and click **Finish** to start the install.




6. A progress dialog appears as Eclipse installs the plug-ins.

7. Click **Install anyway** to close the "Security Warning" popup about unsigned plug-ins.




8. Click **Yes** in the "Software Updates" popup to restart Eclipse.

9. **Done!**

## Uninstallation Instructions

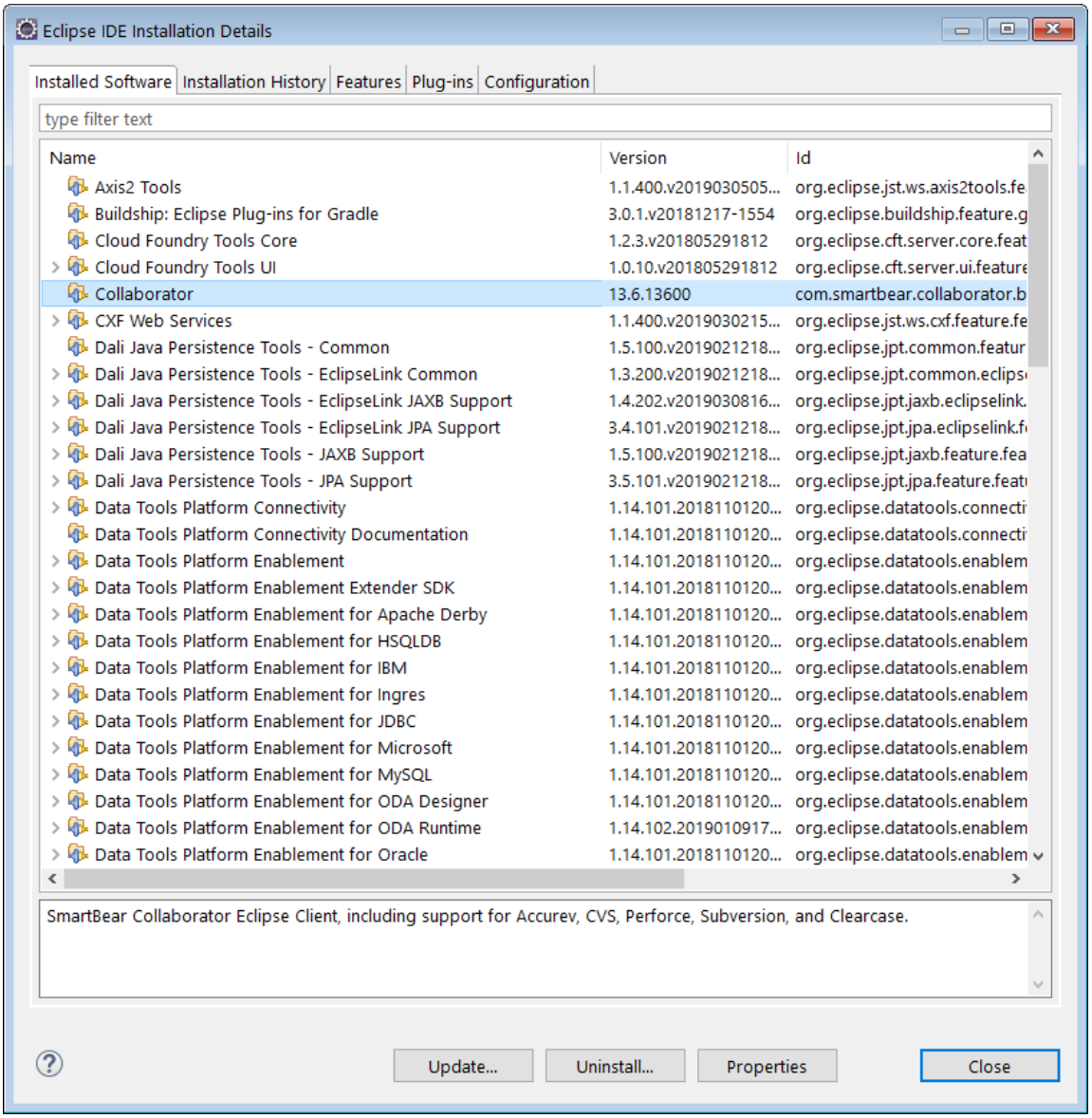
To uninstall the plugin:

1. Open Eclipse IDE.

2. Select **Help | About** from the main menu. Click **Installation Details.** This will open a dialog with more details about your installation.

3. Click the **Installed Software** tab to see a list of the software items that you have installed into your system.

4. Select Collaborator and click **Uninstall.**



Additionally you will need to remove the Collaborator's Eclipse Plug-in files from the Eclipse plugins folder. This is especially important if you plan to re-install the plugin later.

1. Go to Eclipse plugins folder. It is typically located at the following path:

```
<user_home>/.p2/pool/plugins
```

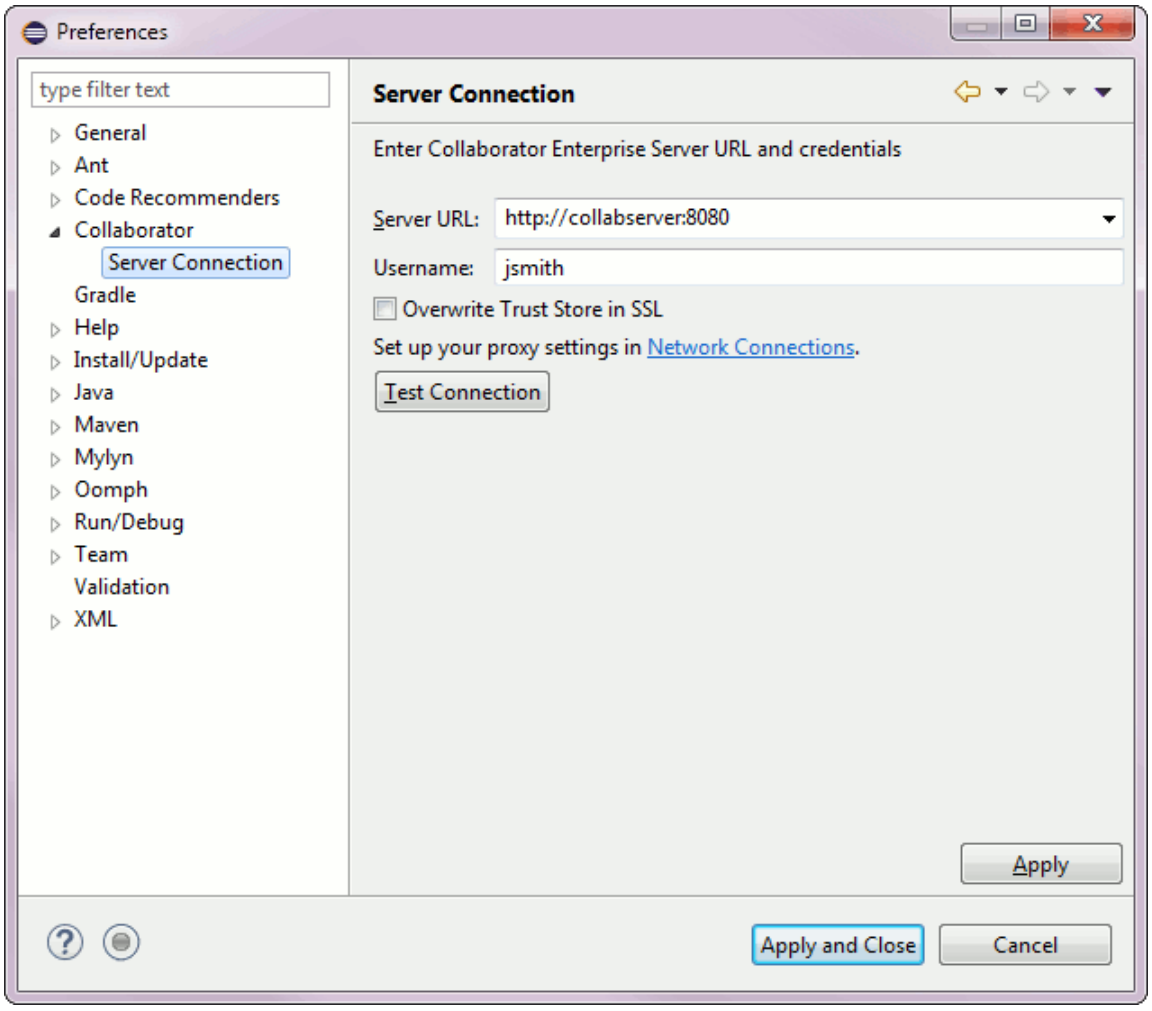
2. Select all jar-files and folders that start with **com.smartbear.collaborator**.

3. Delete these files and folders from the Eclipse plugins folder.

#### 5.6.1.2 Preferences & Configuration

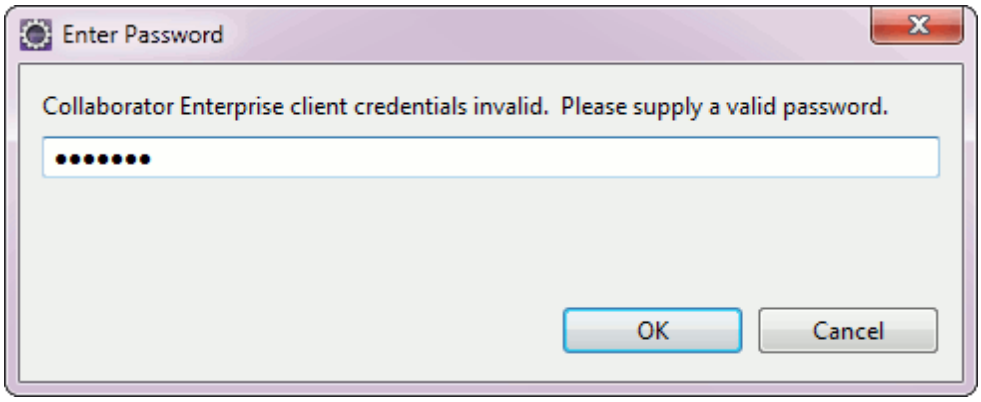
You must configure your connection to the Collaborator server; all other preferences are optional.

You can find the Eclipse Plug-in configuration in the Preferences dialog along with all other application preferences:



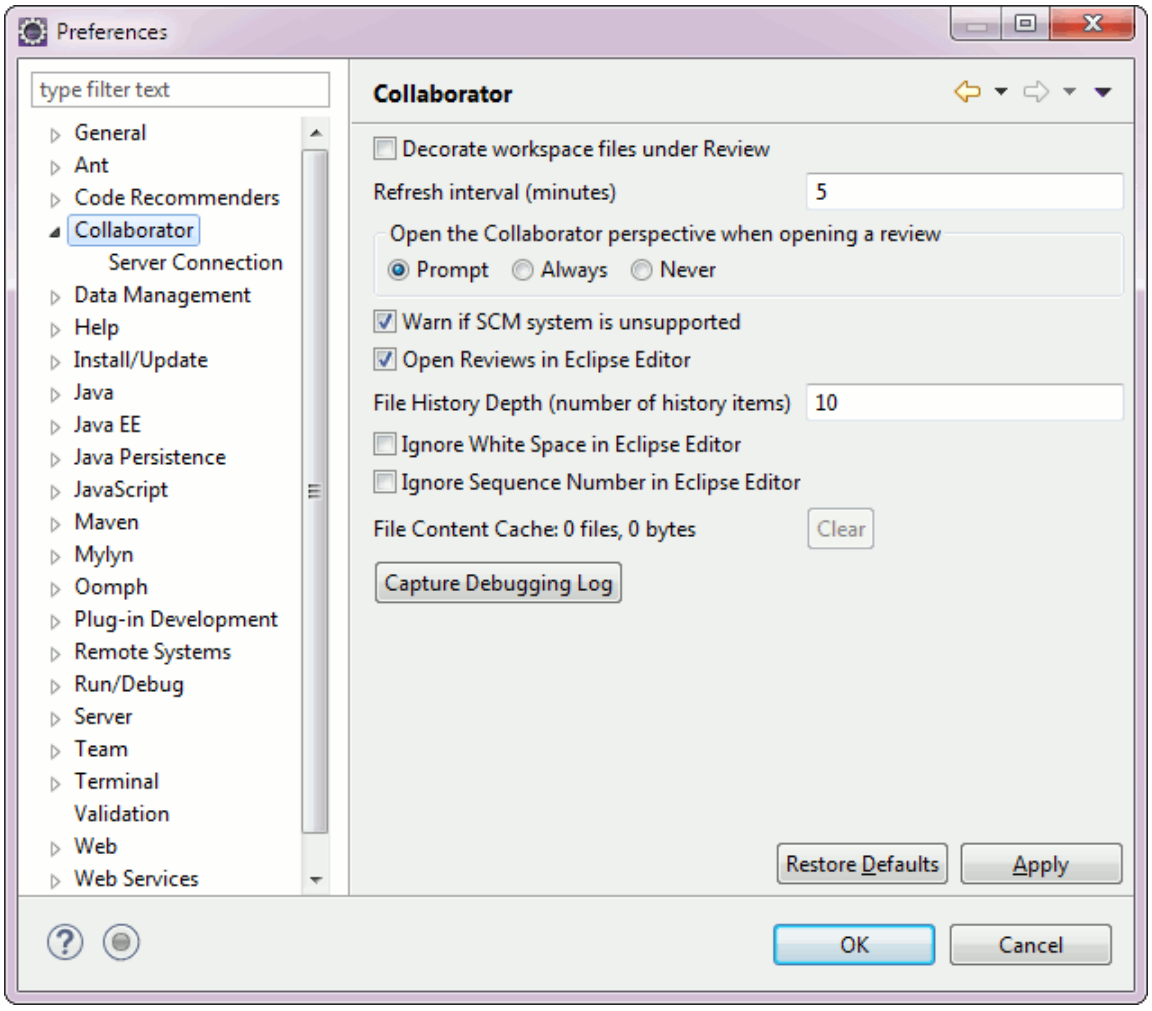

The Server URL must include the correct port number and path if applicable. The Username and Password are the same as you use when logging into the web server 315.

The password entry is displayed on the following connection form. You can specify your password or login ticket 318. When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead:




Also, you may specify whether Eclipse Plug-in should override the `java.net.ssl.truststore` VM option with hard coded default values ($JAVA_HOME\lib\security\cacerts). The latter option requires restarting the Eclipse in order to apply.

Use the Test Connection button to make sure the connection is working. If it fails the error message will be helpful.




**Decorate workspace files under review**

Specifies whether to enable Label Decorations 544 for files which are currently under review.

**Refresh interval**

How often to refresh the Action Items View 537 with data from the server. Markers 544 and Label Decorations 544 are also refreshed at this time.

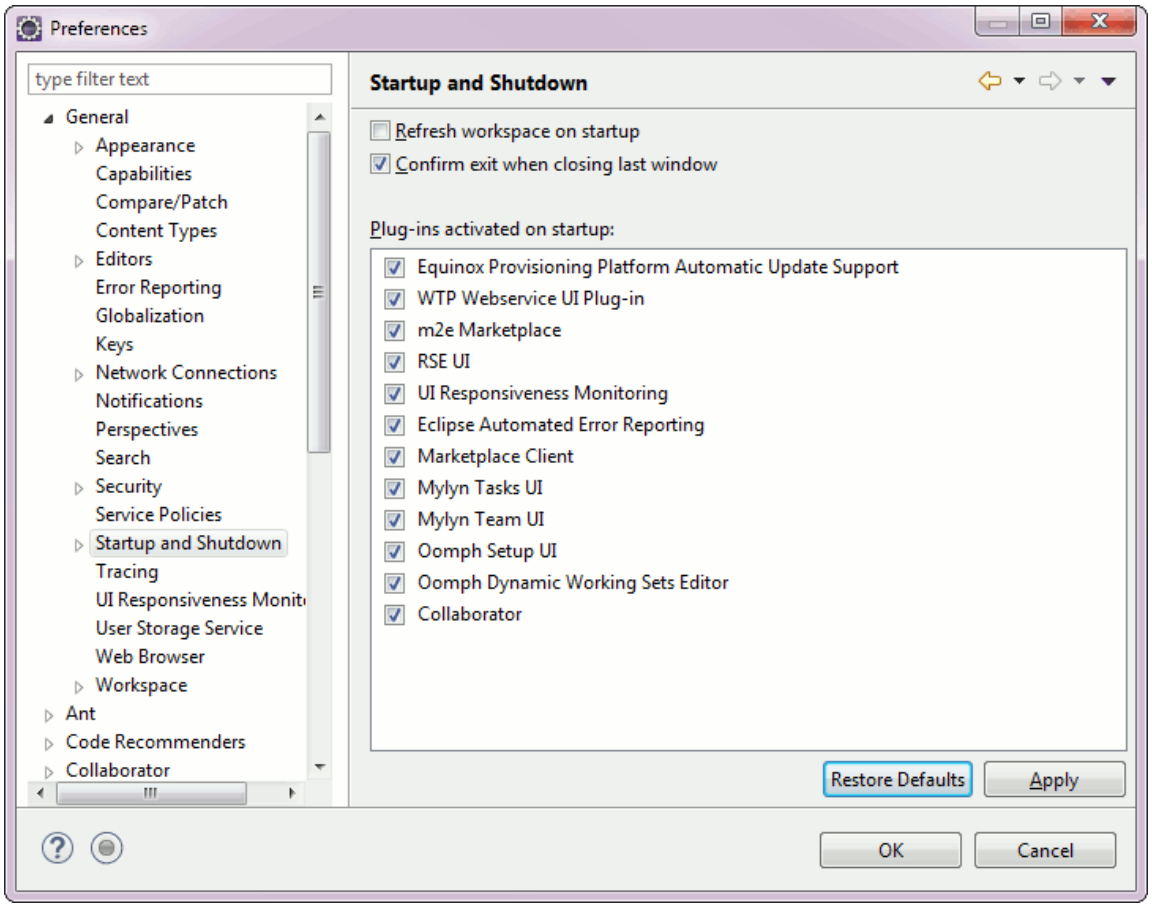
**Open the Collaborator Perspective when opening a Review**

Automatically switch to the Collaborator Perspective 537 when opening a Review.

**Warn if SCM system is unsupported**

When adding files to a Review 551, warn if SCM system (or client) is not supported.

| | |
|---|---|
| **Open Reviews in Eclipse Editor** | Open Reviews in the Eclipse Review Editor 538. If this option is not checked, Reviews will be opened in a browser. |
| **File History Depth** | Specifies the number of file revisions to keep. |
| **Ignore White Space in Eclipse Editor** | Controls whether white-spaces are taken into account when showing differences in Eclipse Compare Editor 543. |
| **Ignore Sequence Number in Eclipse Editor** | Controls whether COBOL sequence numbers are taken into account when showing differences in Eclipse Compare Editor 543. |
| **File Content Cache** | The estimated size of the local file content cache. Press Clear to delete the local content cache (content will be automatically downloaded from the server as needed). |

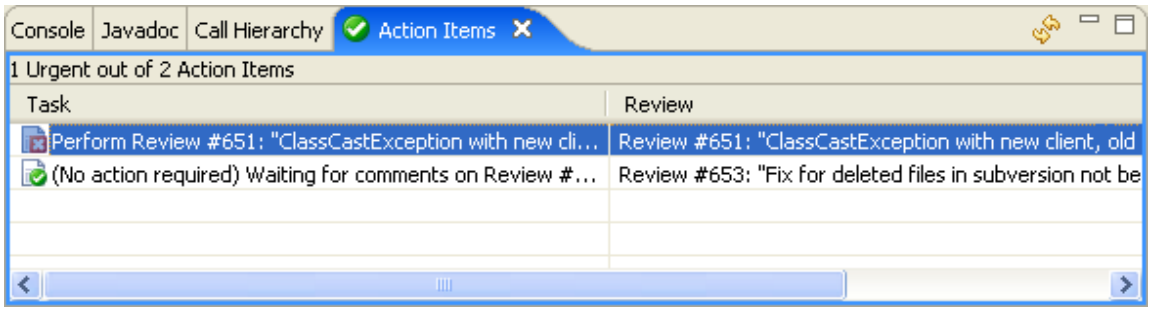



The Collaborator Eclipse Plug-in registers itself with Eclipse to be started automatically when Eclipse starts up. This is necessary to refresh Markers 544 and Label Decorations 544 automatically. If you un-check Collaborator from the Plug-ins activated on startup list then Markers 544 and Label Decorations 544 will not appear until the first time you show the Action Items View 537.

### 5.6.1.3 Action Items

To open your Collaborator Action Items in Eclipse, select Window -> Show View -> Action Items.

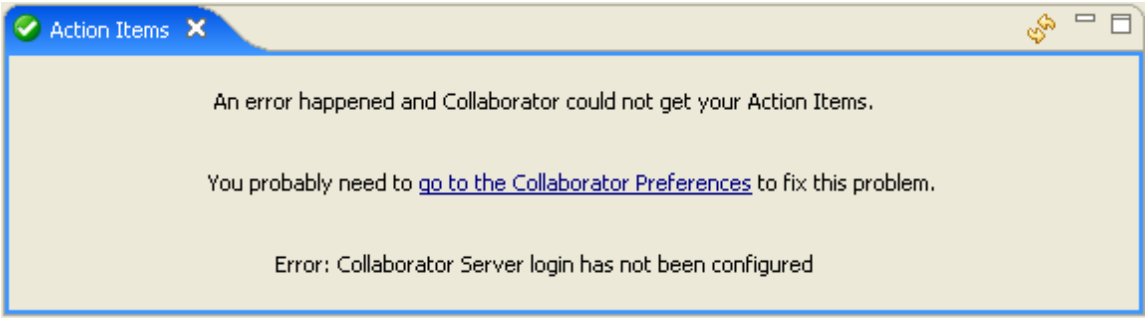
Your Collaborator Action Items appear in Eclipse as a View:




Double-clicking any item will open the Review in the Review Editor 538 or your configured web browser, depending on your preferences 536.

The list of Action Items automatically refreshes periodically (the rate is configurable 535). You can refresh manually using the double-arrow icon in the toolbar. Markers 544 and Label Decorations 544 refresh at the same time as Action Items.

If there was an error communicating with the server, the View will display an error message instead and you might need to visit your preferences 532 to fix connection settings:




### 5.6.1.4 Collaborator Perspective Window

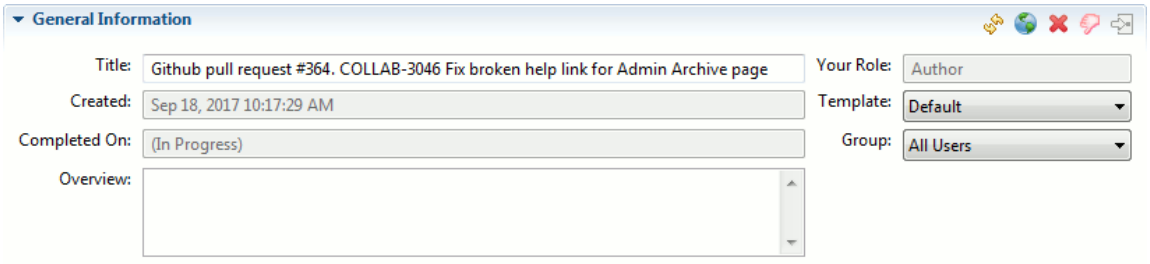
The Collaborator Perspective window is useful for organizing your screen to perform reviews in Eclipse. By default it contains the Editor Area, the Conversations View 541, and the Action Items View 537.

The Collaborator Perspective window can be configured 535 to open automatically when you open a Review from the Action Items View 537.

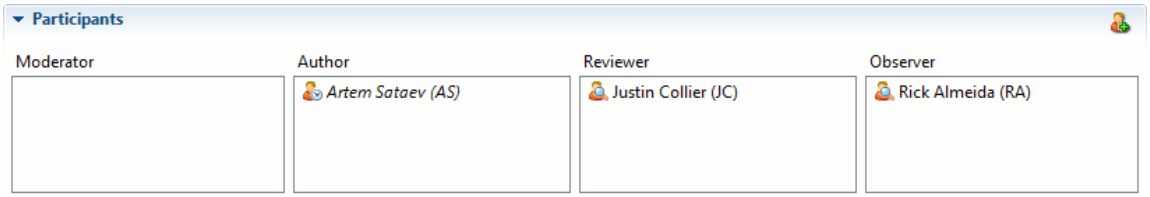
### 5.6.1.5 Review Editor

Double-clicking an Action Item[537] will open the associated Review in the Review Editor. (If you prefer, you can configure your preferences[536] to open the Review in a web browser instead.) This editor lets you perform a Review without leaving Eclipse. The content of the editor updates automatically as the Review changes (other users make comments, new files are uploaded, and so on).

When the Review Editor is active, the Conversations View[541] will show the overall Review conversation.

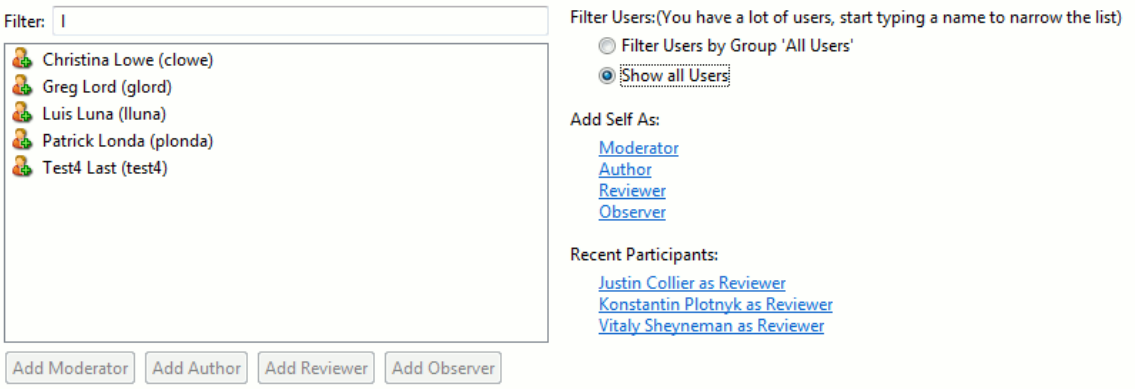



The General Information[348] section shows you information about the Review as a whole, and lets you edit certain fields, including Review custom fields[256]. Toolbar buttons in this section allow to refresh review information, open review in a web-browser, cancel, reject or re-open the review.

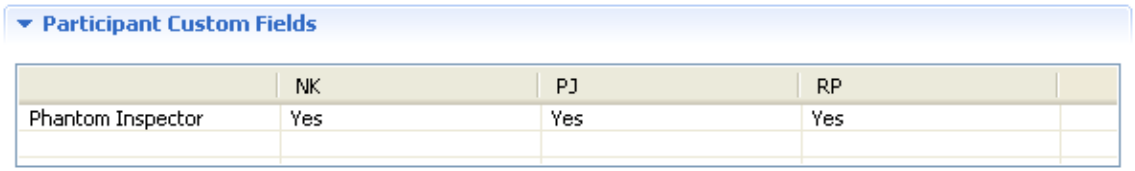



The Participants[352] section shows the users participating in this Review and their Roles. You can change a user's Role by dragging-and-dropping them to a different Role. You can remove a user from the Review by right-clicking and selecting Remove from the context menu.

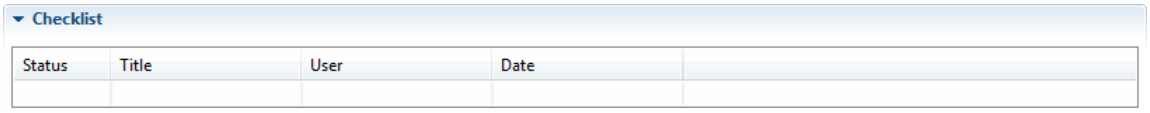
Click the Add Participants... icon in the toolbar of the Participants section to toggle the section for adding a new Participant to the Review.



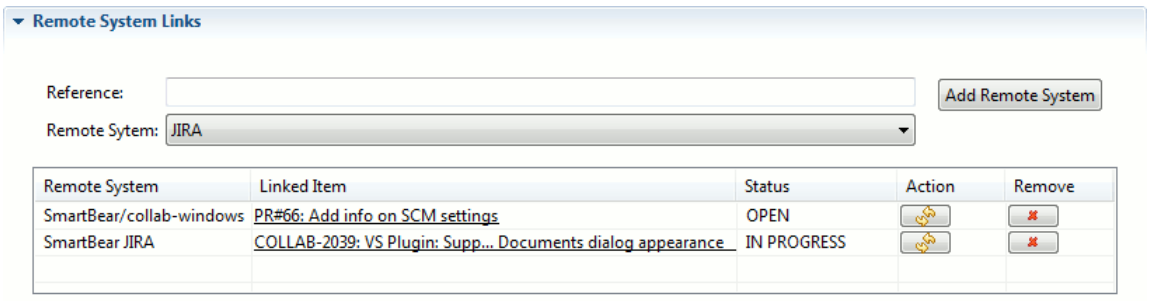

Drag users from the list to one of the Role lists above to add them to the Review in that Role. You can also click one of the **Add <Role>** buttons to add the selected users to the Review in that Role. Click on the links in the **Recent Participants** area to quickly assign recent participants to the Review. Typing in the **Filter** field will filter the users list to help you find the users you want.



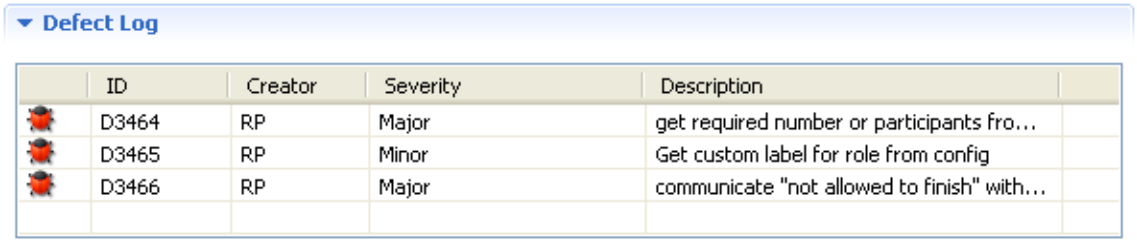

The Participant Custom Fields section shows the value of Participant Custom Fields 256 for each Participant. Note this section only appears if the Template 246 of this Review has Participant Custom Fields configured.



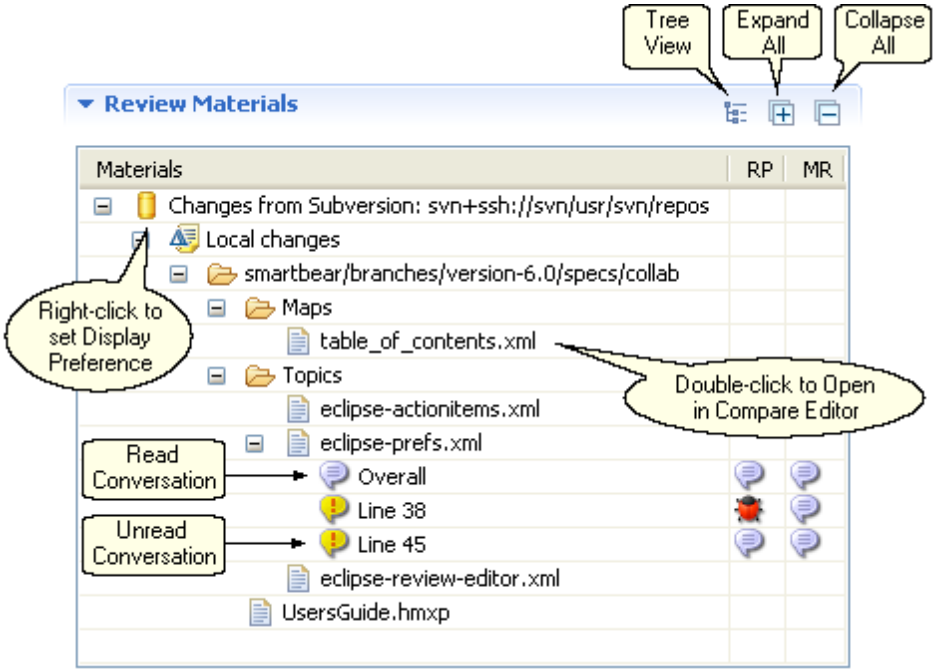

The Checklist 350 section displays the checklist associated with the review (if any).



The Remote System Links 353 section lists all pull requests, direct pushes and issue-tracking items linked with this review. Because of a limitation of the SWT control, the Remote System Link table cannot have multi-line cells. When multiple items are linked (for example, multiple commits), the table displays only the first item. The full list of linked items will be displayed in the cell's tooltip and in the Overall Chat section.




The Defect Log 354 section shows all of the Defects in the Review, both overall Review Defects and Defects on files.




The Review Materials 357 section shows the changelists and files which are part of the Review. Double-click on a file to open it in the Compare Editor.

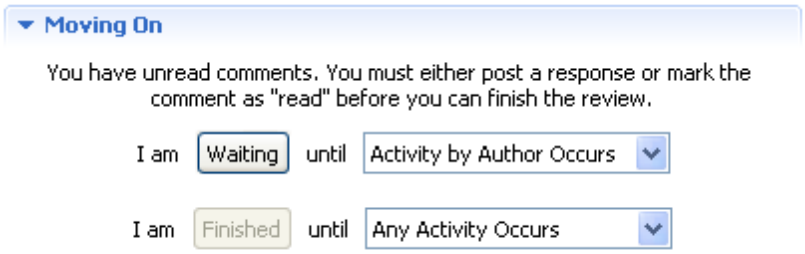
The section toolbar controls how the materials are presented:

| | |
|---|---|
| **Tree View** | The presentation of folders. You can select from **Compressed Tree**, **Tree**, and **Flat**. |
| **Expand All** | Expand all levels in the tree. |

**Collapse All** Collapse all levels in the tree.

Right-click on the "Version Control" icon to set your Display Preference for that Version Control system. Depending on the Version Control system and available changelists, you can select from Do not Roll Up, Roll Up All Changelists, and Roll Up Non-Atomic Changelists.

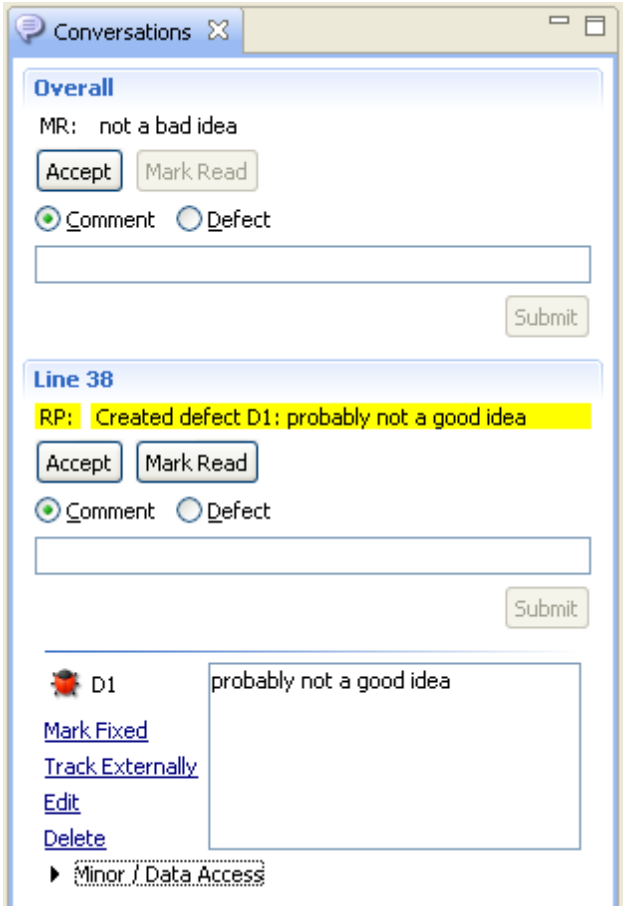
Conversations on files are listed underneath the files in the tree. If the conversation has unread comments the icon is highlighted in yellow.




The Moving On 347 section allows you to specify whether you are **Waiting** on Review activity or have **Finished** with the Review (with the option to re-engage based on future activities). These choices move reviews into different phases. The options in "Moving On" vary depending on the Phase of the Review and your Role. "Moving On" also allows you to control - to a certain extent - the notifications you receive for a particular Review.

#### 5.6.1.6 Conversations View

The Conversations View displays conversations on the most recently focused Review Editor 538 or Compare Editor 543. For the Review Editor 538 the Conversations View shows the overall Review conversation. For the Compare Editor 543 the Conversations View shows all the conversations on the file.




Unread Comments are highlighted with a yellow background. You can mark conversations Accepted or Read. You can add Comments. You can create or modify Defects. You can start a conversation at a new location or add to an existing one.

Right-click on a Comment and select Redact to redact it. Redacted comments are grayed-out.

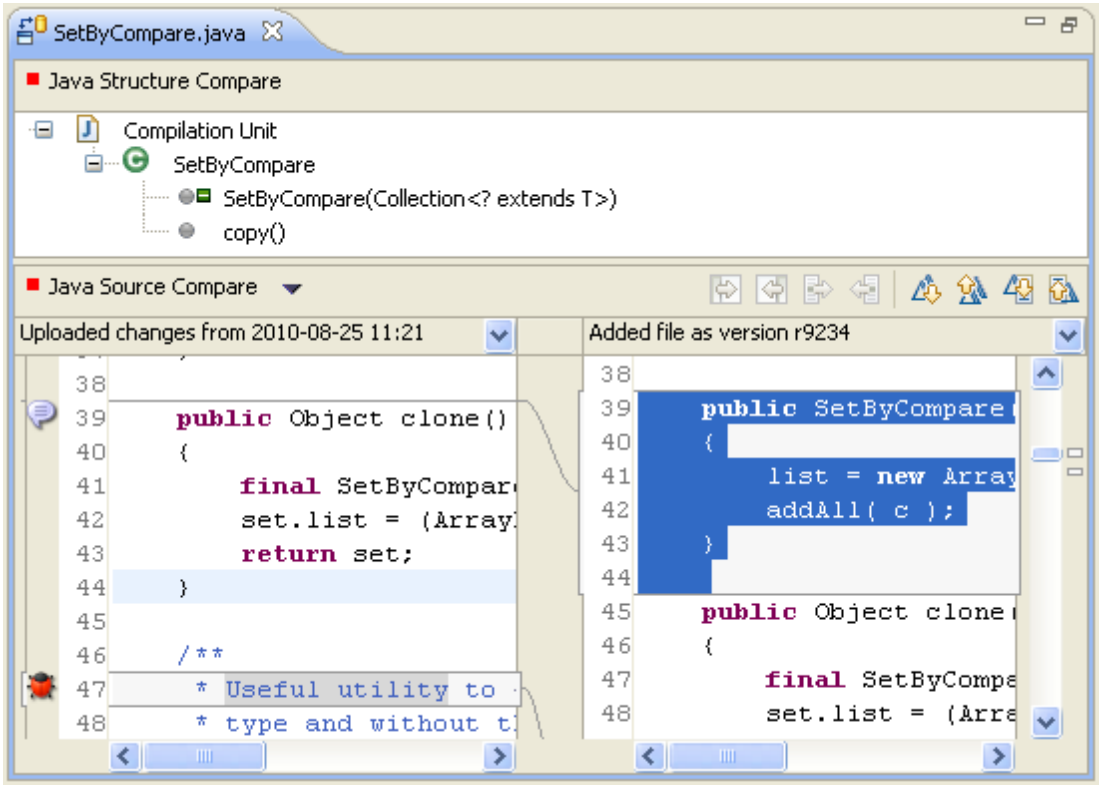
## Known Issue

Conversations View in Eclipse Oxygen does not split long line of text into multiple lines, because of this a horizontal scroll bar can appear if your conversation of checklist item contains long text.

### 5.6.1.7 Compare Editor

Double-clicking on a file in the Review Editor 540 opens it in the standard Eclipse Compare Editor.

**Important:** Collaborator Eclipse Client is designed to perform code reviews, rather than document reviews. That is, from the Compare Editor you can review and compare source code and text files, but may fail to review and compare other types of review materials. To review Word documents, Excel tables, PDF files, images, or URLs you should use the Collaborator Web Client 312.




Conversations on the file are shown in the Conversations View 541 when the Compare Editor is active.

The exact appearance of the Compare Editor depends on the type of file being compared. The Collaborator Eclipse Plug-in uses the compare viewer registered in your Eclipse installation for that file type, and then tries to add review-related controls on a best-effort basis.

## Side-by-side Compare Viewers

If the registered compare viewer for the file type is based on the standard Eclipse side-by-side compare viewer, the Collaborator Eclipse Plug-in adds a drop-down menu above each content pane to select which versions of the file you want to compare.

If the file was uploaded from your machine, you can select `Local File` from the menu to compare against the local version of the file. Comparing against the local version of the file lets you navigate the code in your workspace (for example, `ctrl-click` to open a type or method implementation), and edit the code in-place. You can also right-click on the pane containing the local file and select from many of the same context menu options that you can when you right-click on the file in the normal Eclipse editor.

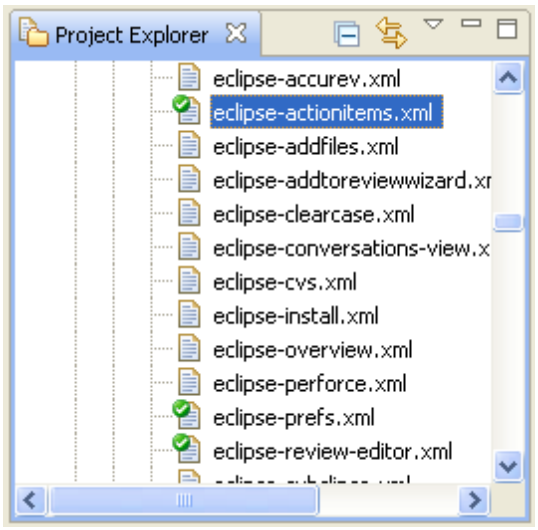
## Text Compare Viewers

If the registered compare viewer for the file type is based on the standard Eclipse text compare viewer, the Collaborator Eclipse Plug-in adds a ruler to the left content pane showing the state of each conversation on the file. Clicking on a line selects that conversation in the Conversations View 541, or starts a new conversation if there is not already one on that line.

Syntax highlighting is shown if it is supported by the registered compare viewer for the file type.

### 5.6.1.8 Label Decorations

A green check-mark is displayed on files in your workspace which are currently under review. This decoration is updated whenever your Action Items 537 refresh. You can disable this decoration using the standard Eclipse preferences page under General->Appearance->Label Decorations->Collaborator.




### 5.6.1.9 Markers

The Eclipse Plug-in creates Eclipse Markers for every Defect in a Review that is on a file that came from your computer. The Markers are updated when your Action Items 537 refresh.

## Problems View

Markers appear as Problems of different severity, depending on the Defect state.

- Open Defect -> Error
- Externalized Defect -> Warning
- Fixed Defect -> Info

## Marker Bar

When you open a file in an Eclipse text-based editor, Markers appear in the Eclipse Marker Bar. Different Icons are used for each Defect state.

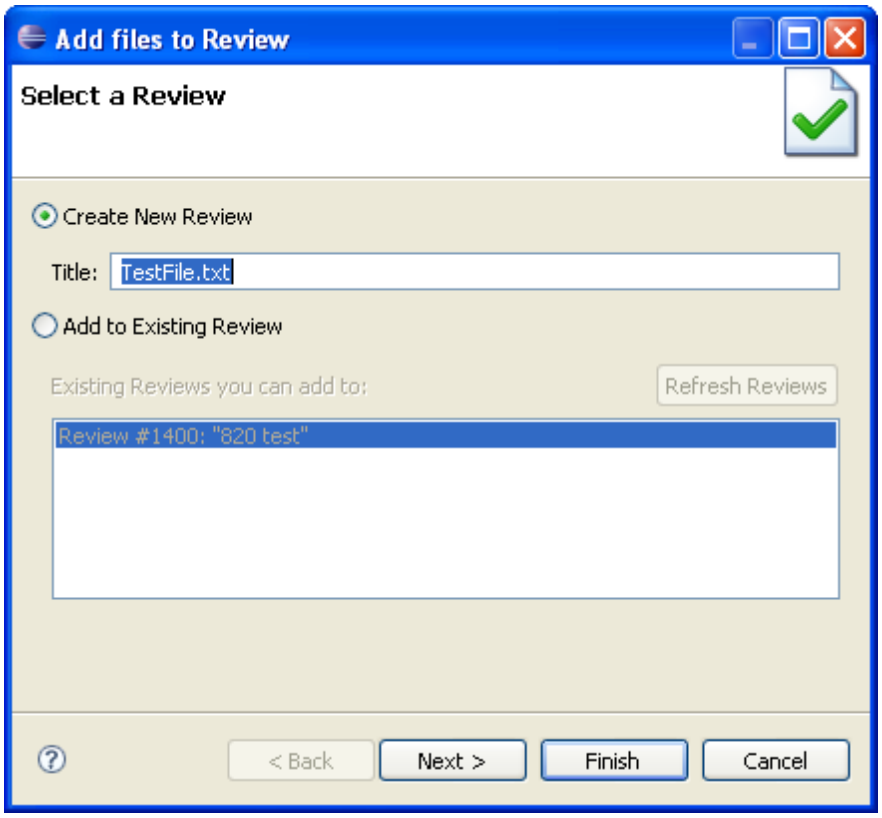
### 5.6.1.10 Add to Review Wizard

The Add to Review Wizard lets you add review materials to a new review or an already existing review.

The first page of the Add to Review Wizard asks you which review you would like to add materials to. You can create a new review and add the materials into the newly created review, or you can add the review materials to an existing review.

To add materials to a new review, select "Create New Review," and type a title to name your new review.

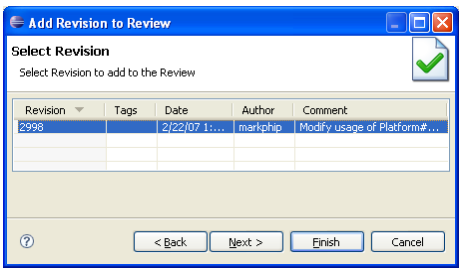
To add materials to an existing review, select "Add to Existing Review". Choose the review you would like to add the materials to. The list of possible reviews is created automatically by querying the server. This will generally be the same as the list of reviews in your Action Items 333 list.




At this point, you can `Finish` the wizard or click `Next` to get more options.

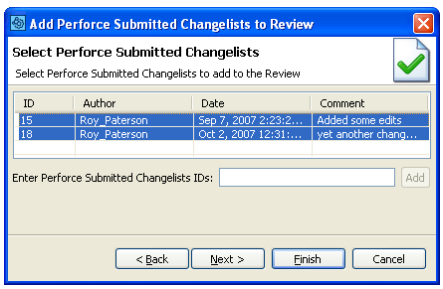
## Select Review Materials

The next page lets you choose exactly which materials are added to the review. This page looks different depending which materials you are adding to the review:




*Adding (Subversion) Revisions to a Review* 564

551

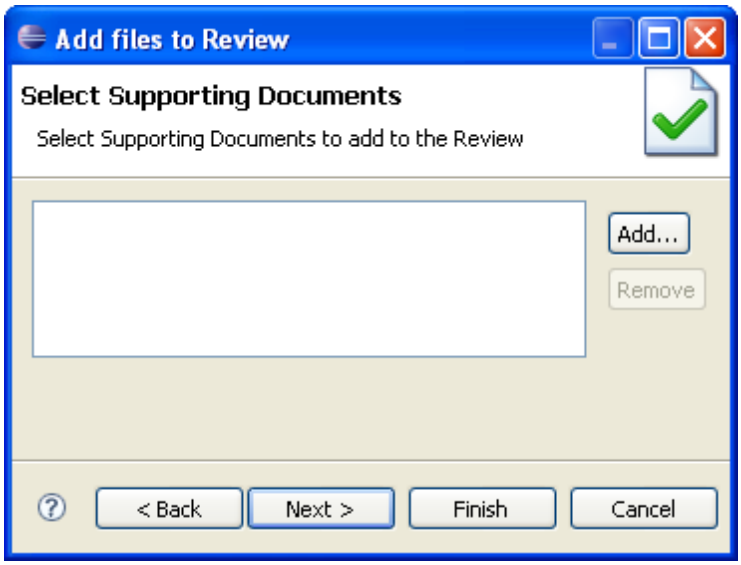



*Adding (Perforce) Submitted Changelists to a Review* 562

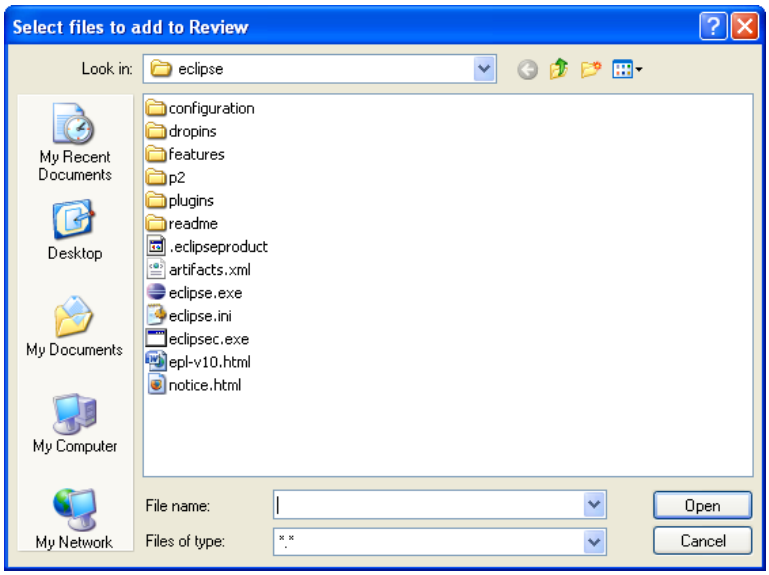
560

## Select Supporting Documents

The next page allows you to upload additional files from anywhere on your local hard drive:



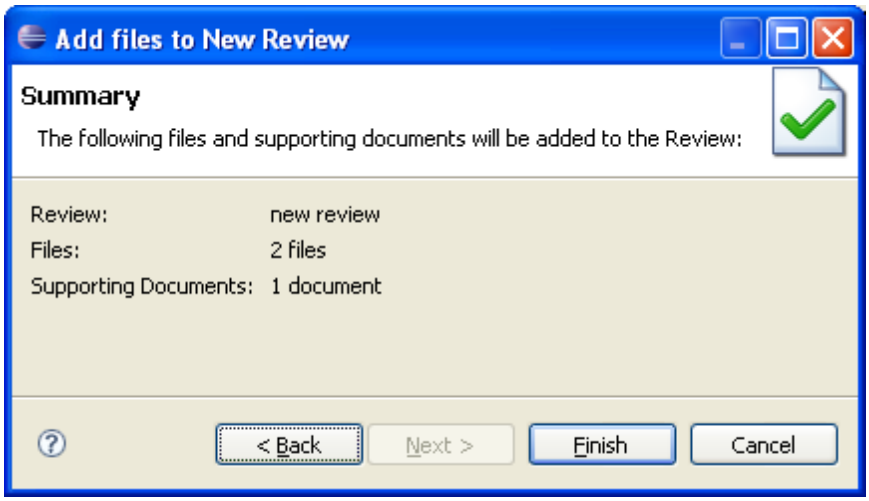

Click "Add" to choose the documents you would like to add:



You can also remove documents by selecting the document in the list above and clicking "Remove".

## Summary

The final page summarizes the selections of the previous pages:

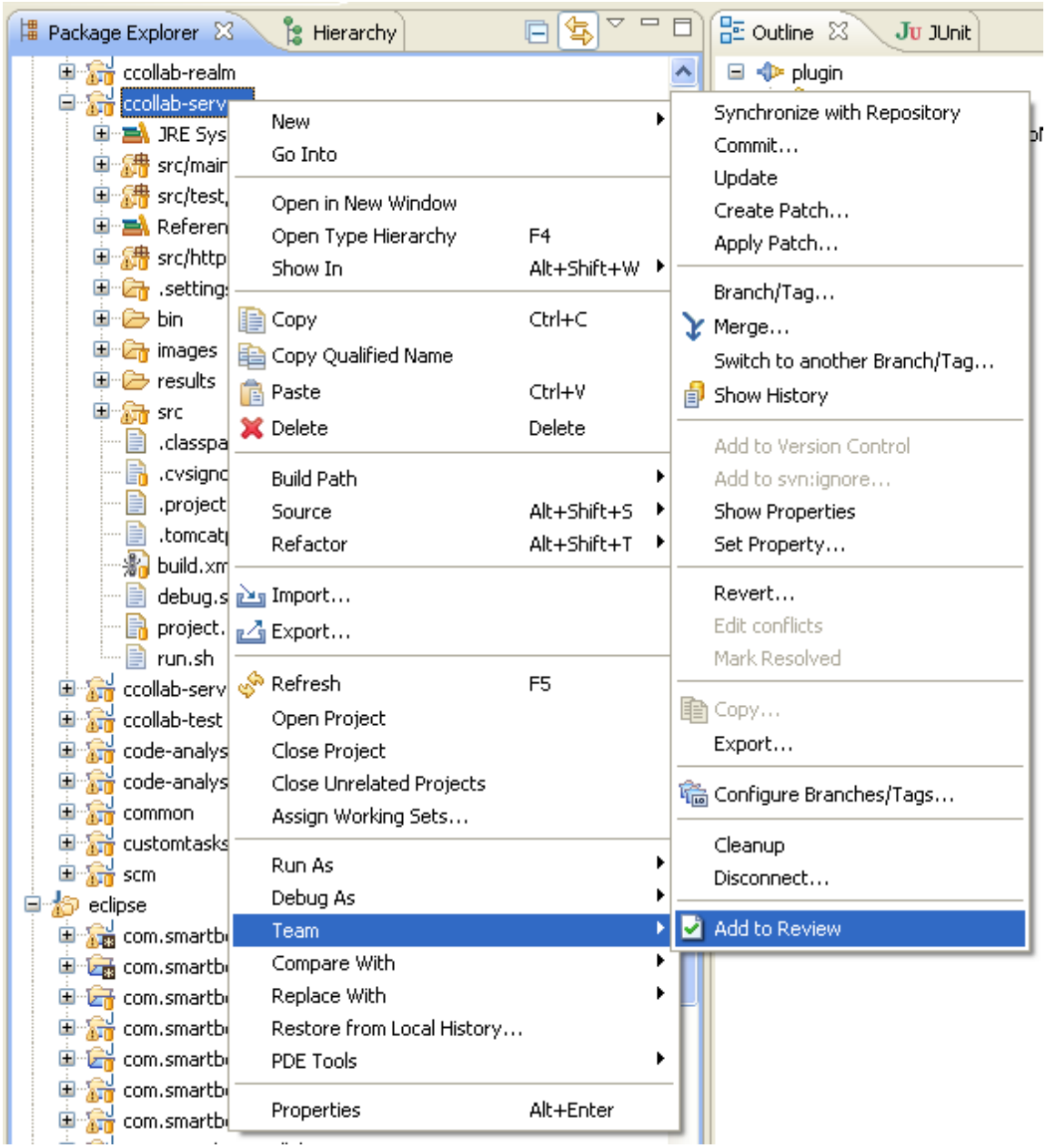



When the Finish button is clicked, all review materials are uploaded to the Collaborator Server for review. If uploading takes a long time, a progress dialog will appear during this operation. The progress dialog can be minimized if you do not want to wait for the operation to complete.

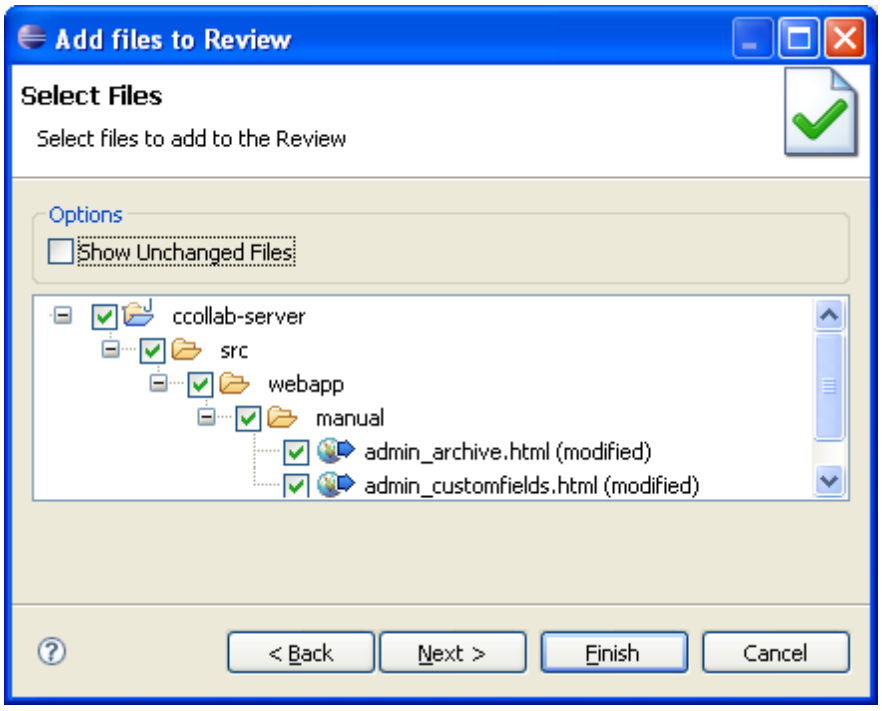
When the operation is complete the Review open in the Review Editor 538, or in the browser based on your preferences 536.

### 5.6.1.11 Adding Files to a Review

Right-click on any Working Sets, Projects, Folders, or Files and select either `Add to Review` from the `Team` menu.




Clicking Add to Review will launch the Add to Review Wizard[545]. The wizard will include a page that shows the list of resources you selected, allowing you to select exactly which ones you want to upload:




Any files you selected specifically will appear here. If you picked groups of files (that is, directories, projects, and so on), that are managed by AccuRev, ClearCase, CVS, or Subversion these will be scanned for modified files, and those modified will be in the list. Check boxes allow you to pick more or fewer files than you originally selected.

The `Show Unchanged Files` option allows you to select files which have not been modified.

> The `Team | Add to Review` command uploads local copies of selected files. Once uploaded, they will be no longer synchronized with the SCM. To denote this, the Eclipse Plug-in displays an "Unsupported SCM System or Client[535]" warning when you try to upload the local copies of files.

To upload files to a review and keep them synchronized with the SCM:

- in CVS, add them from the Synchronize[553] view.
- in Perforce, add them from the P4 Pending Changelists[560] view or from the P4 Submitted Changelists[562] view.
- in Subversion, add them from the Synchronize[563] view or from the History[564] view.

Finishing the Add to Review Wizard[545] will upload your files to the Collaborator Server.

#### 5.6.1.12 AccuRev Integration

The Collaborator Eclipse Plug-in integrates with the AccuRev Plug-in for Eclipse from https://www.microfocus.com/documentation/accurev/.

## Adding AccuRev files to a Review

Files managed by AccuRev can be uploaded to the Collaborator Server by Adding Files to a Review 551.

#### 5.6.1.13 ClearCase Integration

The Collaborator Eclipse Plug-in integrates with the IBM Rational ClearCase SCM Adapter Plug-in from http://ibm.com, or the open source ClearCase plugin for Eclipse from http://sourceforge.net/projects/eclipse-ccase. Note - the Collaborator Eclipse Plug-in ClearCase integration requires that the ClearCase cleartool be installed.

## Adding ClearCase files to a Review

Files managed by ClearCase can be uploaded to the Collaborator Server by Adding Files to a Review 551.

**Supported Versions**

The Collaborator Eclipse Plug-in supports versions through ClearCase 8.x.

The Collaborator Eclipse Plug-in supports *only* ClearCase Remote Client (CCRC) v.8.x.

#### 5.6.1.14 CVS Integration

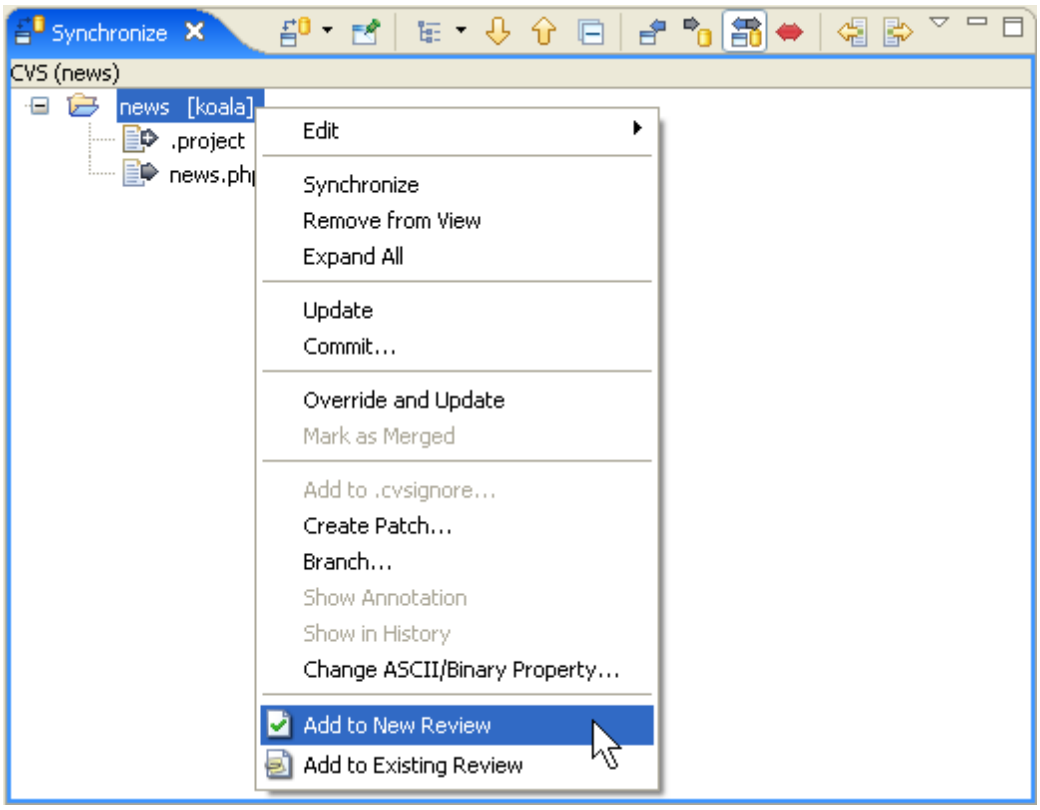
The Collaborator Eclipse Plug-in integrates with the CVS Eclipse Plug-in from http://eclipse.org. (The CVS plug-in usually comes bundled with the base Eclipse install.)

## Adding CVS files to a Review

Files managed by CVS can be uploaded to the Collaborator Server by Adding Files to a Review 551.

## Adding CVS files to a Review from the Synchronize View

Files managed by CVS can also be added to a review from the `Synchronize` view. Right-click on any projects, folders, or files in the `Synchronize` view and select either `Add to New Review` or `Add to Existing Review.`




The Add to Review Wizard[545] will launch with the selected files.

### 5.6.1.15 Git Integration

With the Collaborator Eclipse Plug-in, you can add project files, files from Git commits and repository branches to a review directly from the Eclipse IDE. This topic explains how you can do this.

## Requirements

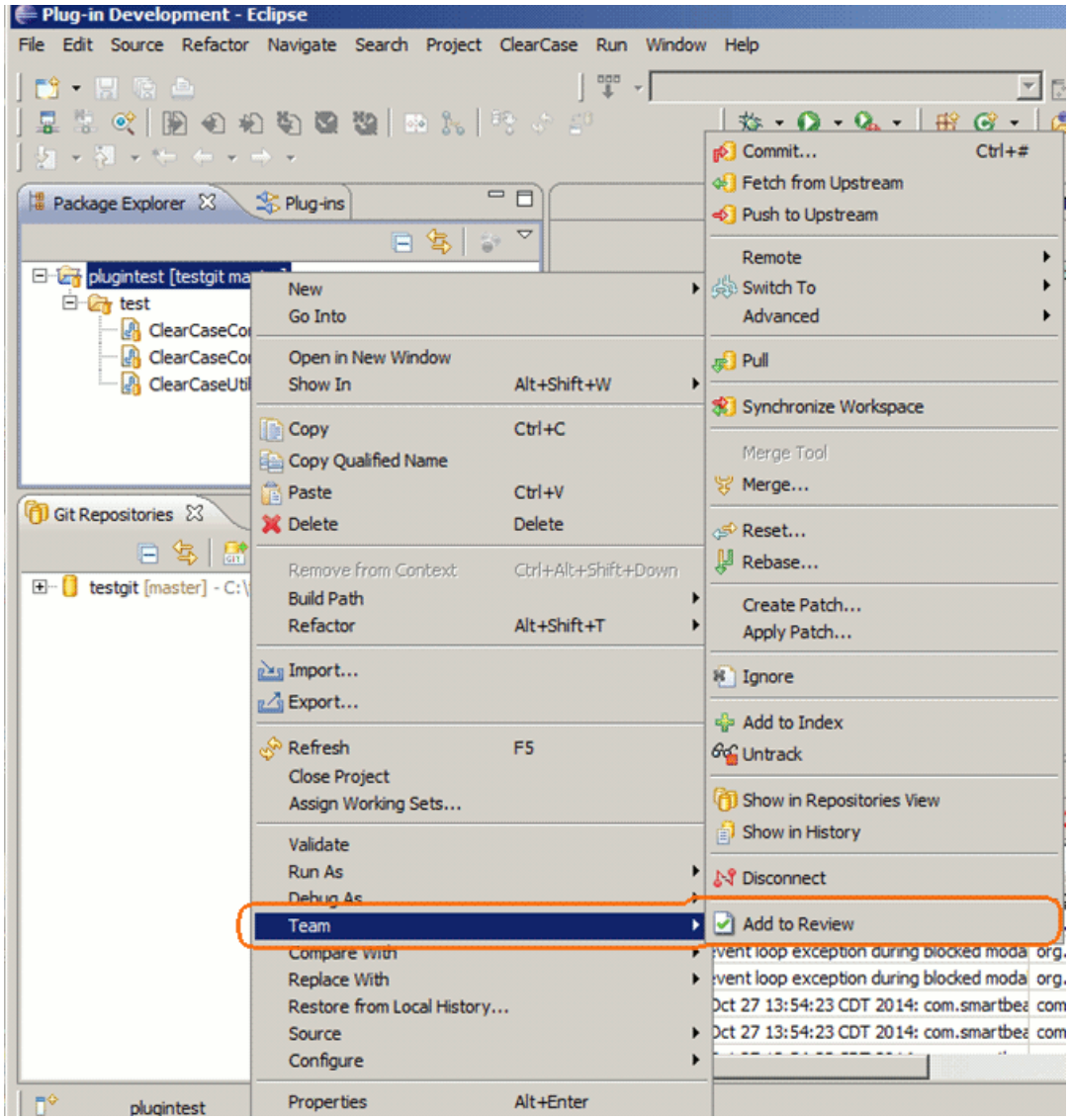
To integrate with Git and Eclipse, Collaborator's Eclipse Plug-in requires that the EGit plug-in to be installed in your Eclipse IDE. The EGit plug-in typically comes bundled with the Eclipse package. For more information on it, see http://wiki.eclipse.org/EGit/User_Guide.

## Add Git Files to a Review

To add your project's files to a review, follow these steps:

1. In Eclipse IDE, open the **Package Explorer** view.

2. Right-click your project or any of the project's files and select **Team | Add to Review** from the context menu:




This will invoke the Add to Review[545] wizard.

3. In the wizard, you specify the review name, and the project files and auxiliary documents to be attached to the review.

For detailed information on steps to be completed, see the wizard description[545].

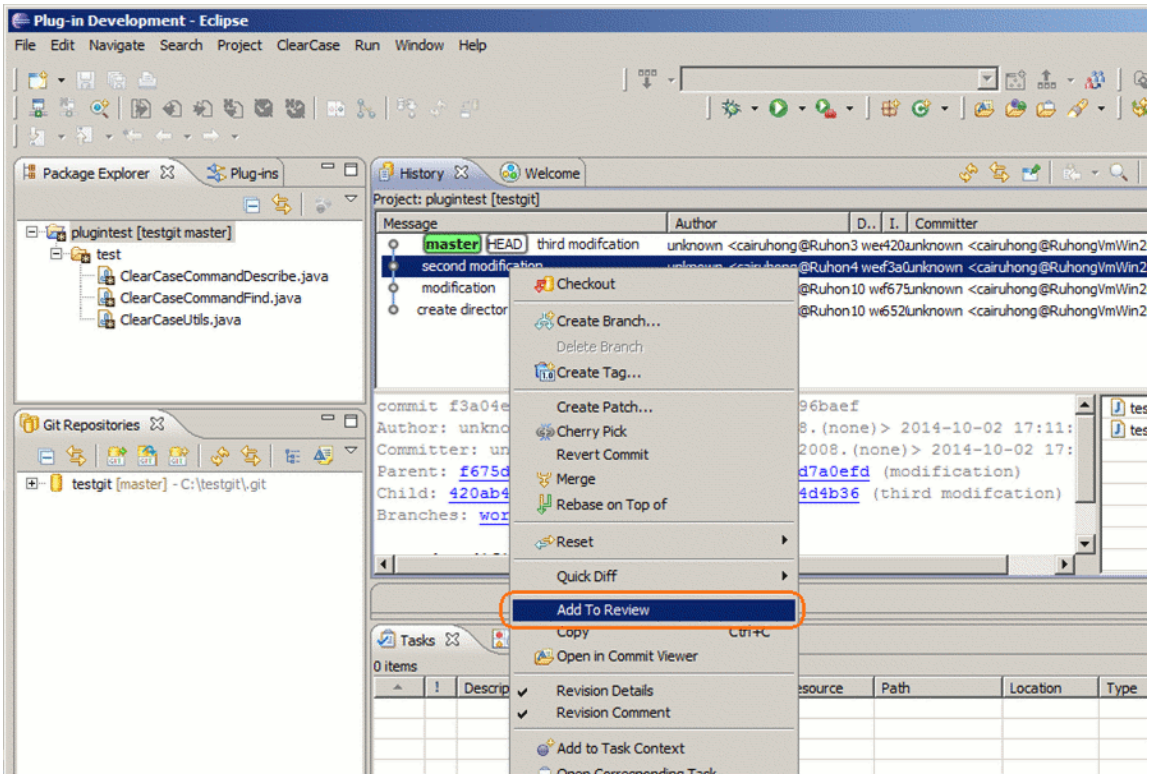
After you click Finish on the last page of the wizard, the wizard will send review data to the Collaborator server.

# Adding Git Commits to a Review

Adding a Git commit to a review means including copies of *all* the files of this commit to the review. Follow these steps:

1. In Eclipse IDE, open the **History** view. It lists Git commits.

2. In the History view, right-click a commit and choose **Add to Review** from the context menu:

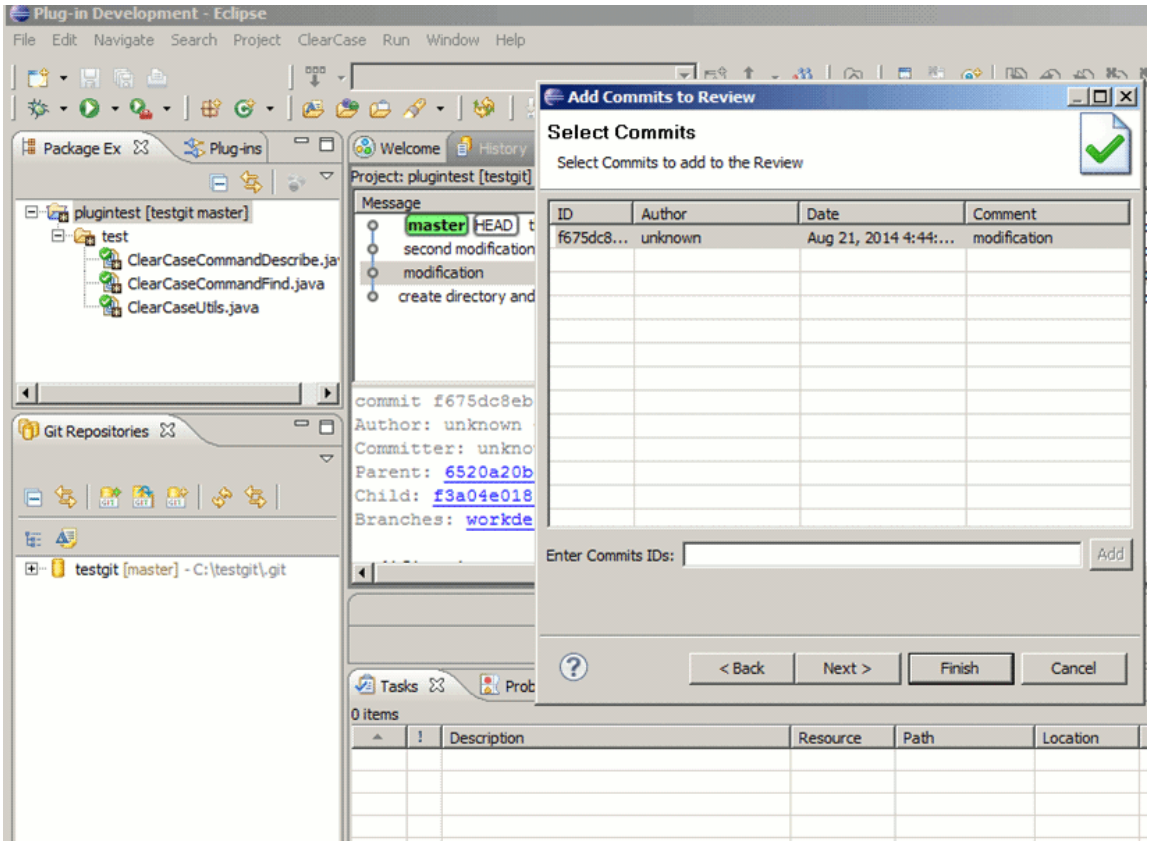



This will invoke the Add to Review 545 wizard.

**Known issue:** In the Indigo version of the Eclipse IDE, the "Add to Review" item is present in the context menu only when you invoke it for the first time. Later, it is hidden from the menu. To solve the problem re-open the History view.

3. On the first page of the Add to Review wizard, you can select the commits, whose files the review will include.

By default, this page contains only the commit that you selected in the History view. To add more commits to the list, type the commit identifier into the **Enter Commit IDs** text box and click **Add**.




Click **Next** to continue.

4. On the subsequent pages of the wizard, you can specify the review name and auxiliary files to be included into the review.

After you go through all the pages of the wizard, it will upload the review data to the Collaborator server.

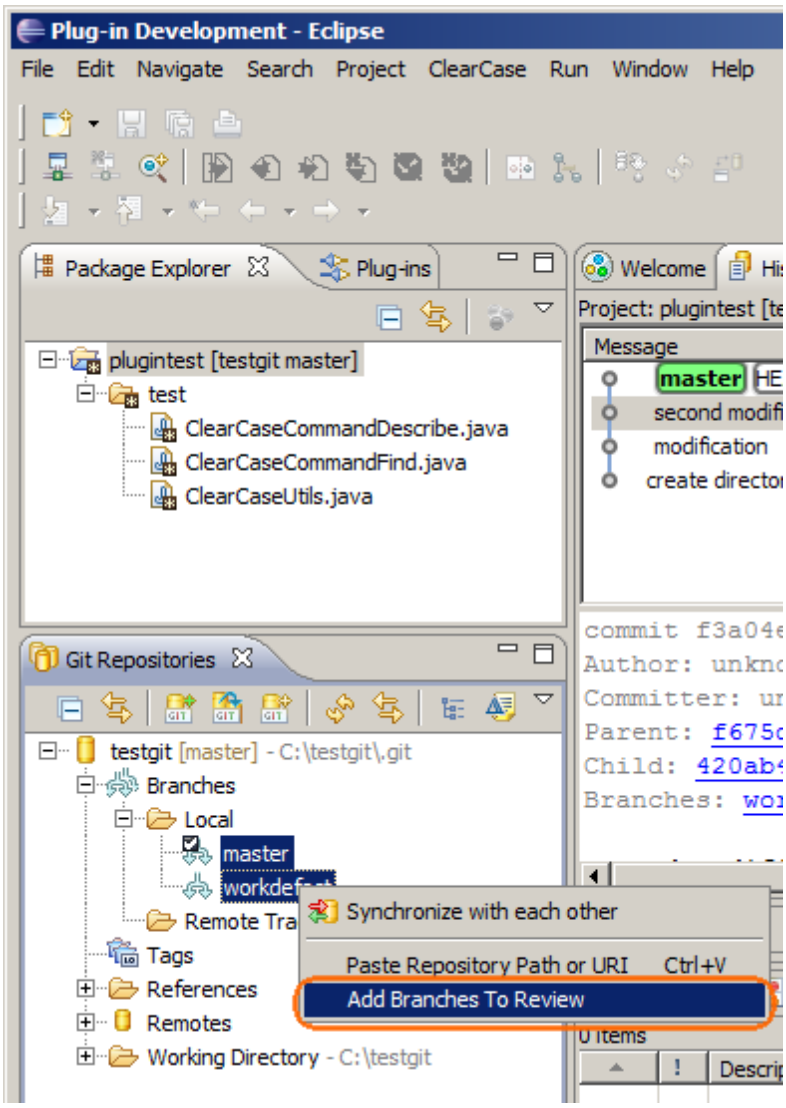
## Adding Files From Different Git Branches to a Review

The difference between two branches is in the committed files and the number of commits. Adding "differences" to a review means adding copies of *all* the files of *all* "different" commits to the review. Follow these steps:

1. In Eclipse IDE, switch to the **Git Repositories** view. It displays the contents of your Git repository.

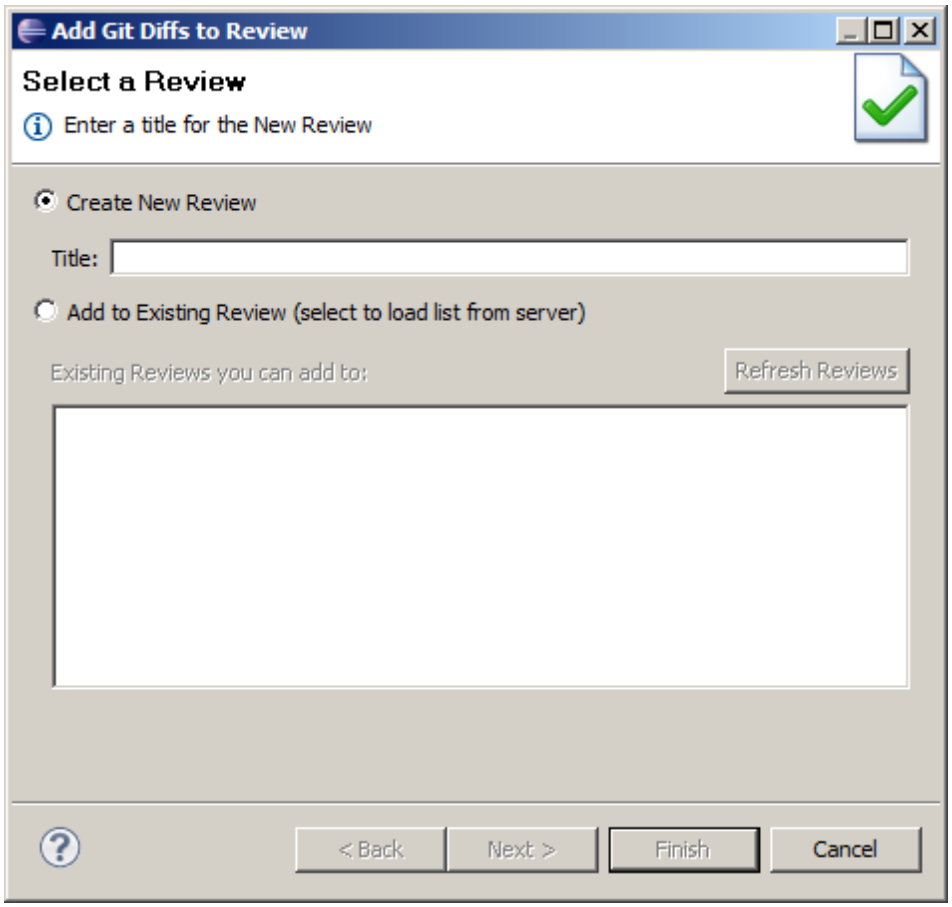
2. In the Get Repositories view, select the the desired branches.

3. Right-click the selection and choose **Add Branches to Review** from the context menu:

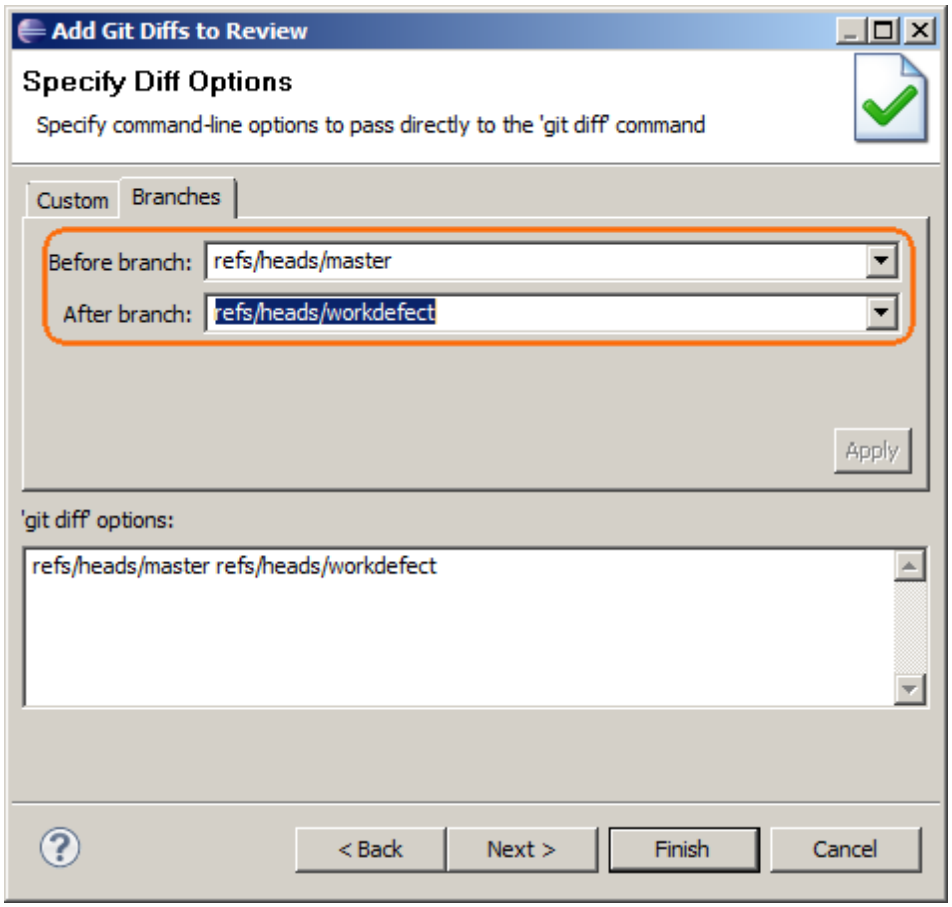



This will invoke the Add Git Diffs to Review wizard.

4. In the wizard, specify the review name --




-- and then select the branches to be passed to the `git diff` command (the Collaborator Eclipse Plug-in calls this command to determine the files to be added to the review):




The *before* and *after* are just Collaborator's terms to differentiate the branches. We suppose that one of these branches matches the original, or main, branch of your repository, and another branch contain the files that you modified while working on some feature.

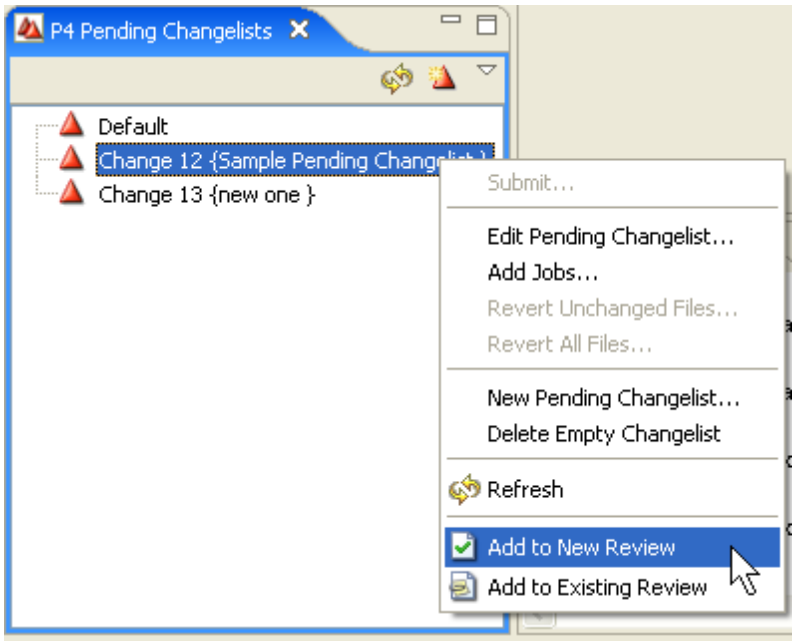
5. Go through other pages of the wizard to complete the review creation.

#### 5.6.1.16 Perforce Integration

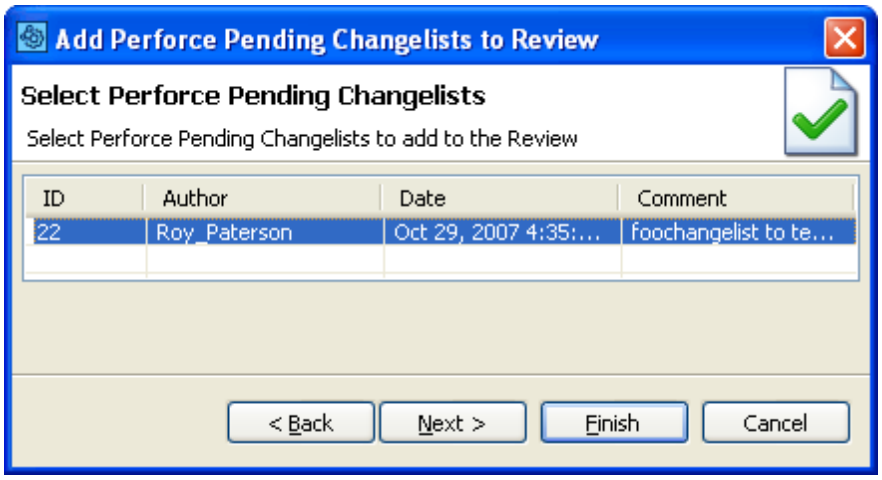
The Collaborator Eclipse Plug-in integrates with the Perforce using the Eclipse (P4Eclipse) Plug-in from https://www.perforce.com/plugins-integrations/eclipse-plugin.

## Adding a Pending Changelist to a Review

Right-click on one or more Pending Changelists in the `P4 Pending Changelists` view and select `Add to New Review` or `Add to Existing Review`.




Clicking one of the `Add to Review` items will launch the Add to Review Wizard[545]. The wizard will include a page that shows the Pending Changelists you selected, allowing you to select exactly which ones you want to upload:

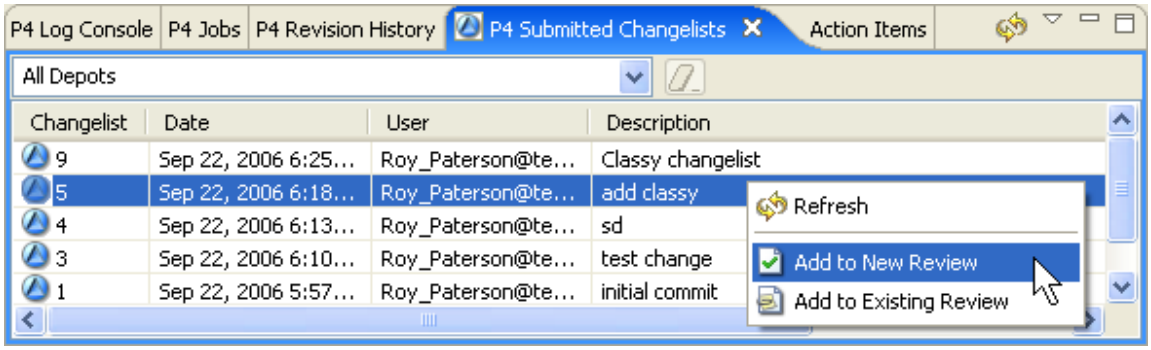



Any Pending Changelists you selected specifically will appear here. You may select more or fewer Pending Changelists than you originally selected.

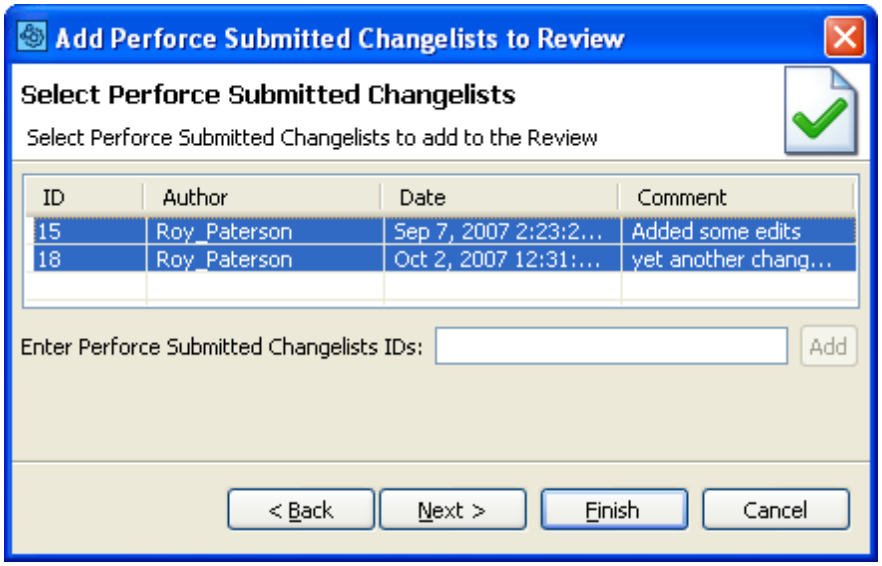
Finishing the Add to Review Wizard[545] will upload your Pending Changelists to the Collaborator Server.

## Adding Submitted Changelists to a Review

Right-click on one or more Submitted Changelists in the `P4 Submitted Changelists` view and select `Add to New Review` or `Add to Existing Review`.




Clicking one of the Add to Review items will launch the Add to Review Wizard[545]. The wizard will include a page that shows the Submitted Changelists you selected, allowing you to select exactly which ones you want to upload:




Any Submitted Changelists you selected specifically will appear here. You may select more or fewer Submitted Changelists than you originally selected.

Finishing the Add to Review Wizard[545] will upload the Submitted Changelists to the Collaborator Server.

#### 5.6.1.17 Subversion Integration

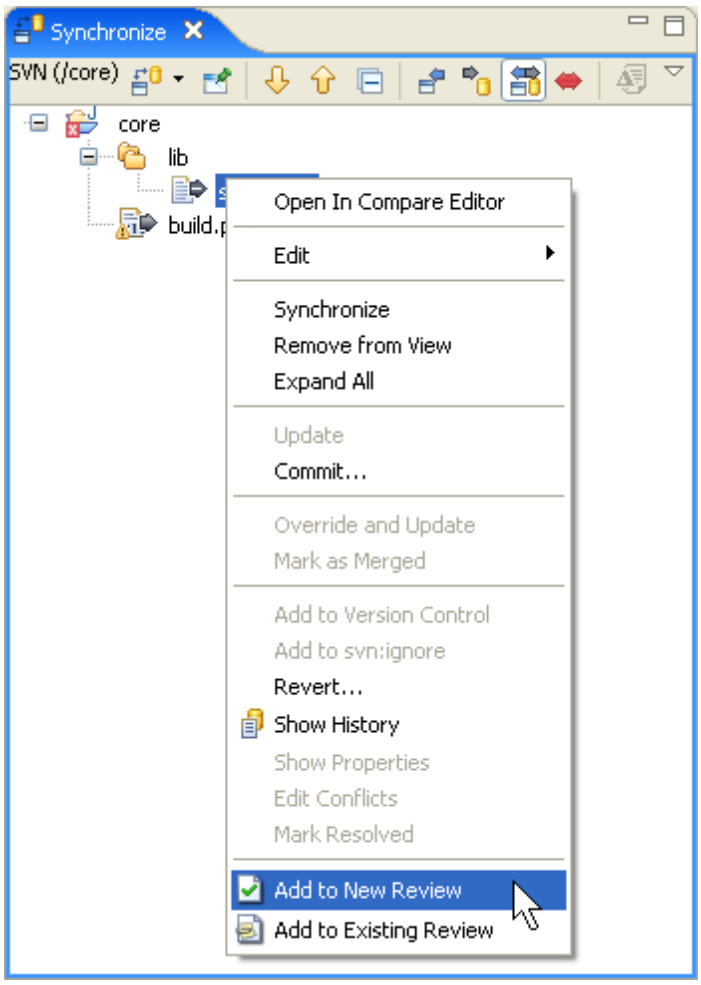
The Collaborator Eclipse Plug-in integrates with the Subclipse and Subversive Subversion Plug-ins. Currently, Collaborator Eclipse Plug-in supports Subclipse versions 1.4.x - 1.6.x and Subversive 4.x (SVN Connector 6.x).

## Adding Subversion files to a Review

Files managed by Subversion can be uploaded to the Collaborator Server by Adding Files to a Review 551.

## Adding Subversion files to a Review from the Synchronize View

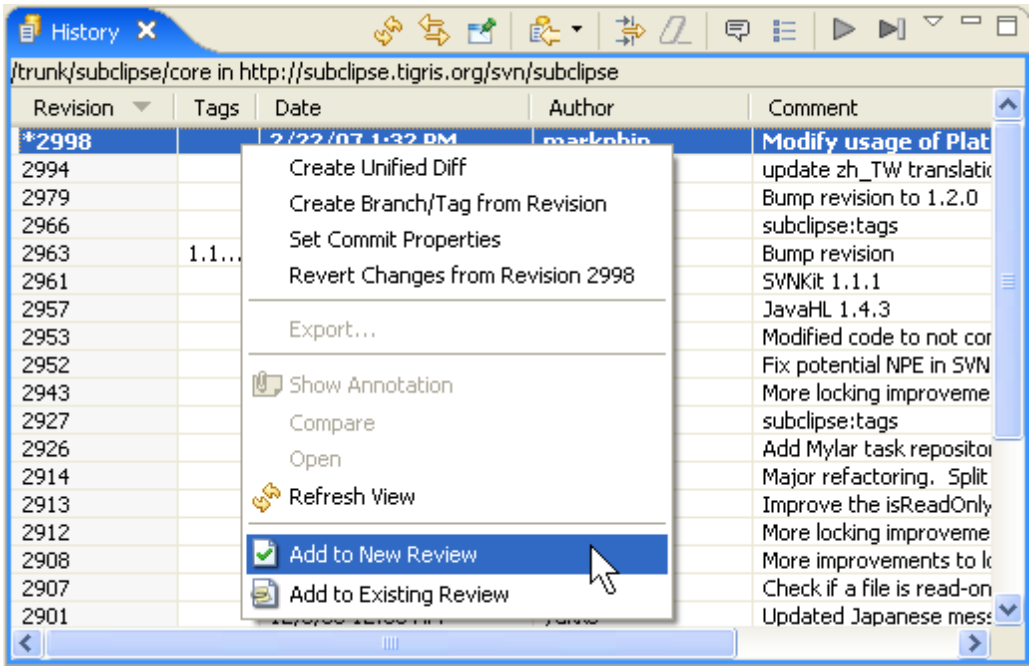
Files managed by Subversion can also be added to a review from the `Synchronize` view. Right-click on any projects, folders, or files in the `Synchronize` view and select `Add to Review.`



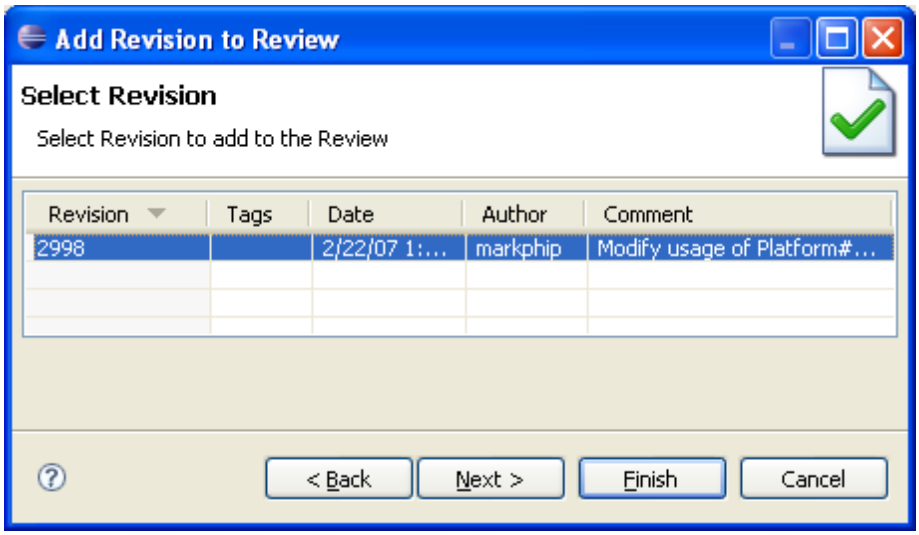

The Add to Review Wizard 545 will launch with the selected files.

## Adding a Subversion Revision to a Review

Revisions are shown in the `History` view. Right-click on a Revision and select `Add to Review`.




Clicking Add to Review will launch the Add to Review Wizard 545. The wizard will include a page that shows the Revision you selected:



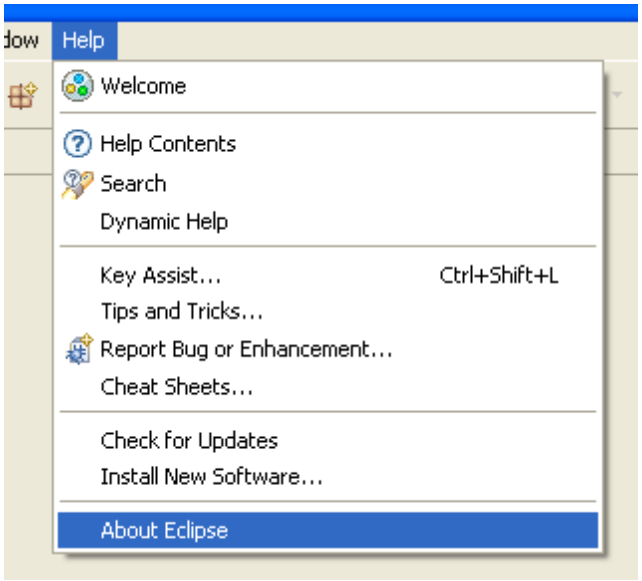

The Revision you selected specifically will appear here. You may select more or fewer Revisions than you originally selected.

Finishing the Add to Review Wizard[545] will upload the Revision to the Collaborator Server.

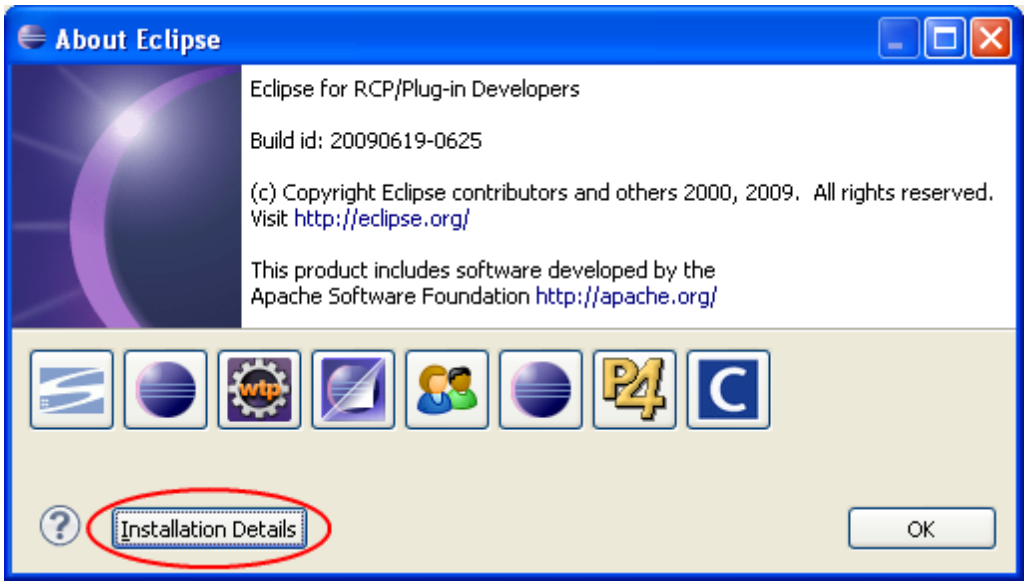
#### 5.6.1.18 Troubleshooting

If you run across any issues or problems, send the error log along with your question to Support[32].

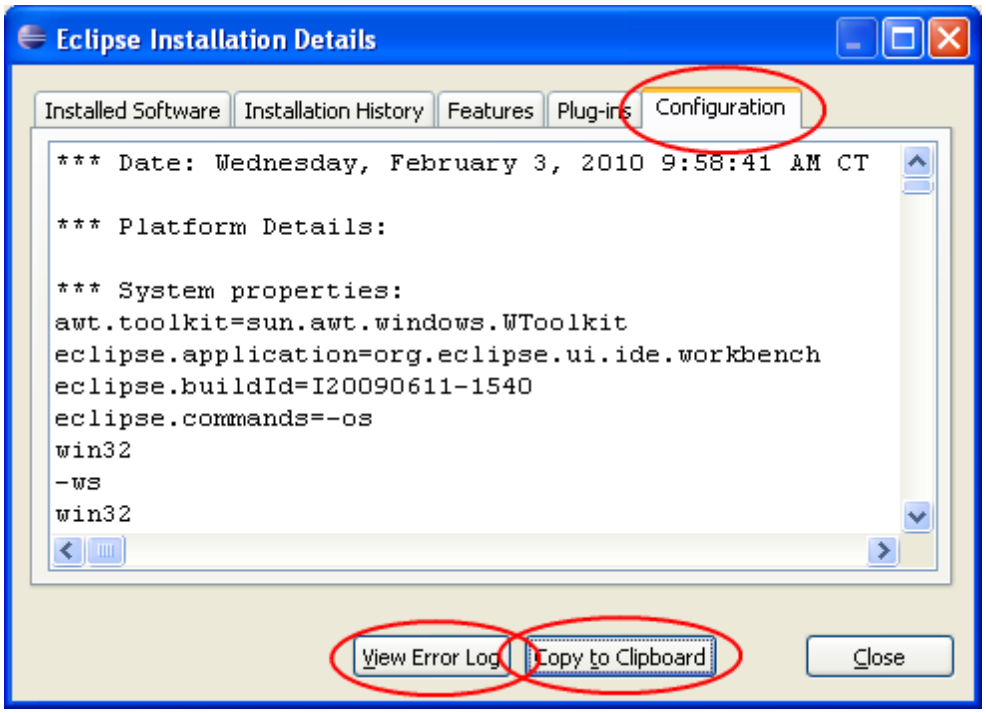
To get the error log, go to "Help" -> "About Eclipse".




Under the Help menu, click on "About"




Then, click on "Installation Details".




Click the "Configuration" tab. Click the "Copy to Clipboard" button and save it in a file. Then click "View Error log" and save that information in a file as well. Send both files to Support 32, along with a description of what you were trying to do when you encountered the error.

## 5.6.2 Visual Studio Extension

This chapter describes Collaborator extension for Microsoft Visual Studio, and includes the following sections:

- Overview 566
- Installing & Removing the Extension 568
- Configuring 569
- Collaborator View (Review Explorer) 578
- Creating Reviews 580
- Review Summary Screen 591
- Code Viewer and Diff Viewer 592

### 5.6.2.1 Overview

The SmartBear Collaborator extension for Microsoft Visual Studio provides integration between Microsoft Visual Studio and Collaborator. It allows you to create and participate in Collaborator

reviews directly from within the Visual Studio IDE.

## Get Started

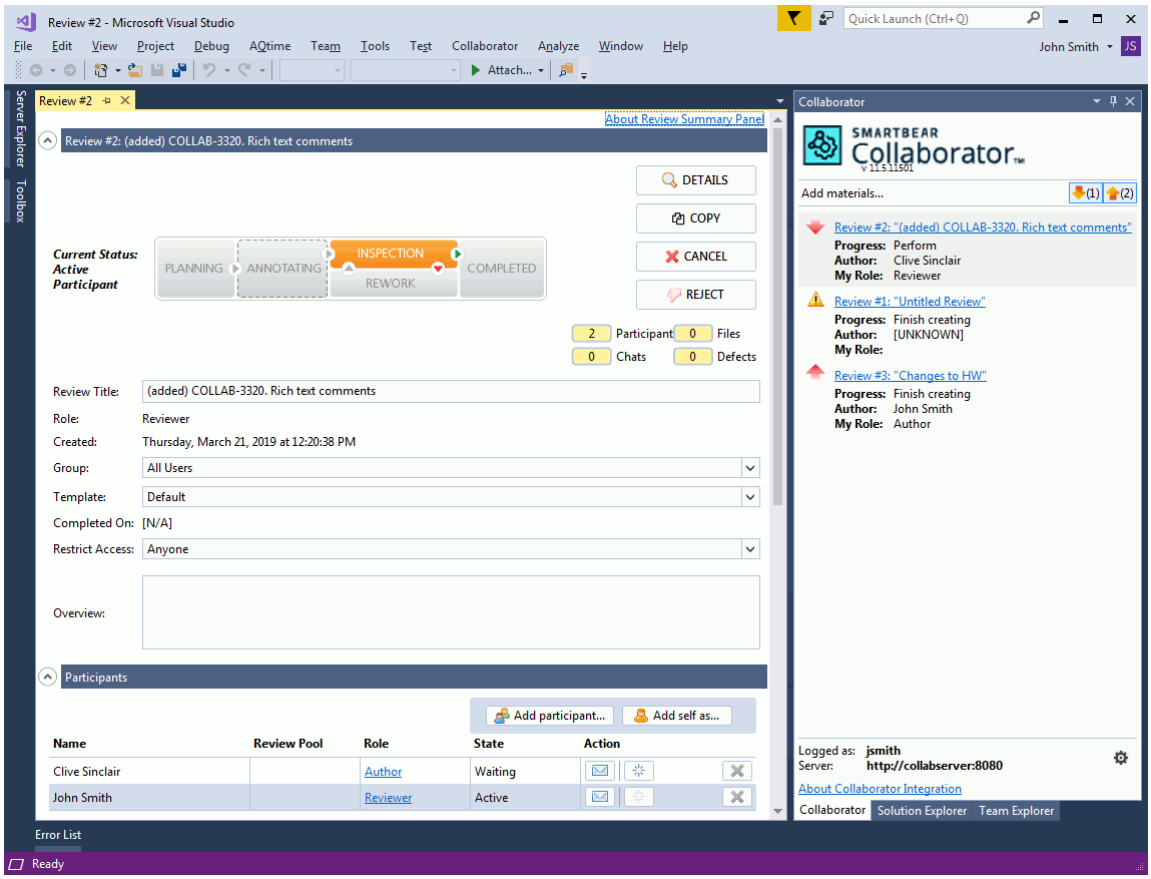
1. Download and install the extension[568] (the extension uses a stand-alone installer).

2. Select **Collaborator | Show Tool Window** from the main menu of Visual Studio to open the Collaborator View[578] panel.

3. If the extension is not configured[569] yet, it will invoke a Getting Started wizard suggesting to configure connection to Collaborator server and environment settings.

## How the Integration Works

The extension adds a number of panels, menus and settings to the Visual Studio. The "entry point" is the Collaborator[578] view:




In this panel, you can view the reviews and invoke other panels and dialogs to perform all types of peer review tasks:

- Get information about incoming and outgoing reviews[578] on the specified Collaborator server.
- Create reviews[580] as an author.
- Participate[578] in someone else's reviews as a reviewer or observer.
- Modify review details[591], add or remove review participants, annotate, reject or cancel your reviews, communicate with other participants.
- Inspect the review materials[592].
- Create comments and defects[593].

## Requirements

- Microsoft Visual Studio 2019, 2017 or 2015. The Community, Professional and Enterprise editions are supported.
- Collaborator Client on the computer, where you are going to installed the extension. The extension uses the Client to upload review materials to the server and to work with source controls.
- Collaborator Client must be properly configured to work with your source control system. See GUI Client: SCM Configuration[498] and Version Control Integration[620].
- Collaborator Server 12.0 and later.

## Known Issues

- Since the extension checks the Collaborator server periodically, it may take up to 2 minutes to retrieve information about newly created user from the server.

## Troubleshooting

If you run across any issues or problems, send the error log along with your question to Support[32].

The extension's error log resides at `%userprofile%\.smartbear\vsext\log\vsext.log`

#### 5.6.2.2 Install and Remove

## Installing the Extension

1. Download the Collaborator Visual Studio Extension from our website: http://support.smartbear.com/downloads/collaborator/

2. Unpack file from the downloaded .zip archive and run the CollaboratorVSExtension.vsix package.

3. In the VSIX Installer, choose the desired Visual Studio version and press **Install**.

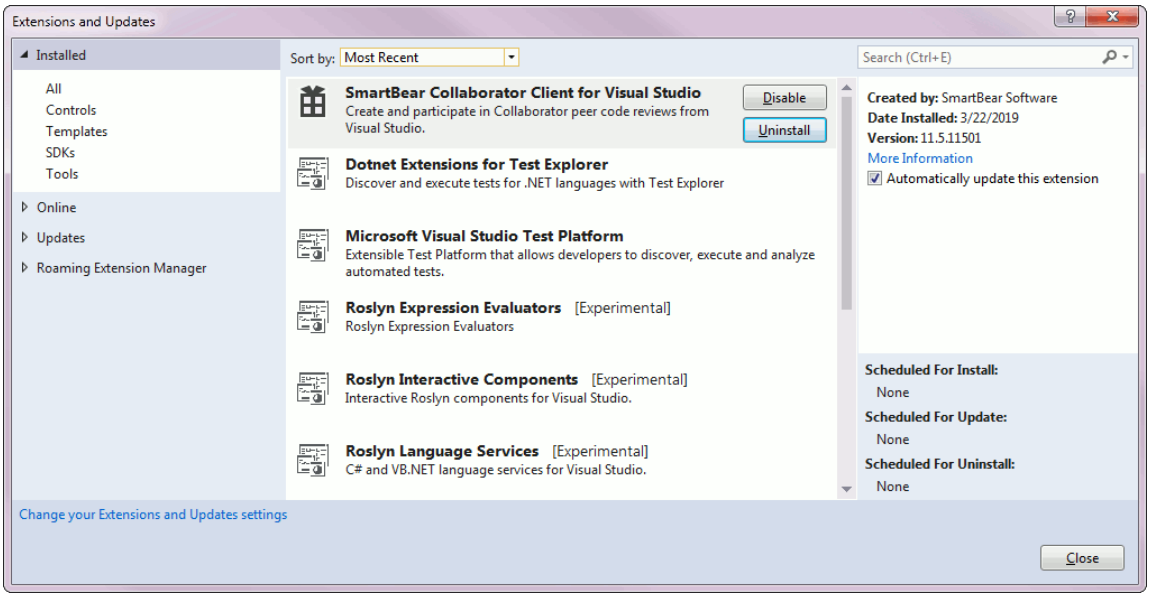
**IMPORTANT:** After installing the extension, you need to **configure its settings**[569] before the first use. Also, we recommend to configure SCM settings[498] in the Collaborator Client so that it can connect to your source control repository.

## Removing the Extension

1. Start Visual Studio.

2. Select **Tools > Extensions and Updates** from the main menu.

3. In the ensuing dialog, find the **SmartBear Collaborator for Visual Studio** extension and press **Uninstall**.

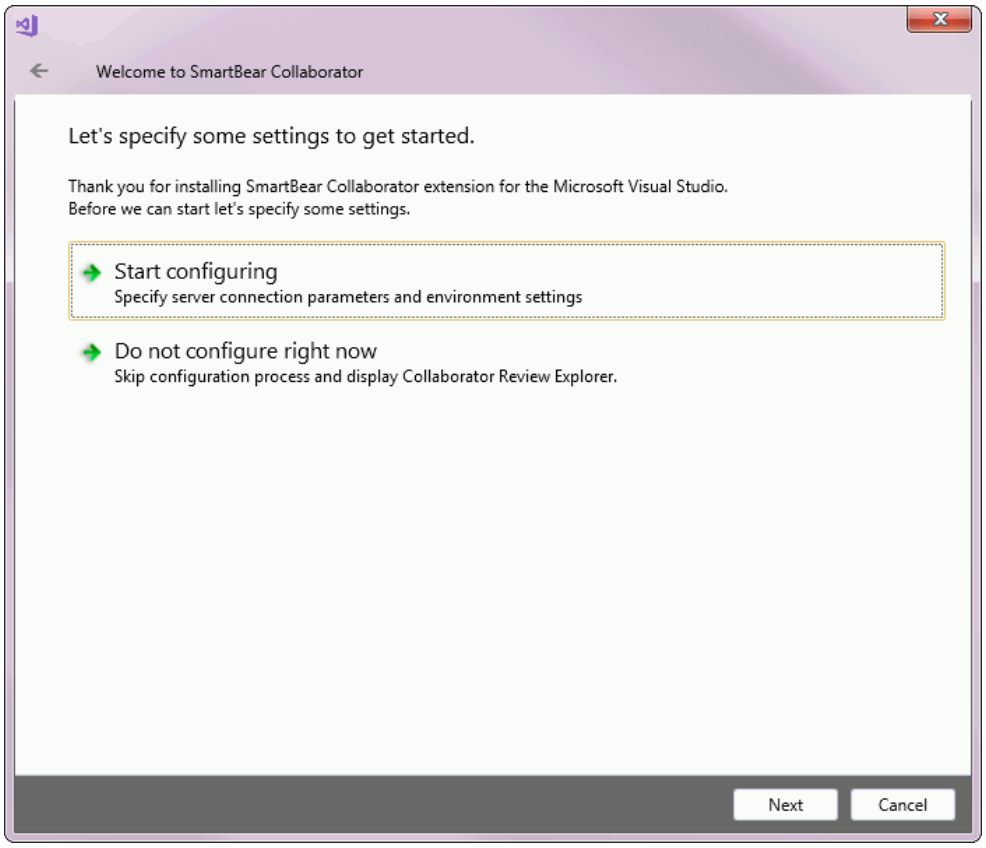



4. Restart Visual Studio to finalize the uninstallation.

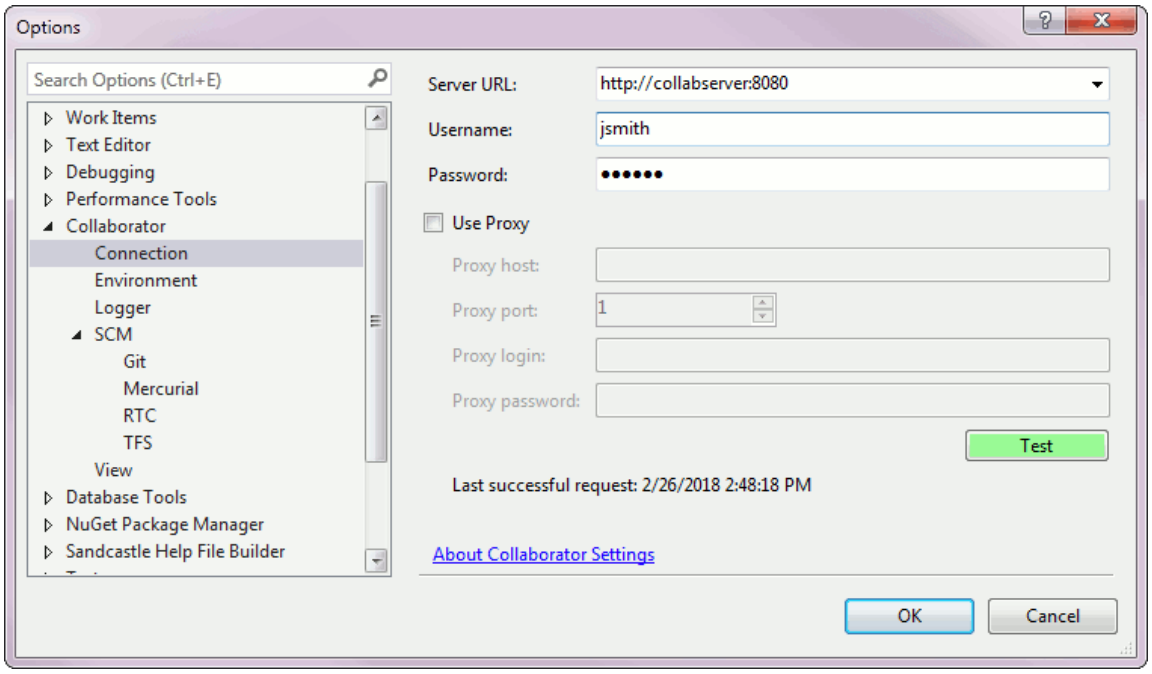
### 5.6.2.3 Configure

Before using the Collaborator extension for Visual Studio, you must configure it.

When you open the Collaborator View[578] window and the extension is not configured yet, it will invoke a Getting Started wizard suggesting to configure connection to Collaborator server and environment settings.




To change the extension settings afterwards (or if you have skipped the wizard), select **Tools | Options** from the main menu of Visual Studio and find **Collaborator** in the tree of the left:



The Collaborator extension includes the following settings:

## Connection (required)

**Server URL**

Denotes which Collaborator server to use. The Server URL must include the correct port number and path if applicable.

If your server uses HTTPS, you may need to install its certificate as described below 574.

**Username** and **Password**

Specifies user credentials. The Username and Password are the same as you use when logging into the web server 315. It is not recommended to use empty password.

You can specify your password or login ticket 318. When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead.

**Use Proxy**

Enable if you connect to the Internet via proxy server.

If you use proxy and want to connect to a Collaborator server that is deployed on your local machine, then specify **Server URL** in the "`Machine_Name:8080`" format rather than "`localhost:8080`" or "`127.0.0.1:8080`" formats.

**Proxy host** and **Proxy port**

The URL of a proxy server and its port number.

**Proxy login** and **Proxy password**

The login and password, if proxy server requires authentication.

Click **Test** to check if you can connect to the specified Collaborator server with the user name and password entered.

## Environment (required)

**Path to ccollab.exe**

The fully-qualified name of the Collaborator Command-Line Client executable.

| | |
|---|---|
| **Local Cache Folder** | The fully-qualified path to a temporary folder, where the extension will keep review materials obtained from the server. |

## Logger (optional)

This section controls what information will be displayed in the Visual Studio's Output window and in the debug log. You can toggle whether to display or not the information about exceptions, errors, warnings, debug messages or general information messages.

**SCM** (required)

Visual Studio extension share SCM configurations with GUI Client and command-line client. In this section you must select which of SCM configurations the extension will use to upload review materials. Optionally, in the subsections of this section, you may create new or modify existing SCM configurations.

| | |
|---|---|
| **Use Solution Path** | Specifies whether to treat path of current open solution as local SCM folder when uploading local changes. |
| **Configurations** | Lists the available configurations and displays their parameters.<br><br>To specify which of SCM configurations the extension will use to upload review materials, select a configuration in the drop-down list and press **Set as Default**. |

To create a new SCM configuration, specify its parameters and press **Add**.

To modify an existing SCM configuration, select it in the **Configuration** drop-down list, change parameters and press **Save**. For TFS configurations, you can also press **Test Connection** to verify the connection parameters.

To delete an existing SCM configuration, select it in the **Configuration** drop-down list and press **Remove**.

**Git**

| | |
|---|---|
| **Configuration** | Lists the available Git configurations and displays their parameters. |
| **Repository Path** | Specifies the path of local source code location. |

| | |
|---|---|
| **Git Exe** | Specifies the full path to the 'git' command line client. |
| **Mercurial** | |
| **Configuration** | Lists the available Mercurial configurations and displays their parameters. |
| **Repository Path** | Specifies the path of local source code location. |
| **Mercurial Exe** | Specifies the full path to the 'mercurial (hg)' command line client. |

**RTC** (To use RTC integration from Visual Studio Extension, you need to configure Collaborator server and client and RTC server 713)

| | |
|---|---|
| **Team Repository URI** | The URI of the repository to work with. |
| **User name** and **Password** | Specifies user credentials on the chosen team repository. |
| **TFS** | |
| **Configuration** | Lists the available TFS configurations and displays their parameters. |
| **Server URL** | For self-hosted version of Team Foundation Server, specify the URL of Team Foundation Project Collection to work with.<br><br>For SaaS version of Team Foundation Server, specify the URL of your Visual Studio Team Services account (without project or collection names). |
| **Local Path** | Specifies the path of local source code location. |

After specifying these values, you can click **Test Connection** verify if the TFS server is available. If you are connecting to this server for the first time, a Visual Studio sign-in prompt will be displayed.

## View (optional)

| | |
|---|---|
| **Group conversations by lines** | Controls how Code Viewer and Diff Viewer display comments and defects.<br><br>If enabled, the Conversations Pane 593 displays comments and defects that relate to a particular line. Otherwise, it displays the entire list of comments and defects. |

## Configuring HTTPS Connection

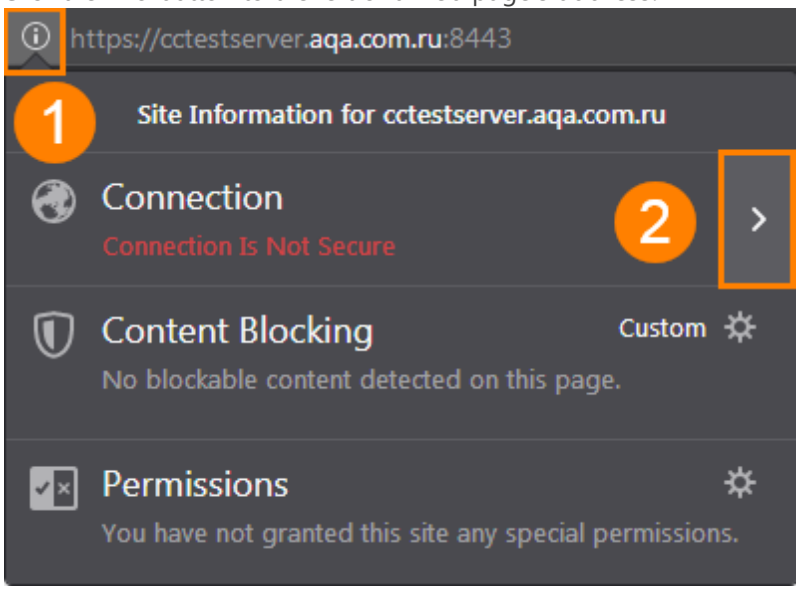
When your Collaborator server uses HTTPS connection and its certificate cannot be automatically verified, or server uses a self-signed certificate, then you will need to install the certificate manually.

**Note on self-signed certificates:** You have to use certificate signed with Certificate Authority (CA). It can be any CA - even yourself. And you must install that CA certificate in trusted authorities. You cannot use self-signed certificate directly.

### Obtaining Certificate

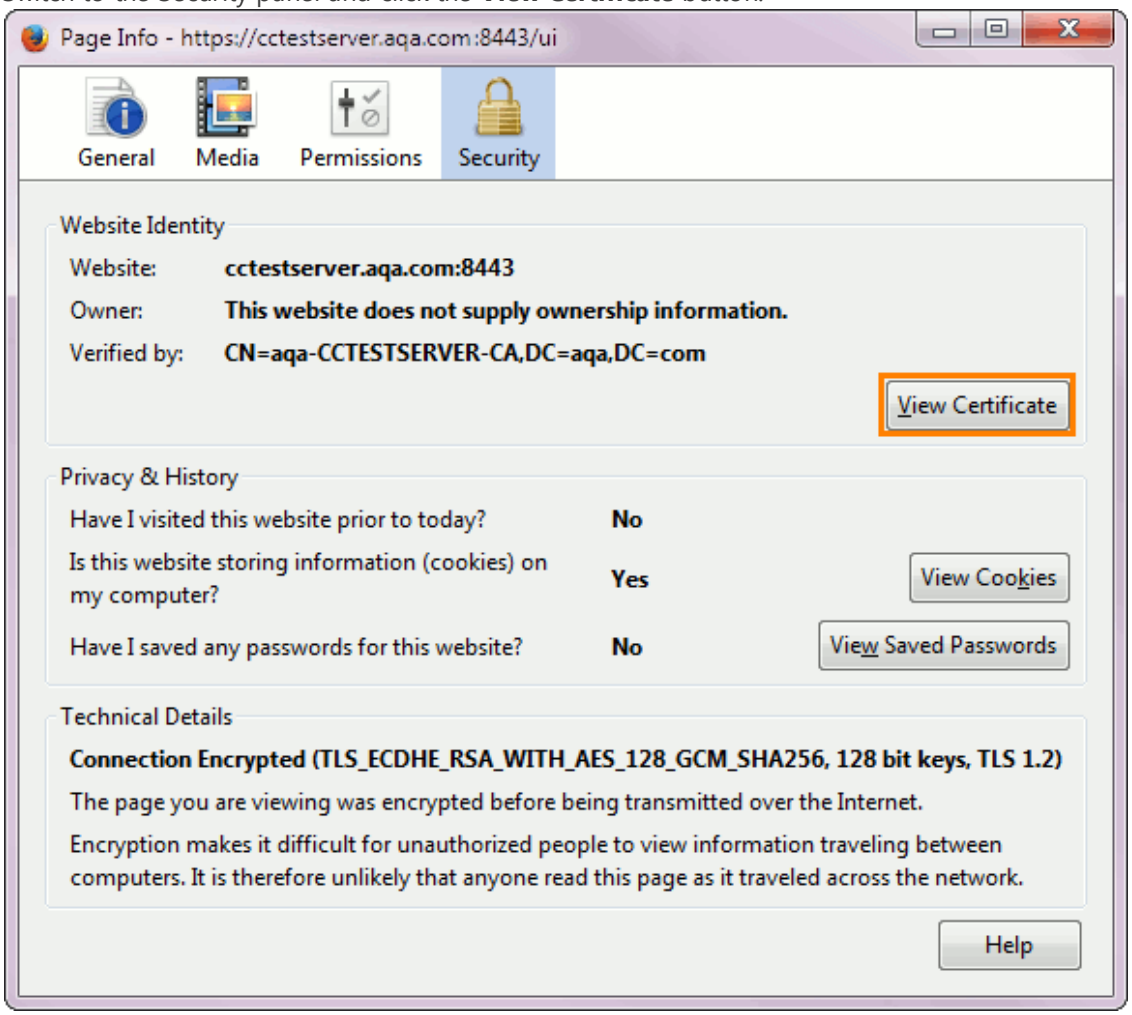
At first, you need to obtain the certificate from the Collaborator server administrator, or obtain it yourself as described below:

1. Launch Firefox browser and navigate to Web UI of your Collaborator server.
2. Click the info button to the left of a web page's address.



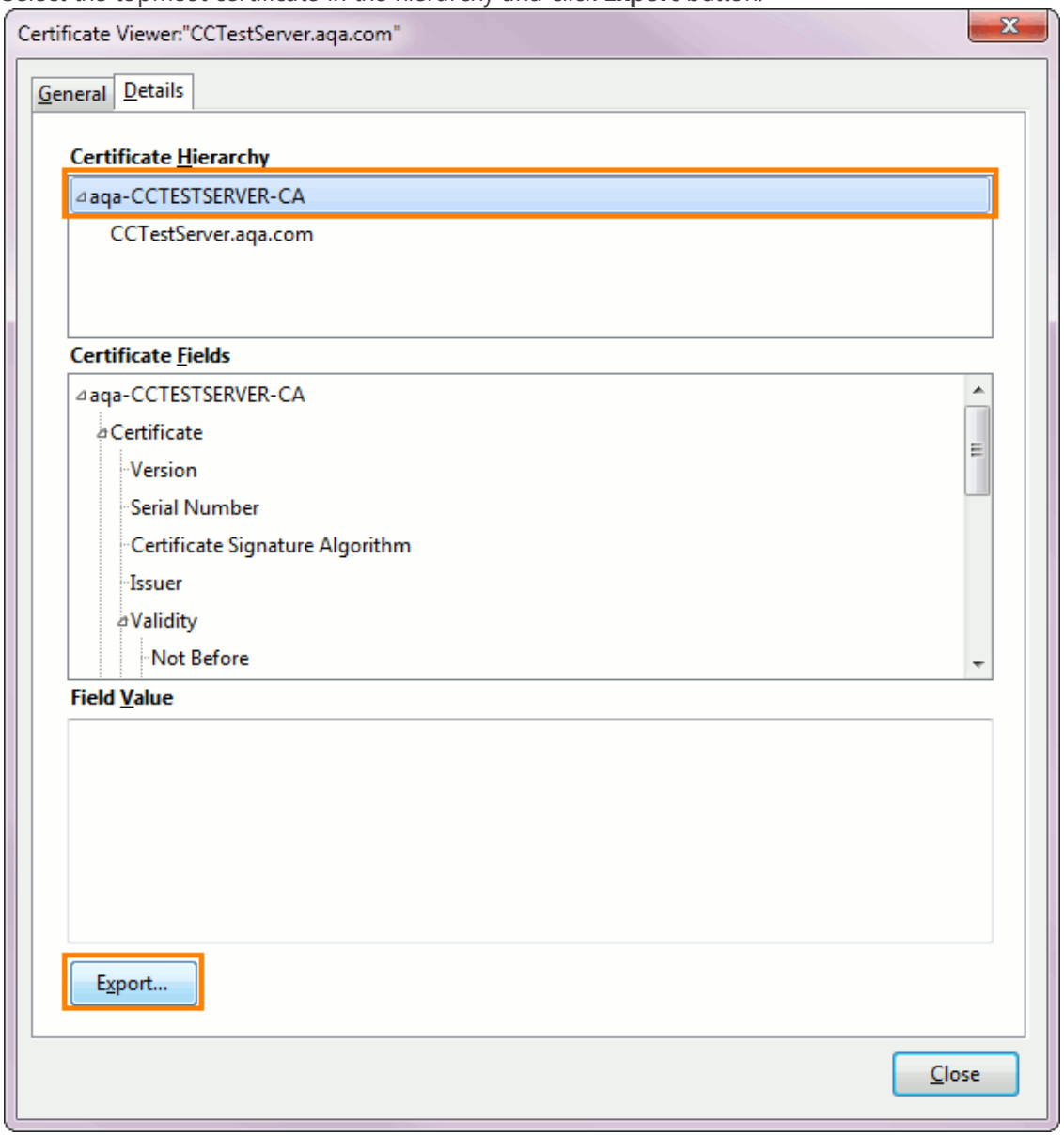

3. Click the **More Information** button in the next prompt. This will open the Page Info window.

4. Switch to the Security panel and click the **View Certificate** button.




5. In the ensuing Certificate Viewer dialog, switch to the Details tab.

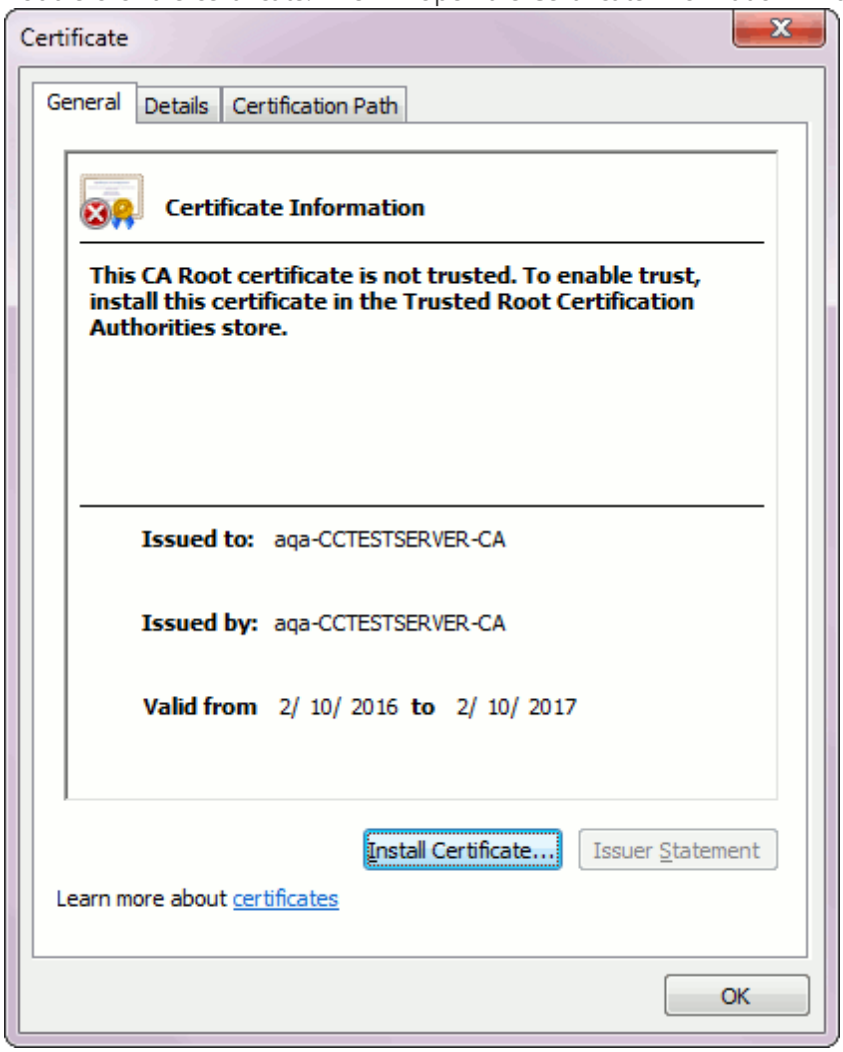
6. Select the topmost certificate in the hierarchy and click **Export** button.




7. Specify a path where to save the certificate in a standard Save File dialog.

## Installing Certificate

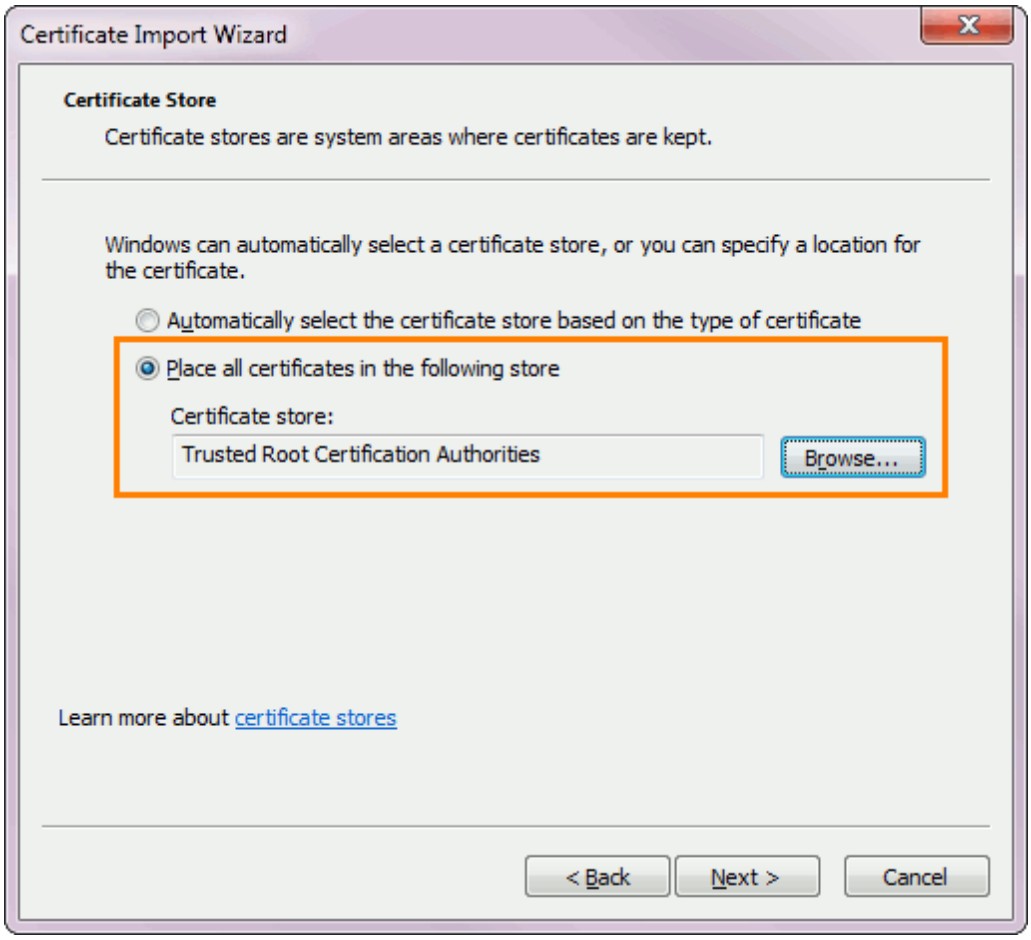
1. Double click the certificate. This will open the Certificate Information window.




2. Click **Install Certificate** button.

3. In the ensuing Import Certificate Wizard, click **Next** to proceed to the Certificate Store page.

4. Choose the "Place all certificates in the following store" option, click Browse and select the " Trusted Root Certificate Authorities" store.




5. Click **Next** and **Finish**.

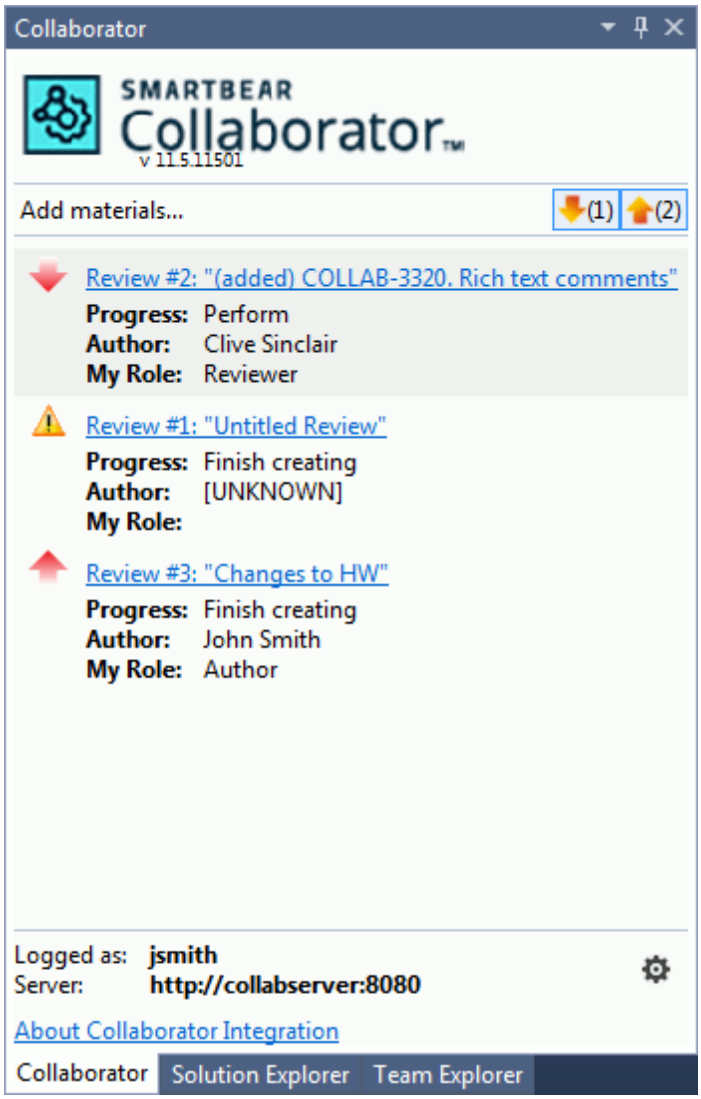
6. For self-signed certificates, a Security Warning will be displayed. Press **Yes** to confirm certificate installation.

Now Visual Studio Extension is properly configured for HTTPS connection.

### 5.6.2.4 Collaborator View

Collaborator Visual Studio extension adds a new window to Visual Studio IDE. The Collaborator View window shows current user name and server URL and lists the incoming and outgoing reviews on the specified Collaborator server. This window is analogous to the Home page with the Action Items 333 list of Web Client 313.

To display the window, select **Collaborator | Show Tool Window** from the main menu of Visual Studio.




From the Collaborator View window you can --

## Create new reviews or upload materials to existing reviews

Click "Add materials" from the Collaborator View's toolbar. For detailed information, see Creating Reviews 580.

## Open existing reviews

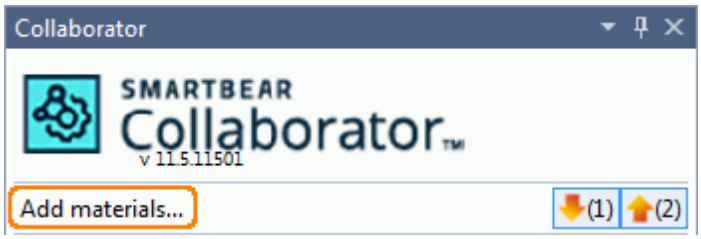
Click a review in the list and explore its details in the subsequent Review Summary Screen 591.

### Configure extension settings

Click the extension version number in the header, or the gear icon in footer. This will open the Options dialog with the extension settings 569.

#### 5.6.2.5 Creating Reviews

To create Collaborator from within Visual Studio IDE, use the **Collaborator | Add materials** main menu item, or the **Add materials** command in the toolbar of the Collaborator View 578 in Visual Studio:




## Requirements

Before using this command, we recommend that you configure SCM settings 569 to let it connect to your source control repository. See also Version Control Integration 620.

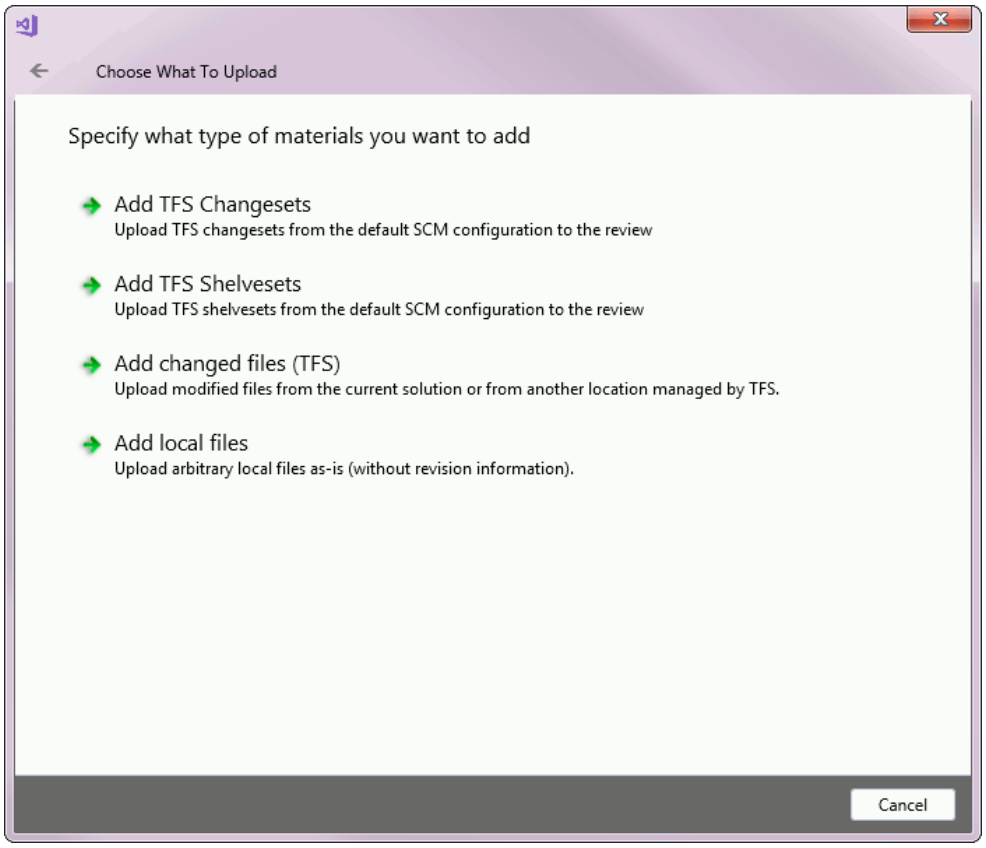
## How It Works

The command invokes the Add to Review Wizard that will lead you through the review creation process.

### 1. Select Review Materials

At first, you should select what type of materials to upload. The number of available options would depend on what SCM configuration you have chosen as the default in the extension settings.

- **Add changelists/changesets/shelvesets** - Upload changes that have been submitted to source control repository. The created review will contain all files that belong to the specified changelists, changesets or shelvesets along with the history of changes between revisions.
- **Add changed files** - Upload upload pending changes (staged for, but not yet submitted to source control repository) from the current solution or from another location. The created review will contain the selected files along with the history of changes between the staged and the submitted revisions.
- **Add local files** - Upload arbitrary local files. The created review will contain only the selected files without their revision history.

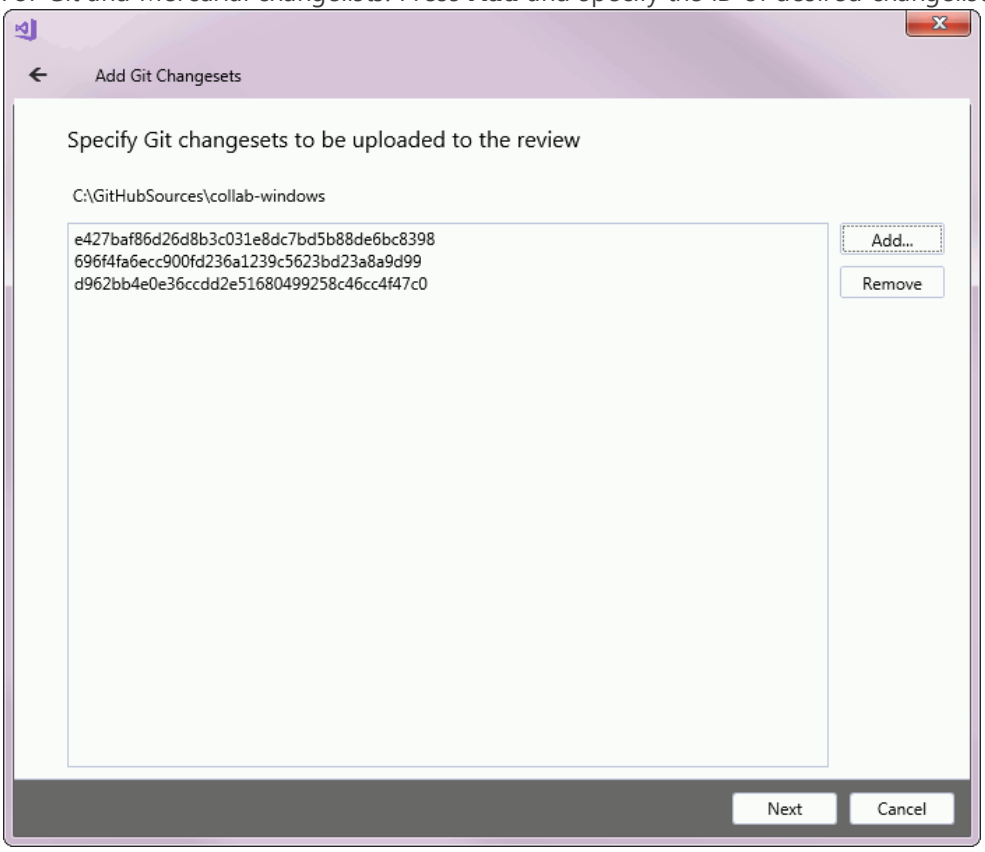



The next page lets you choose exactly which materials will be uploaded added to the review. This page looks different depending which materials you are adding to the review:

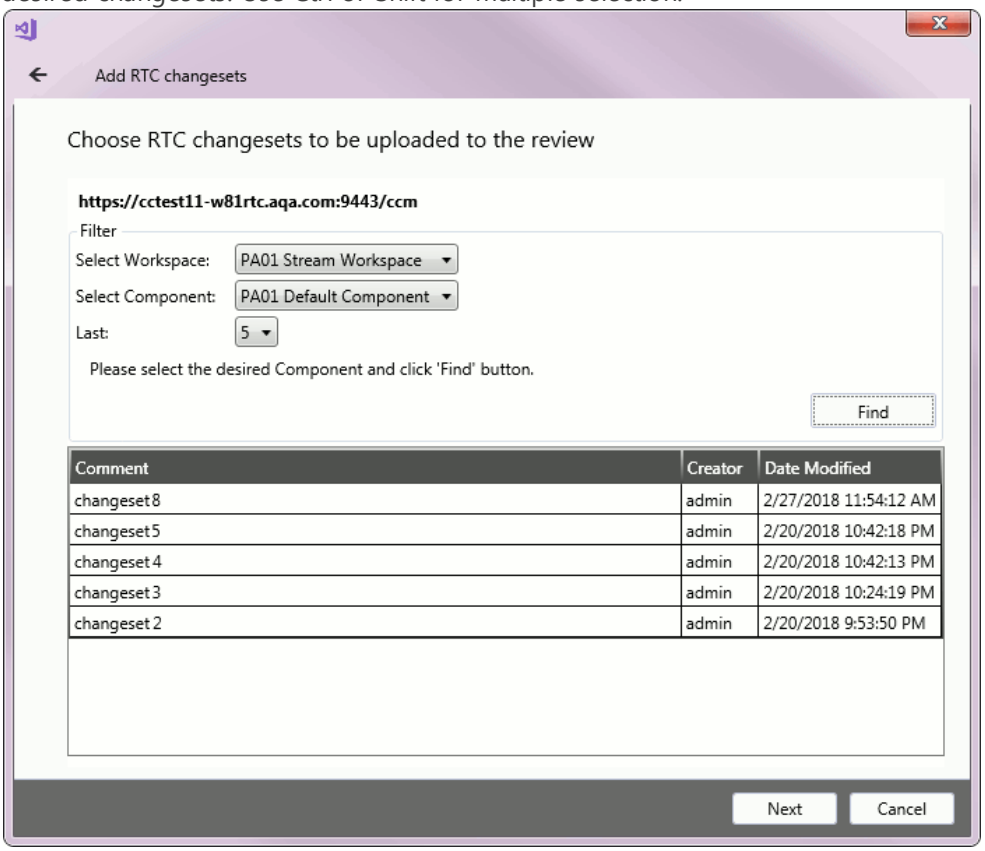
### 1.1 Uploading changelists/changesets/shelvesets

This page also depends on type of chosen SCM system and item.

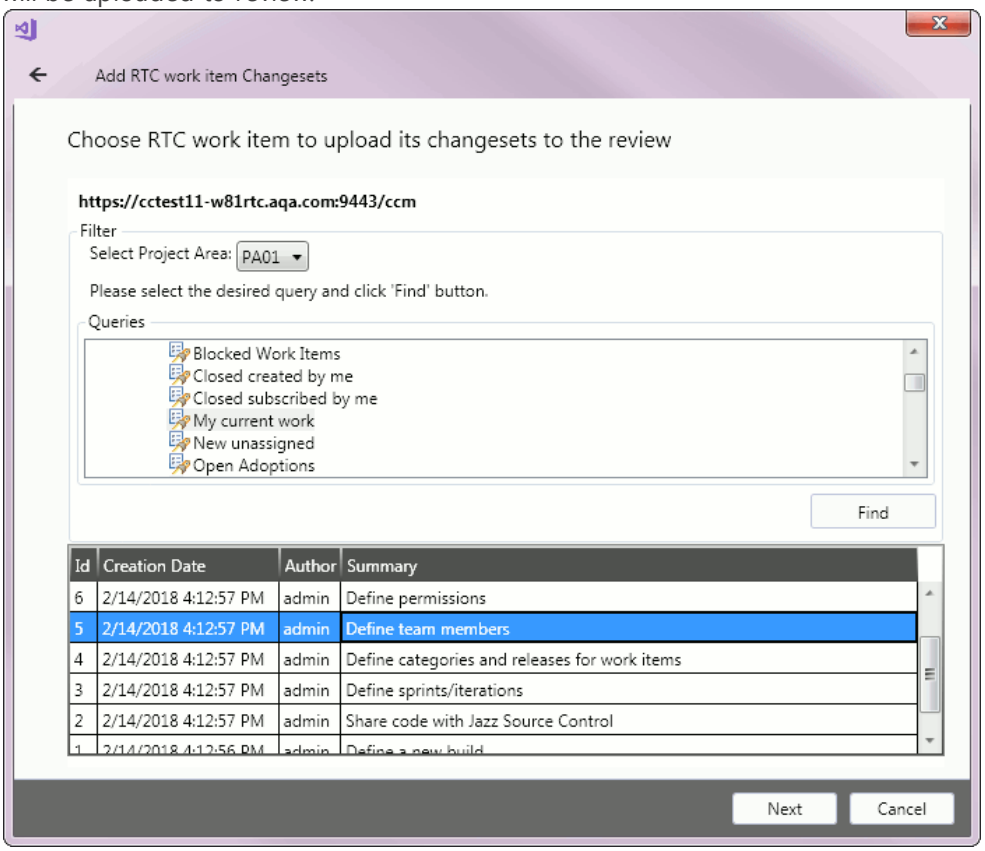
- For Git and Mercurial changelists: Press **Add** and specify the ID of desired changelist.



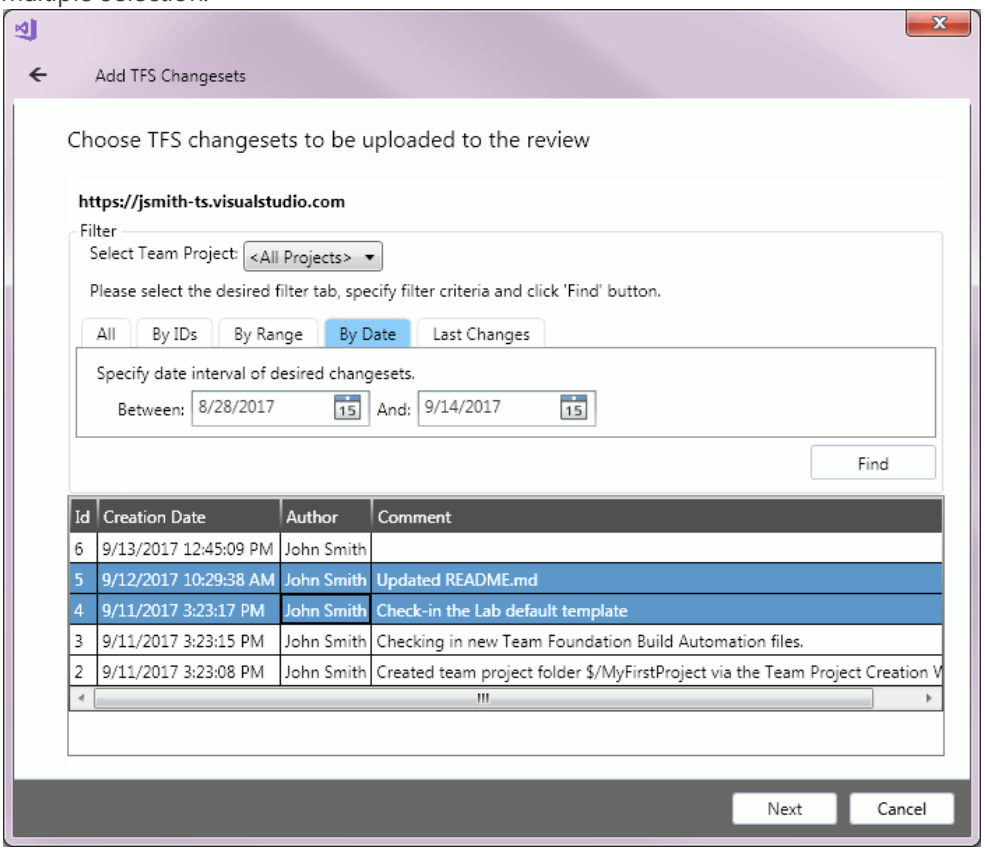
- For RTC changesets: Select desired workspace and component, click **Find** and then select the desired changesets. Use Ctrl or Shift for multiple selection.



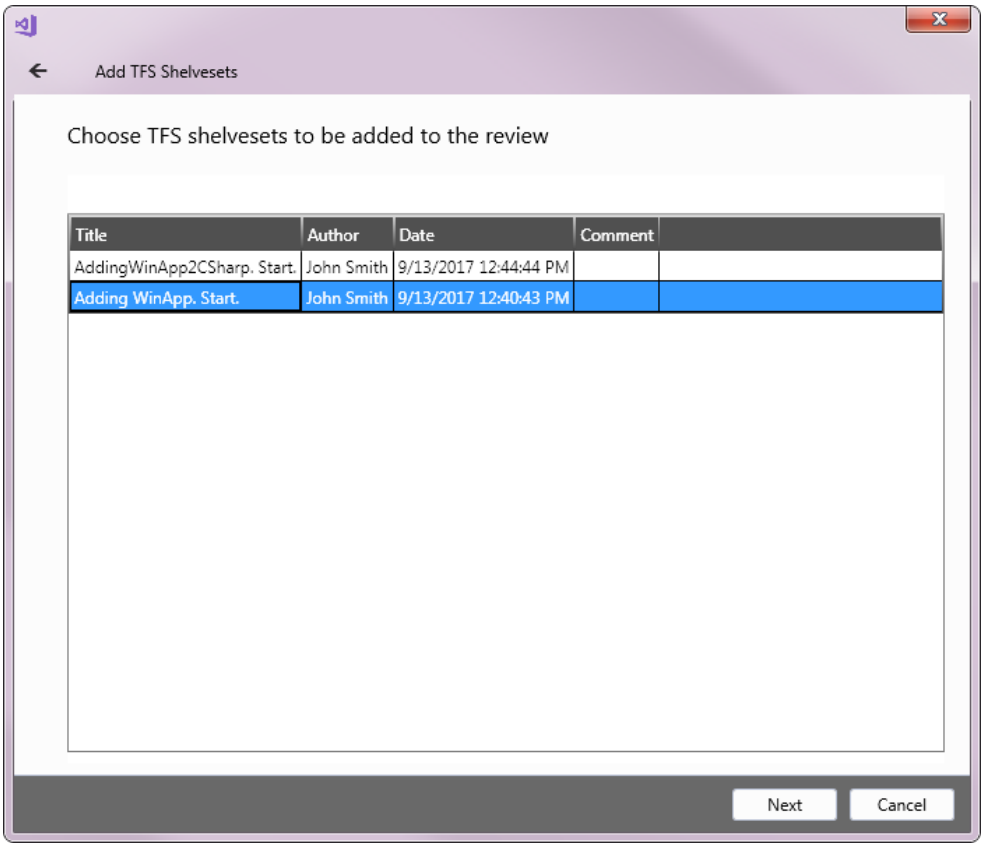
- For RTC work item changesets: Select project area and perform one of the listed queries, click **Find** and then select the desired work item. All changesets that belong to the work item will be uploaded to review.



- For TFS changesets: Select desired team project, specify changesets by their ID, range of IDs or date interval, click **Find** and then select the desired changesets. Use Ctrl or Shift for multiple selection.

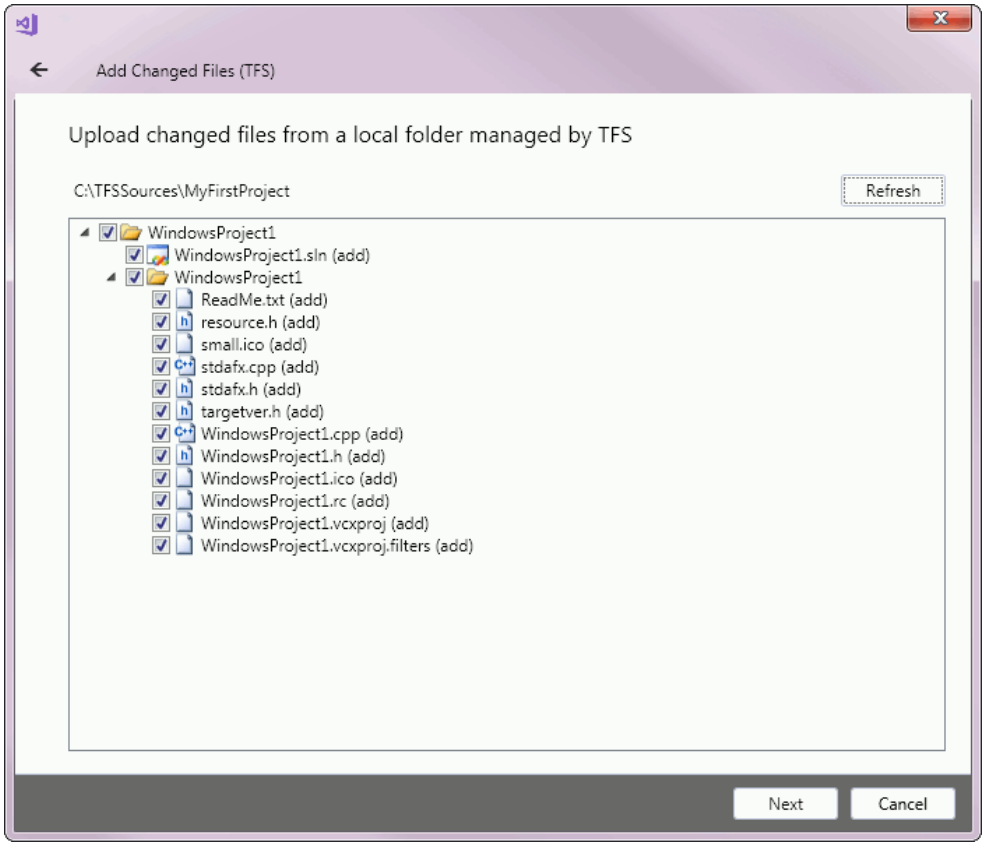


- For TFS shselvesets: Select the desired shelveset from the list. Use Ctrl or Shift for multiple selection.




When done, press **Next** to proceed.

## 1.2 Uploading changes

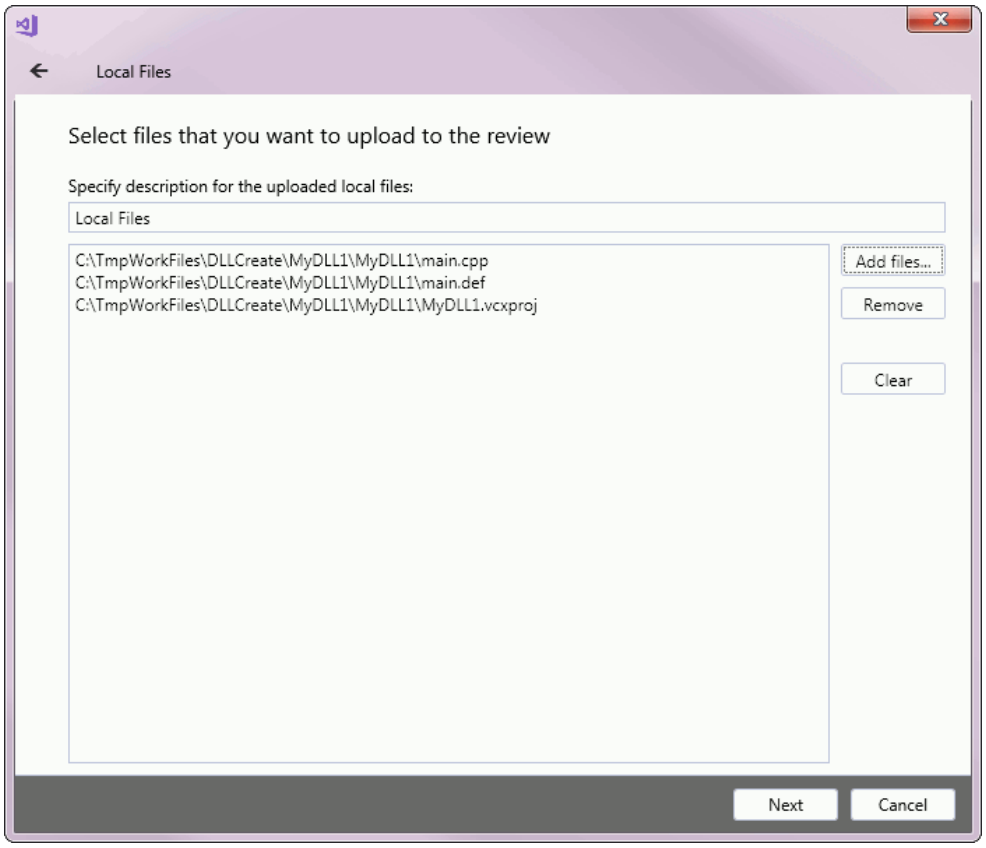



Select which of the modified files to upload.

When done, press **Next** to proceed.

## 1.3 Uploading local files

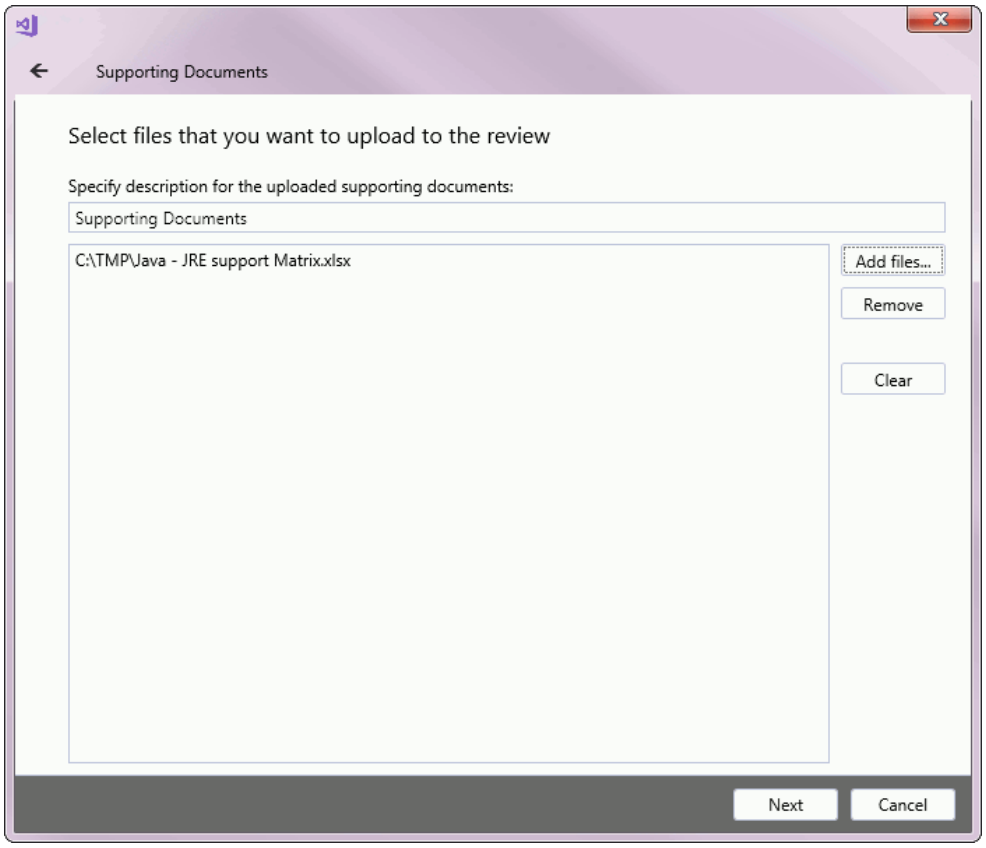
Select this command to upload arbitrary local files. This command does not check if these files are in a source control repository or not, and does not add information on differences to the review along with the files. Use it if you need to create a review for the files that are not in source control.




Click **Add files** to select the desired files in the ensuing Open File dialog, specify the commit message and press **Next**.

## 2. Select Supporting Documents

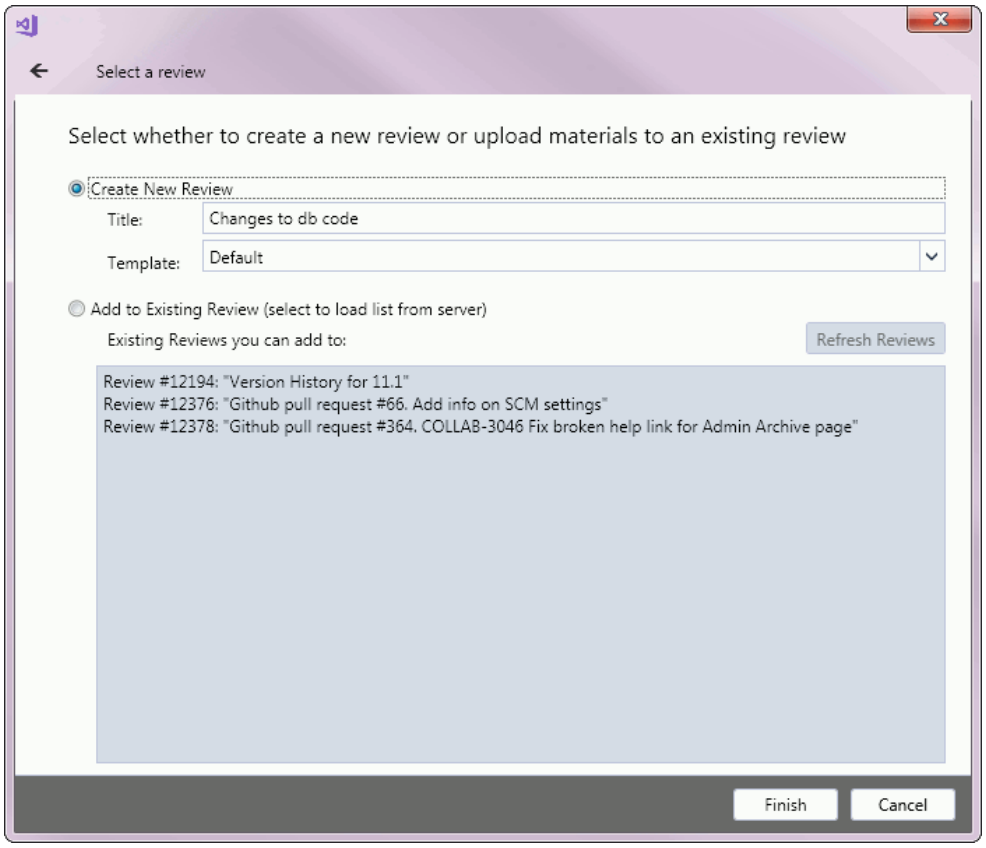
On the second page, you may upload any other documents that supplement the uploaded files.




Click **Add files** to select the desired files in the ensuing Open File dialog, specify the commit message and press **Next**.

## 3. Select Review

The third page of the Add to Review Wizard asks you to specify which review you would like to add materials to. You can create a new review and add the materials into the newly created review, or you can add the review materials to an existing review.




To add materials to a new review, select **Create New Review**, type a title to name your new review and select the review template.

To add materials to an existing review, select **Add to Existing Review**. Choose the review you would like to add the materials to. The list of possible reviews is created automatically by querying the server. This will generally be the same as the list of reviews in the Collaborator View.

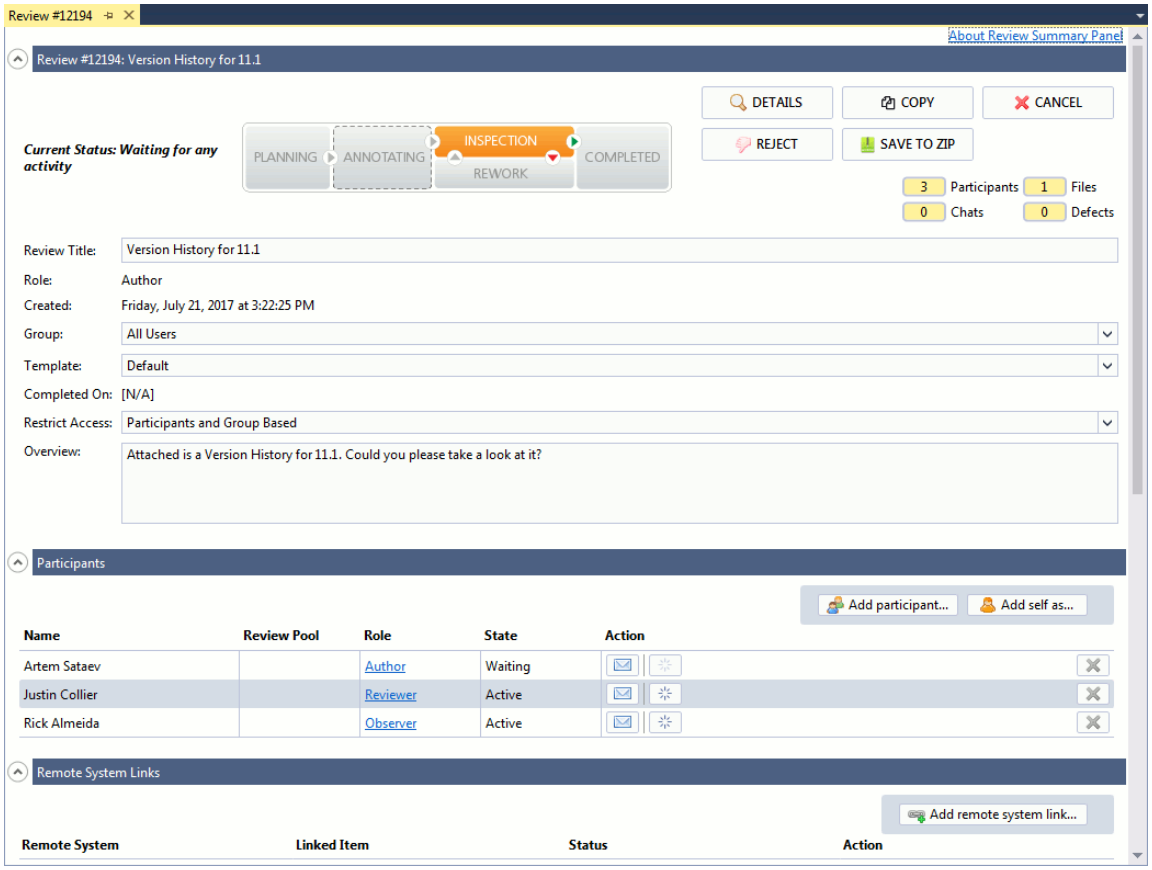
Once you click **Finish** all chosen review materials will be uploaded to the Collaborator server.

**Note:** Uploading files may take some time. Please be patient. During the upload, a progress dialog will appear.

When the upload is complete, the review will be displayed in the Review Summary Screen 591.

### 5.6.2.6 Review Summary Screen

When you click some review in the Collaborator View[578] it opens in the Review Summary screen. This window is analogous to the Review Summary Screen[345] of Web Client.




You will use the Review Summary Screen during the entire process of code review: creating new reviews, participating in other reviews, correcting found defects.

This screen allows you to --

- Modify general information about the review.
- Reject, cancel, copy, archive your reviews.
- Add or remove participants.
- Send e-mail and poke notifications to participants.
- Check linked remote system items and add new remote system links.
- View defect log.
- Communicate with other participants.

- View and upload materials for the review.
- Start, annotate, approve the review or wait until certain specified activity occur.

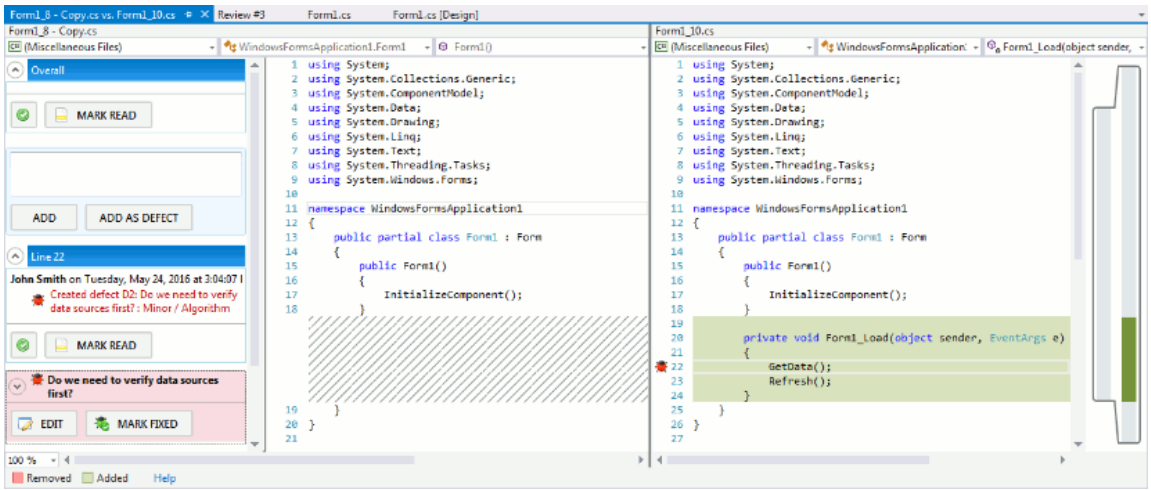
### 5.6.2.7 Code Viewer and Diff Viewer

Clicking on a file in the Review Summary Screen 591 opens it in a Code Viewer or in a Diff Viewer.

**Important:** Collaborator Visual Studio extension is designed to perform code reviews, rather than document reviews. That is, from the Visual Studio IDE you can review and compare source code and text files, but may fail to review and compare other types of review materials. To review Word documents, Excel tables, PDF files, images, or URLs you should use the Collaborator Web Client 312.

If a review has only one revision of the desired file, it will be opened in a Code Viewer (standard code editor of Visual Studio in read-only mode).

If a review has several revisions of the desired file, it will be opened in a Diff Viewer (standard Diff Window of Visual Studio).




*Two revisions of a file in a Diff Viewer*

In the Diff Window, the most recent revision is displayed on the right pane and the previous revision is displayed in the left pane. New text is highlighted in green and removed text is highlighted in red.

The comment icon ( ) in the viewer's gutter indicates that a line has some comments related to it. The defect icon ( ) indicates that a line has some defects related to it. Click the icon to navigate to the corresponding comment or defect.

## Conversation Pane

On the left side of a Code Viewer and Diff Viewer, a Conversation Pane is displayed. It lists the comments and defects that were created for the file being displayed. You can read the conversation, reply to and/or argue with other participants, mark the defects as fixed and so on.

Comments and defects may relate to the entire file (these are called **overall comments and defects**) as well to a particular line of code (these are called **line comments and defects**).

To create an overall comment:

1. Specify your message in the Overall section of the Conversation pane
2. Press **Add**.

To create an overall defect:

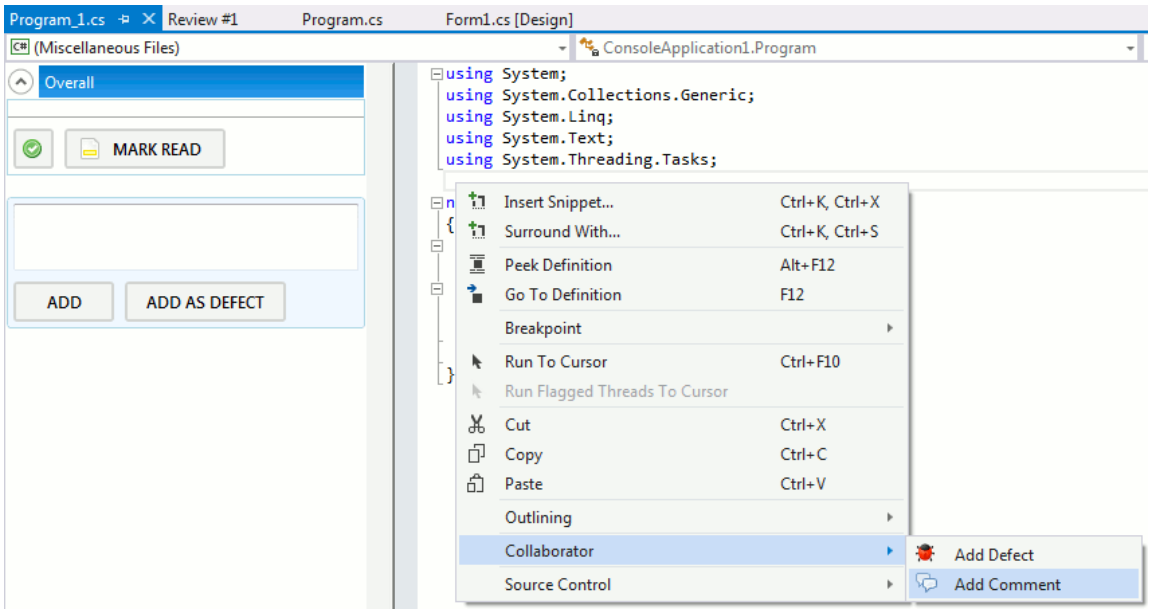
1. Specify your message in the Overall section of the Conversation pane.
2. Press **Add as Defect** and specify defect type and severity.

To create a line comment:

1. Right-click the desired line and choose **Collaborator | Add Comment** from the context menu.
2. Specify your message in the subsequent dialog and press **Add comment**.

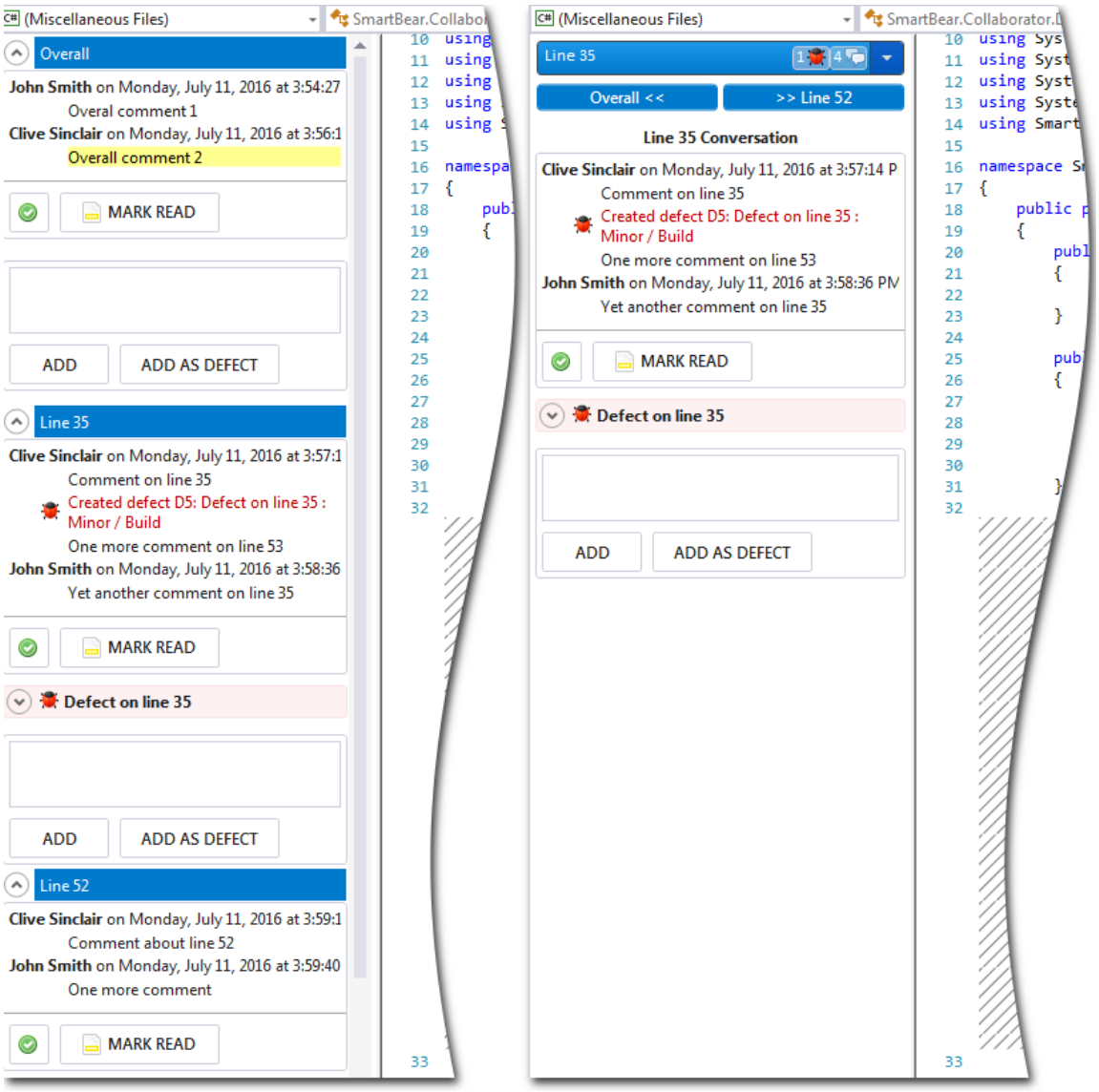
To create a line defect:

1. Right-click the desired line and choose **Collaborator | Add Defect** from the context menu.
2. Specify your message, defect type and severity in the subsequent dialog and press **Add defect**.




*Creating line comments and defects*

Depending on the Group conversations by lines 569 setting, the Conversation Pane may display all defects and comments simultaneously or group comments and defects that relate to the same line. In the latter layout, you may use combobox and buttons to navigate between conversations that relate to different lines.




*Conversation pane: "Standard" and "Group by lines" layouts*

## 5.7 Simulink Reviewer App

This chapter describes Collaborator Simulink Reviewer App, and includes the following sections:

- Overview 596
- Install and Remove 598
- Configure 598

## 5.7.1 Overview

The Collaborator Simulink Reviewer App makes it easy to create or update Simulink model reviews 404 with just a few clicks. When using the Reviewer App, there is no need to export and upload the model manually to Collaborator - you can upload models directly from MathWorks Simulink.

### Get Started

1. Download and install the app 598.

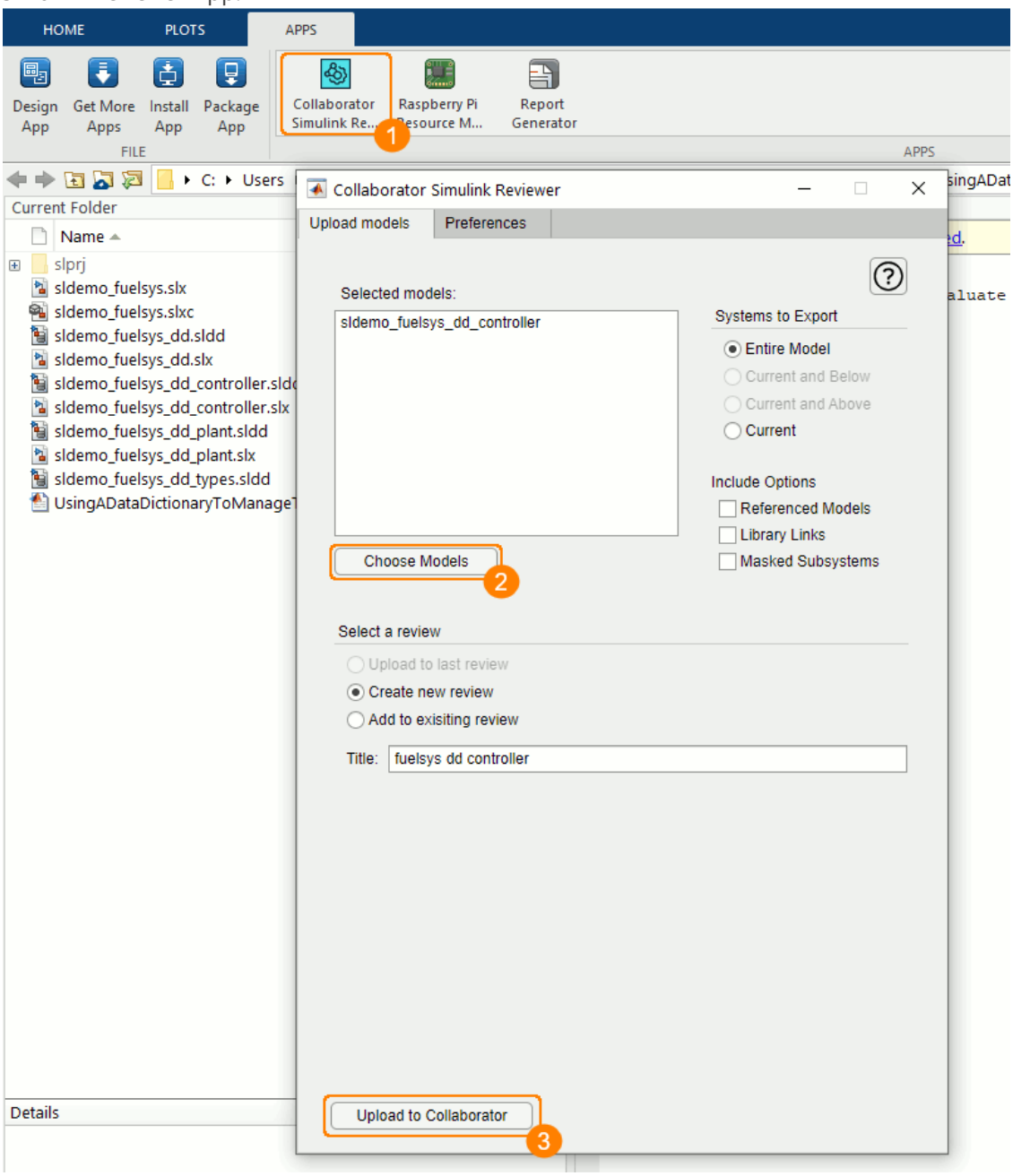
2. Launch MathWorks Simulink.

3. Specify Collaborator server and user credentials 607 in app preferences.

### How the Integration Works

1. (Optional.) Open the desired model in MathWorks Simulink.

2. Switch to the **Apps** tab and click **Collaborator Simulink Reviewer**. This will launch the Simulink Reviewer App.




3. In the subsequent dialog, switch to the **Upload models** tab, press **Choose models** and select one or more models you want to upload.

4. Select which systems to export and specify include options.

5. Specify whether to upload model to the last review that was created with Simulink Reviewer App , to create a new review or to upload it to some existing review.

6. Click **Upload to Collaborator**.

### Requirements

- MathWorks Simulink version 2019 and newer with Simulink Report Generator is installed.
- Collaborator Enterprise edition with additional Simulink integration licenses is installed.
- Simulink integration license is assigned 210 to your user account.

## 5.7.2 Install and Remove

### Install the App

1. Download Simulink Reviewer App from our web site:
   https://support.smartbear.com/collaborator/downloads/simulink-app/
2. Open the downloaded *Collaborator Simulink Reviewer.mlappinstall* file.
3. Click **Install**.
4. Wait till the installation is complete.

After installing the App, you need to configure 598 its settings before the first use.

### Remove the App

1. Open MathWorks MATLAB.
2. Switch to **Home** tab and click the down-arrow next to **Add-Ons** button.
3. In the drop-down menu select **Manage Add-Ons**.
4. Locate the Collaborator Simulink Reviewer App and click **Options > Uninstall**.

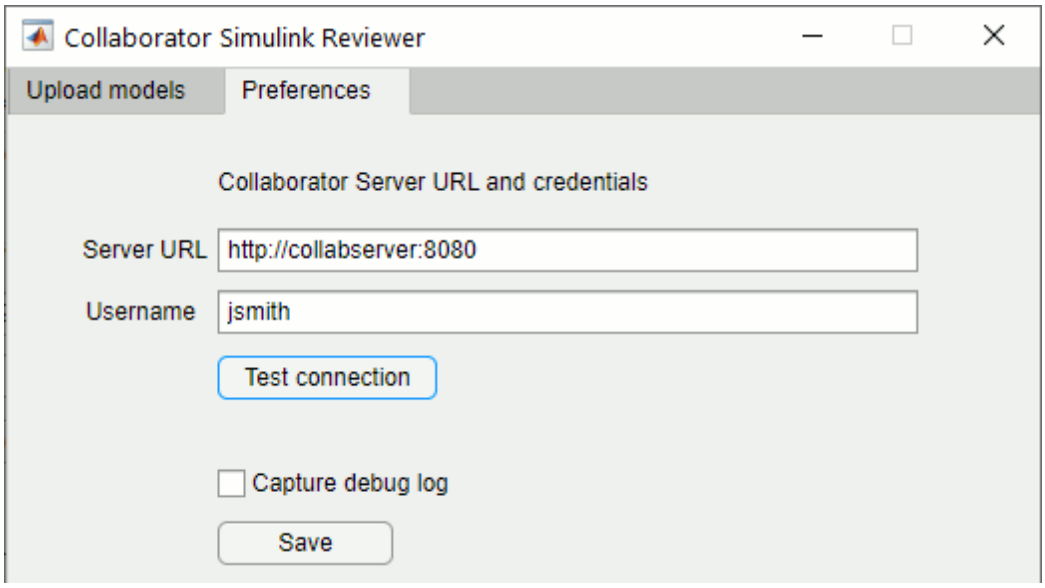
## 5.7.3 Configure

Before using the Simulink Reviewer App, you must specify Collaborator server and user credentials.

## Settings

To change the app settings:

1. Open MathWorks MATLAB.
2. Switch to the **Apps** tab and click **Collaborator Simulink Reviewer**. This will launch the Simulink Reviewer App.
3. Switch to the **Preferences** tab.




*Collaborator plugin options*

The Simulink Reviewer App have the following settings:

**Server URL** — Denotes which Collaborator server to use. The Server URL must include the correct port number and path if applicable.

If your server uses HTTPS, you may need to install its certificate as described below 608.

**Username** — Specifies user credentials. The user name and password are the same as you use when logging into the web server 315 (the password or login ticket 318 is requested when you log in for the first time).

You can specify your password or login ticket 318. When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead.

| | |
|---|---|
| **Capture debug log** | Once enabled, the Simulink Reviewer App will gather debug information and store it at `USERDIR/.smartbear/logs/simulink.app.log`<br><br>Here `USERDIR` refers to the user's home directory (under Windows, your "`Documents and Settings`" Profile directory). |

## Configure HTTPS Connection

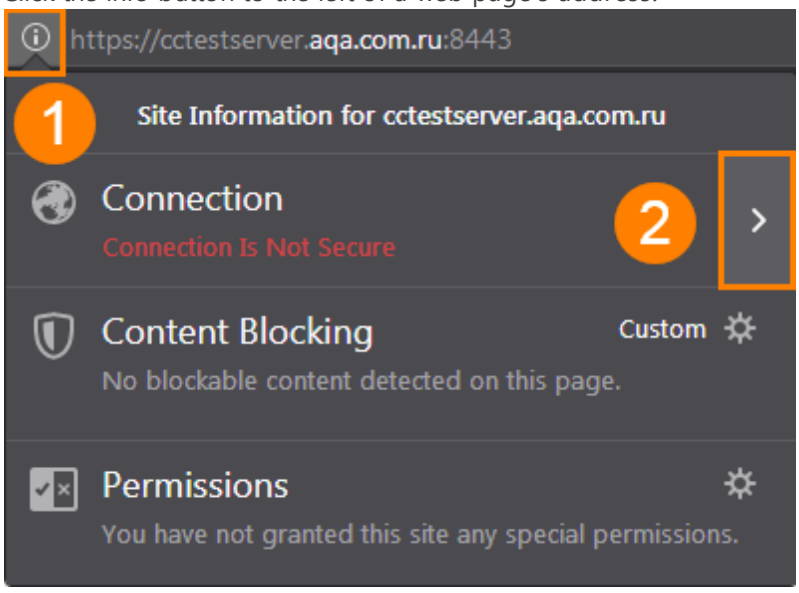
When your Collaborator server uses HTTPS connection and its certificate cannot be automatically verified, or server uses a self-signed certificate, then you will need to install the certificate manually.

**Note on self-signed certificates:** You have to use certificate signed with Certificate Authority (CA). It can be any CA - even yourself. And you must install that CA certificate in trusted authorities. You cannot use self-signed certificate directly.

### Obtaining Certificate

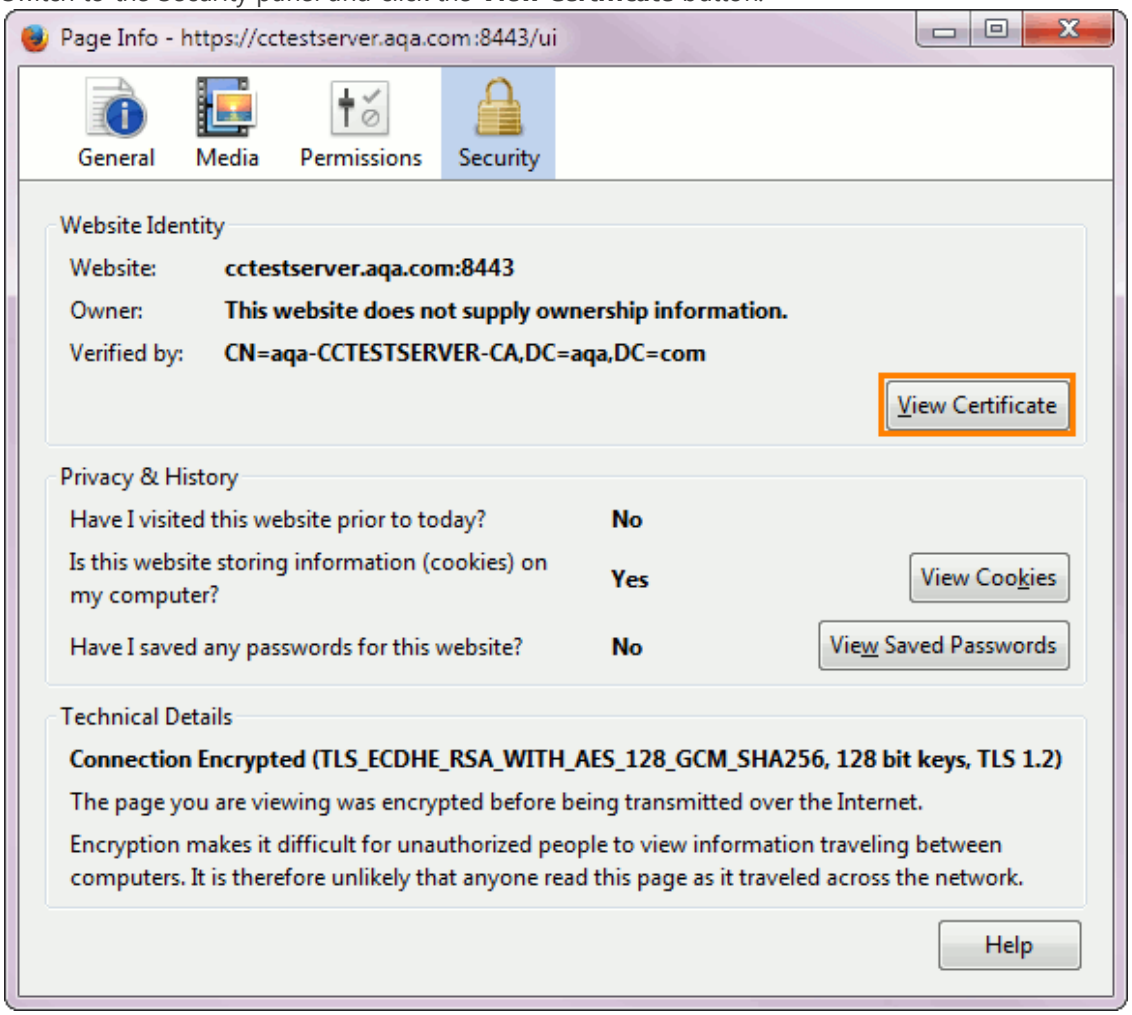
At first, you need to obtain the certificate from the Collaborator server administrator, or obtain it yourself as described below:

1. Launch Firefox browser and navigate to Web UI of your Collaborator server.
2. Click the info button to the left of a web page's address.



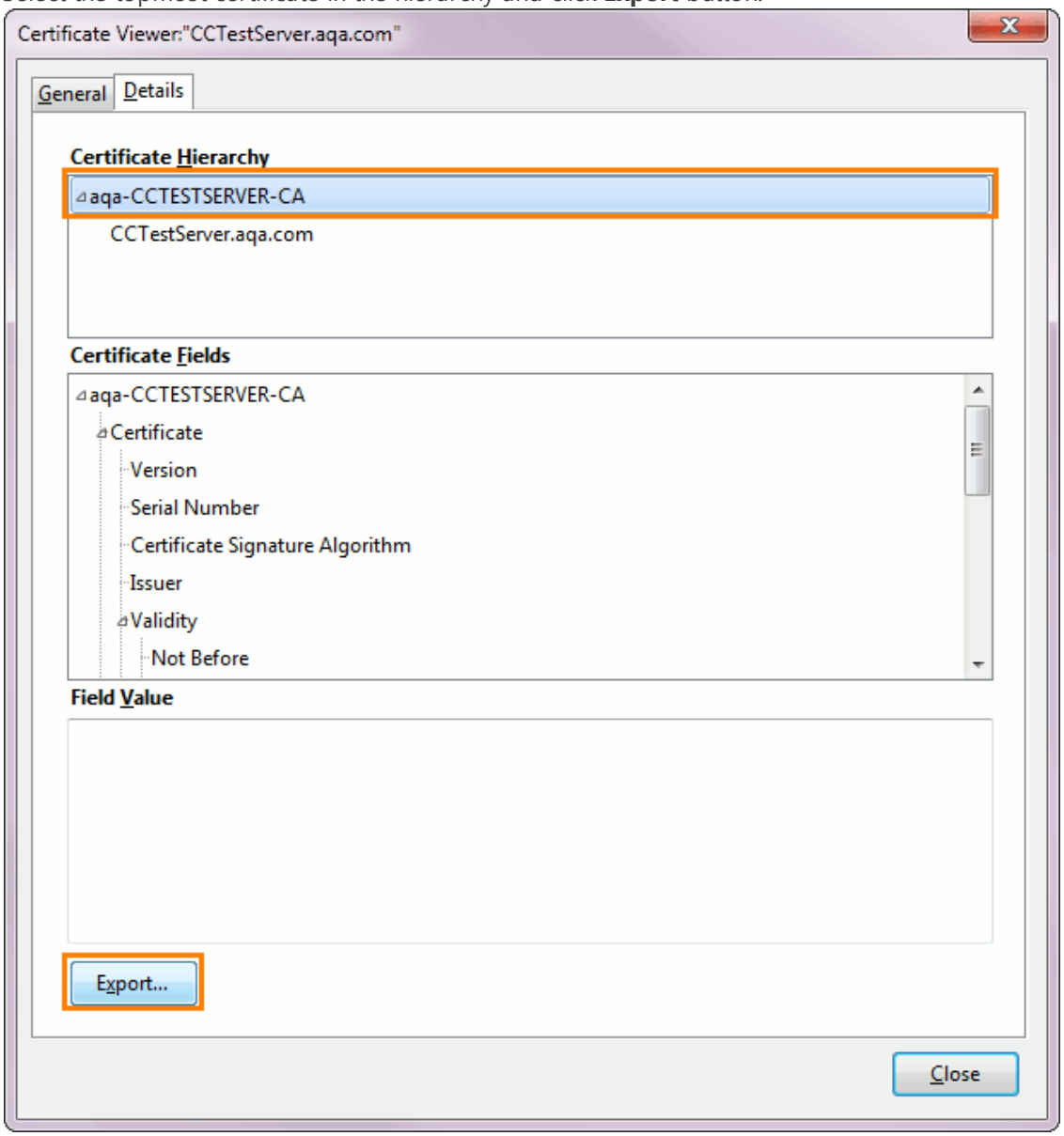

3. Click the **More Information** button in the next prompt. This will open the Page Info window.

4. Switch to the Security panel and click the **View Certificate** button.




5. In the ensuing Certificate Viewer dialog, switch to the Details tab.

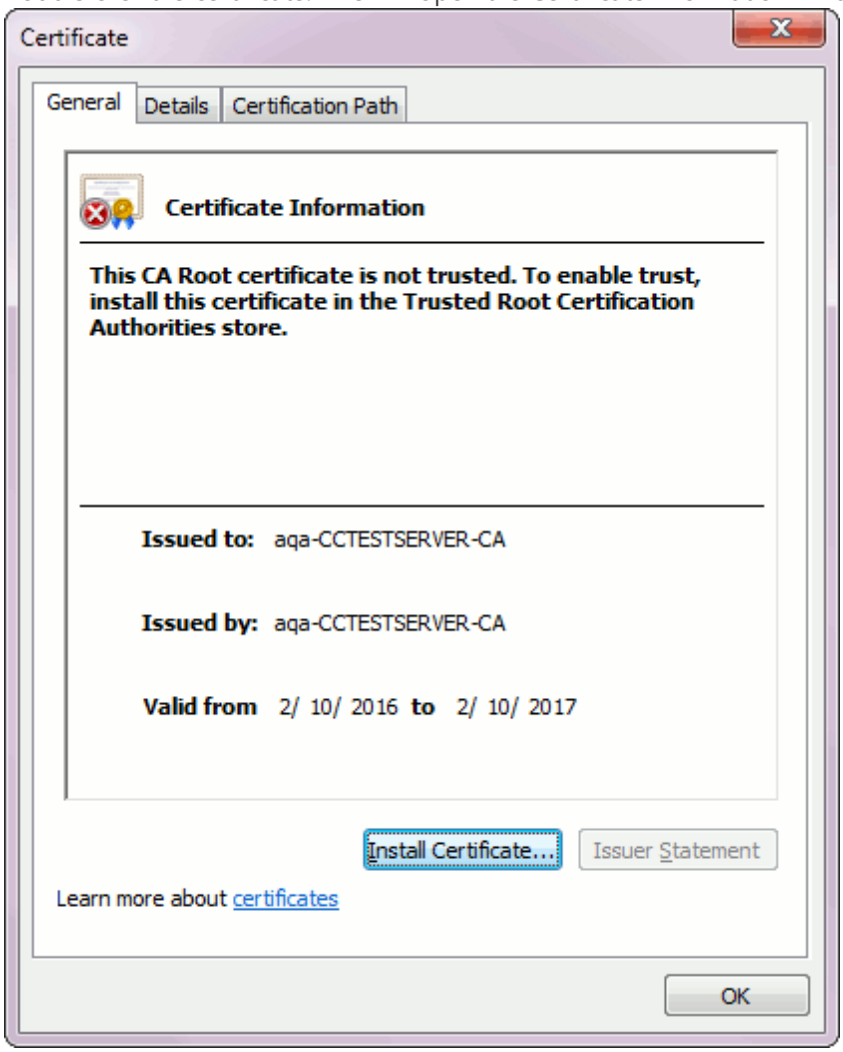
6. Select the topmost certificate in the hierarchy and click **Export** button.




7. Specify a path where to save the certificate in a standard Save File dialog.

## Installing Certificate

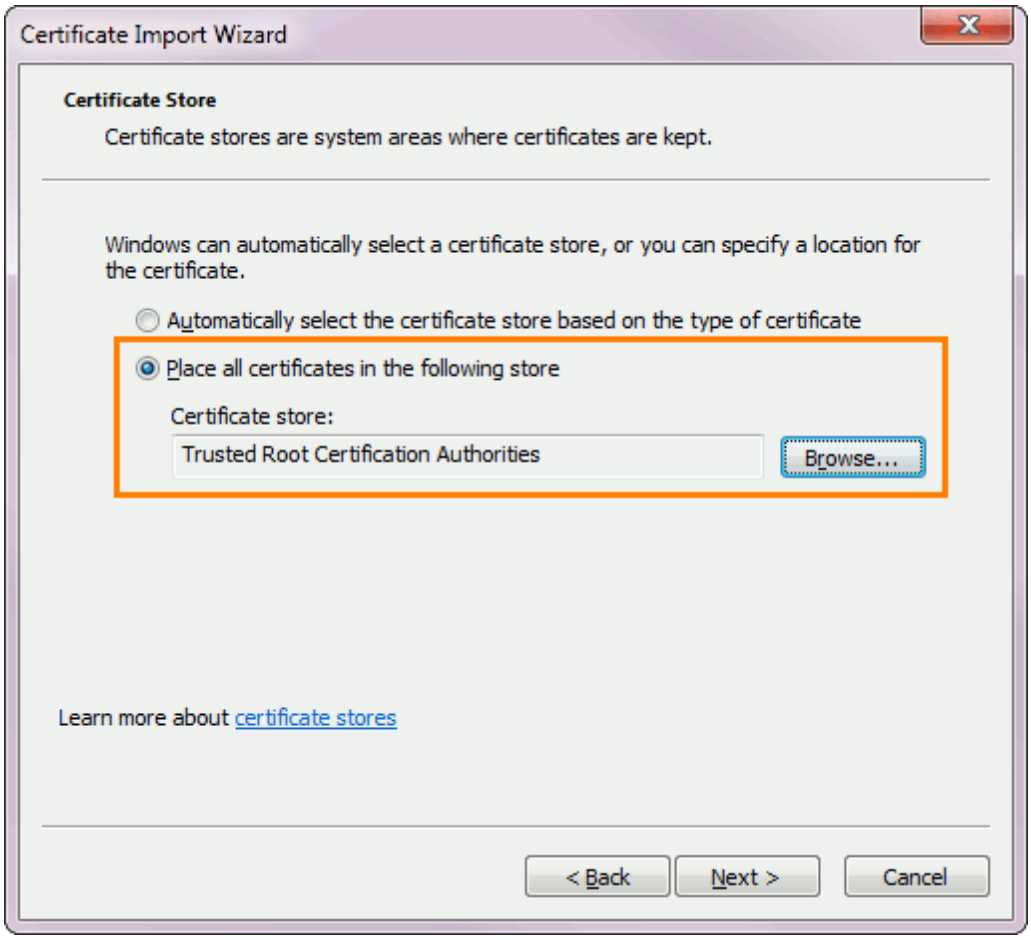
1. Double click the certificate. This will open the Certificate Information window.




2. Click **Install Certificate** button.

3. In the ensuing Import Certificate Wizard, click **Next** to proceed to the Certificate Store page.

4. Choose the "Place all certificates in the following store" option, click Browse and select the " Trusted Root Certificate Authorities" store.




5. Click **Next** and **Finish**.

6. For self-signed certificates, a Security Warning will be displayed. Press **Yes** to confirm certificate installation.

## 5.8 Microsoft Office Plug-ins

This chapter describes Collaborator plug-ins for Microsoft Office, and includes the following sections:

- Microsoft Office Plug-ins Overview 605
- Install and Remove the Plug-ins 606
- Configure Microsoft Office Plug-ins 607

## 5.8.1 Overview

The Collaborator plug-in for Microsoft Word, PowerPoint or Excel makes it easy to create or update reviews with just a few clicks. When using the plug-in, there is no need to save the file and upload it manually to Collaborator - you can upload documents directly from Word, PowerPoint or Excel.

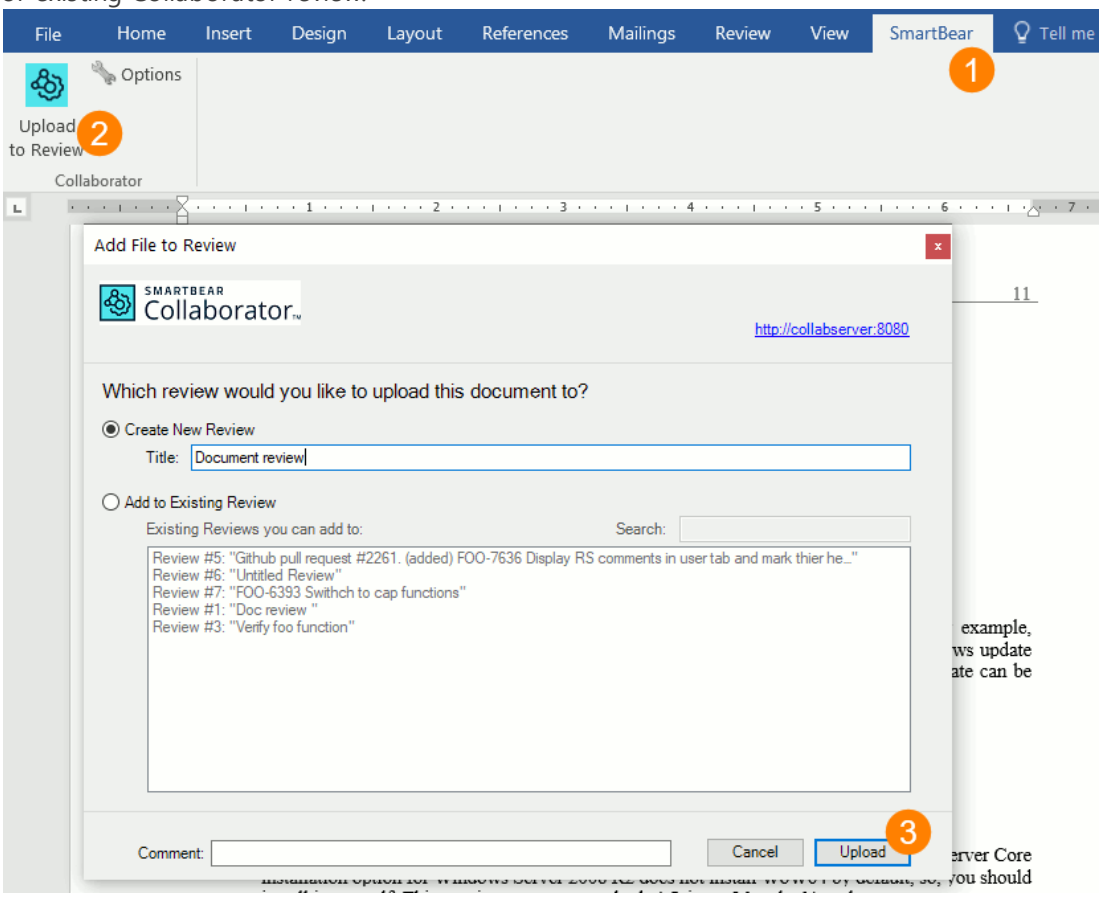
### Get Started

1. Download and install the plug-ins 606.
2. Launch Microsoft Word, PowerPoint or Excel.
3. Specify Collaborator server and user credentials 607 in plug-in options.

### How the Integration Works

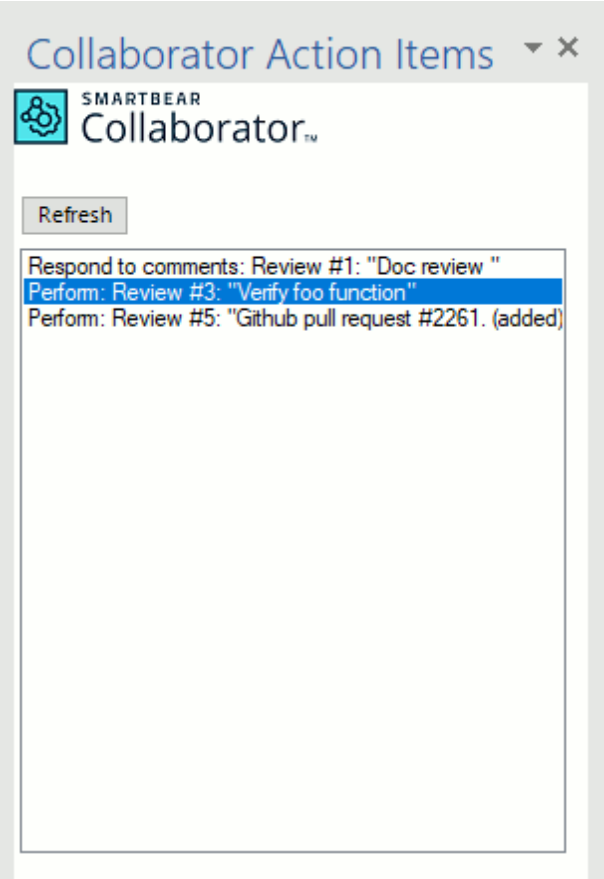
The plug-in adds a number of panels, menus and settings to the Microsoft Office applications.

- The **SmartBear > Upload to review** menu item allows uploading current document to new or existing Collaborator review.




*Uploading document to review*

- The **SmartBear > Options** menu item allows re-configuring connection settings[607].
- The **Collaborator Action Items** pane displays information about incoming and outgoing reviews and allows opening them in Web Client.




*Collaborator Action Items*

## Requirements

- Microsoft Office 2019, 2017 or 2015.

## 5.8.2 Install and Remove

### Install the Plug-in

1. Download Microsoft Office Plug-ins from our website:
   https://support.smartbear.com/downloads/collaborator/

2. Unpack the downloaded .zip archive.

3. Navigate to **Word**, **PowerPoint** or **Excel** subfolder and launch *setup.exe*

4. Click **Install**.

5. Wait till the installation is complete and click **Close**.

After installing the plug-in, you need to configure its settings before the first use.

## Remove the Plug-in

1. In the search box on the taskbar, type **Control Panel** and select it from the results.

2. Select **Programs > Programs and Features**.

3. Locate the desired plug-in (**SmartBear Collaborator for Word**, **SmartBear Collaborator for PowerPoint** or **SmartBear Collaborator for Excel**) and click **Uninstall**.

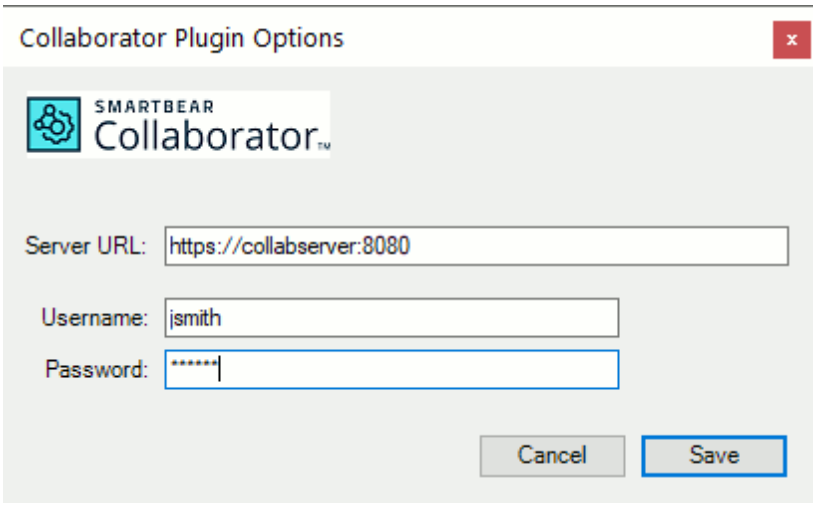
# 5.8.3 Configure

Before using the Collaborator plug-ins for Microsoft Word, PowerPoint or Excel, you must specify Collaborator server and user credentials.

When you open the Microsoft Word, PowerPoint or Excel and the plug-in is not configured yet, it will suggest to do so.

## Settings

To change the plug-in settings, select **SmartBear > Options** from the main menu.




*Collaborator plugin options*

The Collaborator plug-ins have the following settings:

**Server URL** — Denotes which Collaborator server to use. The Server URL must include the correct port number and path if applicable.

If your server uses HTTPS, you may need to install its certificate as described below 608.

**Username** and **Password** — Specifies user credentials. The Username and Password are the same as you use when logging into the web server 315. It is not recommended to use empty password.

You can specify your password or login ticket 318. When single sign-on authentication is disabled, specify password. When single sign-on authentication is enabled, specify login ticket instead.

## Configure HTTPS Connection

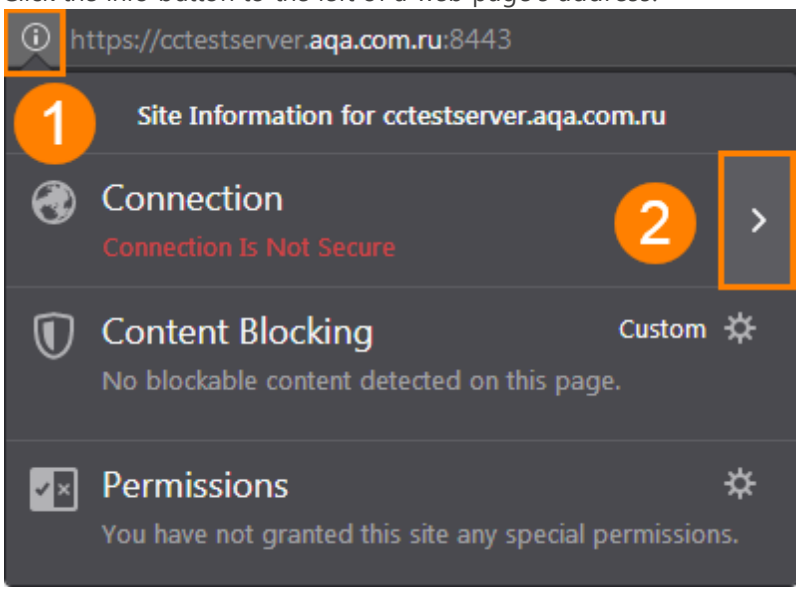
When your Collaborator server uses HTTPS connection and its certificate cannot be automatically verified, or server uses a self-signed certificate, then you will need to install the certificate manually.

**Note on self-signed certificates:** You have to use certificate signed with Certificate Authority (CA). It can be any CA - even yourself. And you must install that CA certificate in trusted authorities. You cannot use self-signed certificate directly.

### Obtaining Certificate

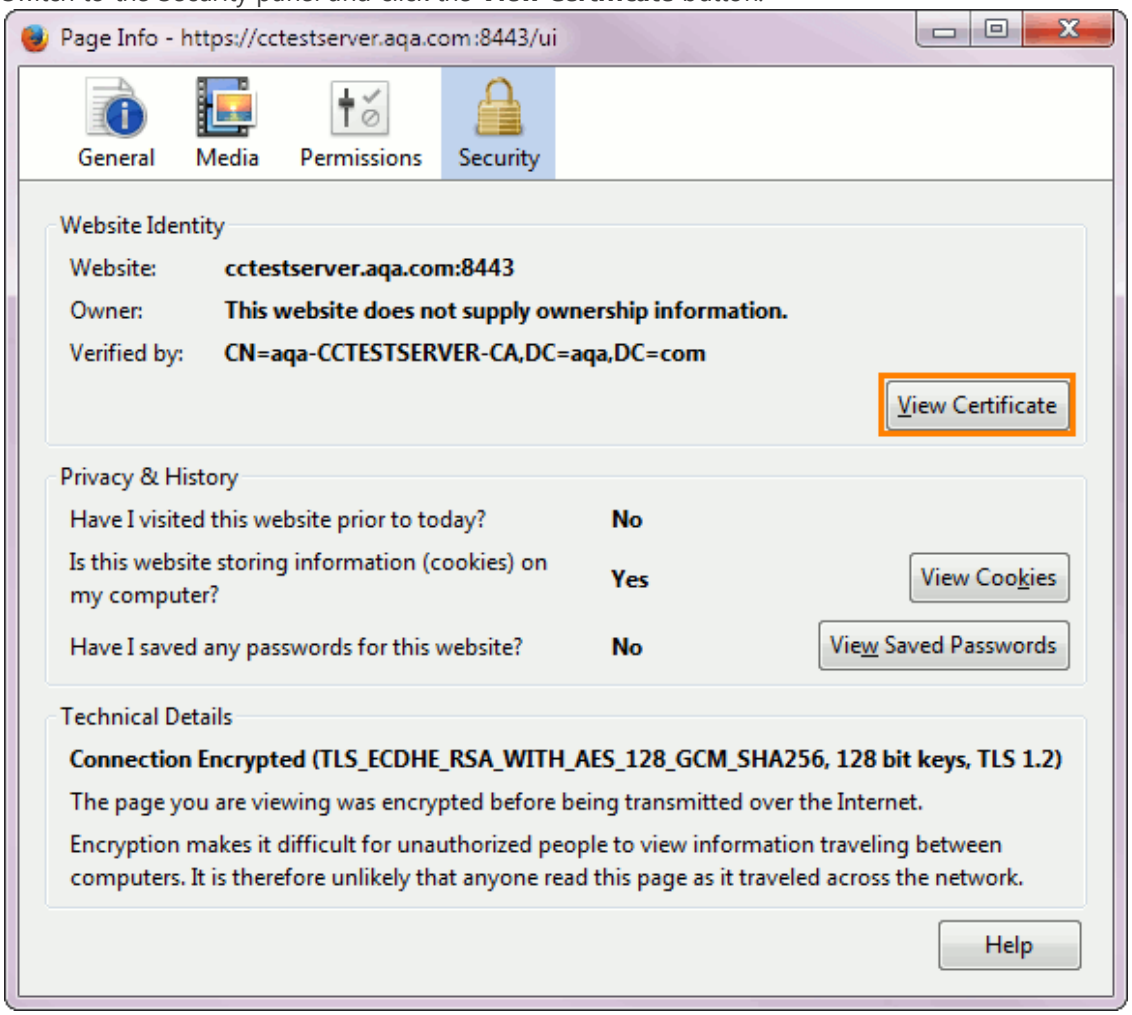
At first, you need to obtain the certificate from the Collaborator server administrator, or obtain it yourself as described below:

1. Launch Firefox browser and navigate to Web UI of your Collaborator server.
2. Click the info button to the left of a web page's address.



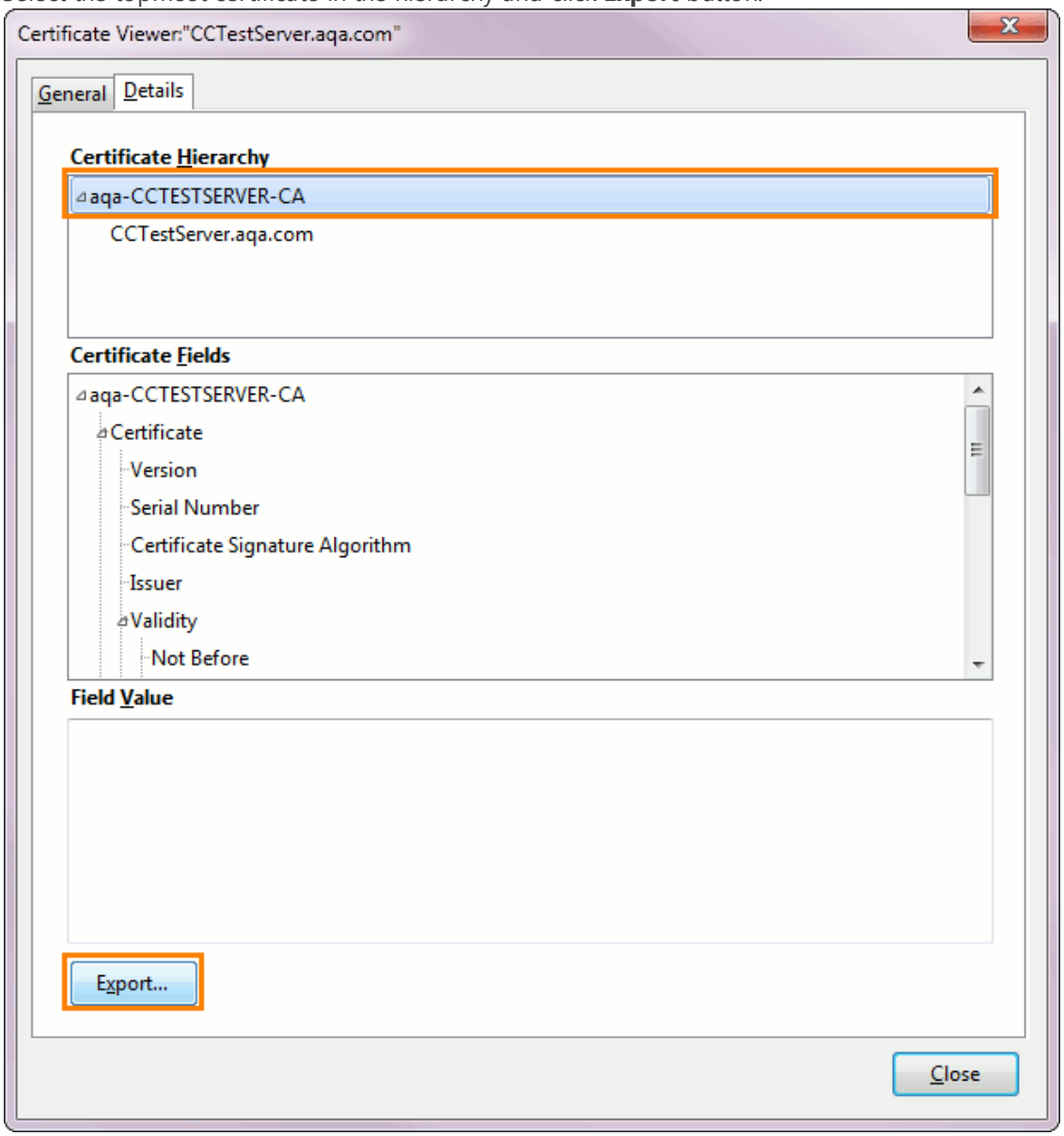

3. Click the **More Information** button in the next prompt. This will open the Page Info window.

4. Switch to the Security panel and click the **View Certificate** button.




5. In the ensuing Certificate Viewer dialog, switch to the Details tab.

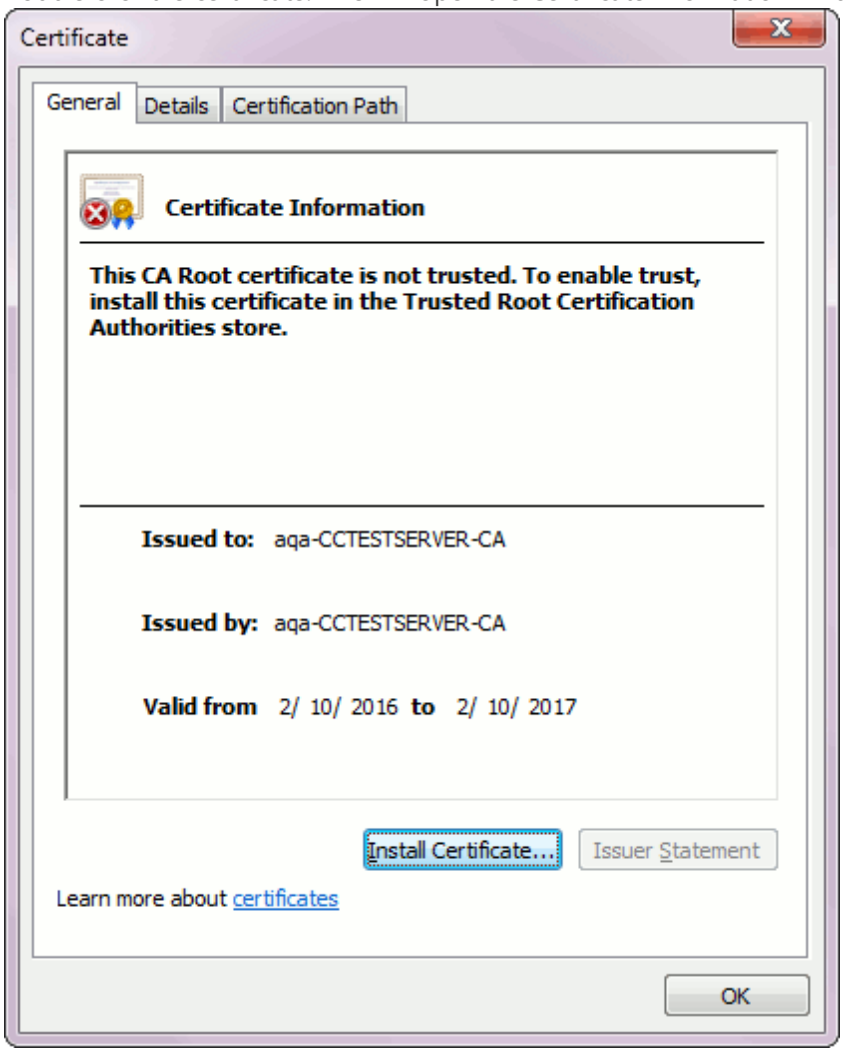
6. Select the topmost certificate in the hierarchy and click **Export** button.




7. Specify a path where to save the certificate in a standard Save File dialog.

## Installing Certificate

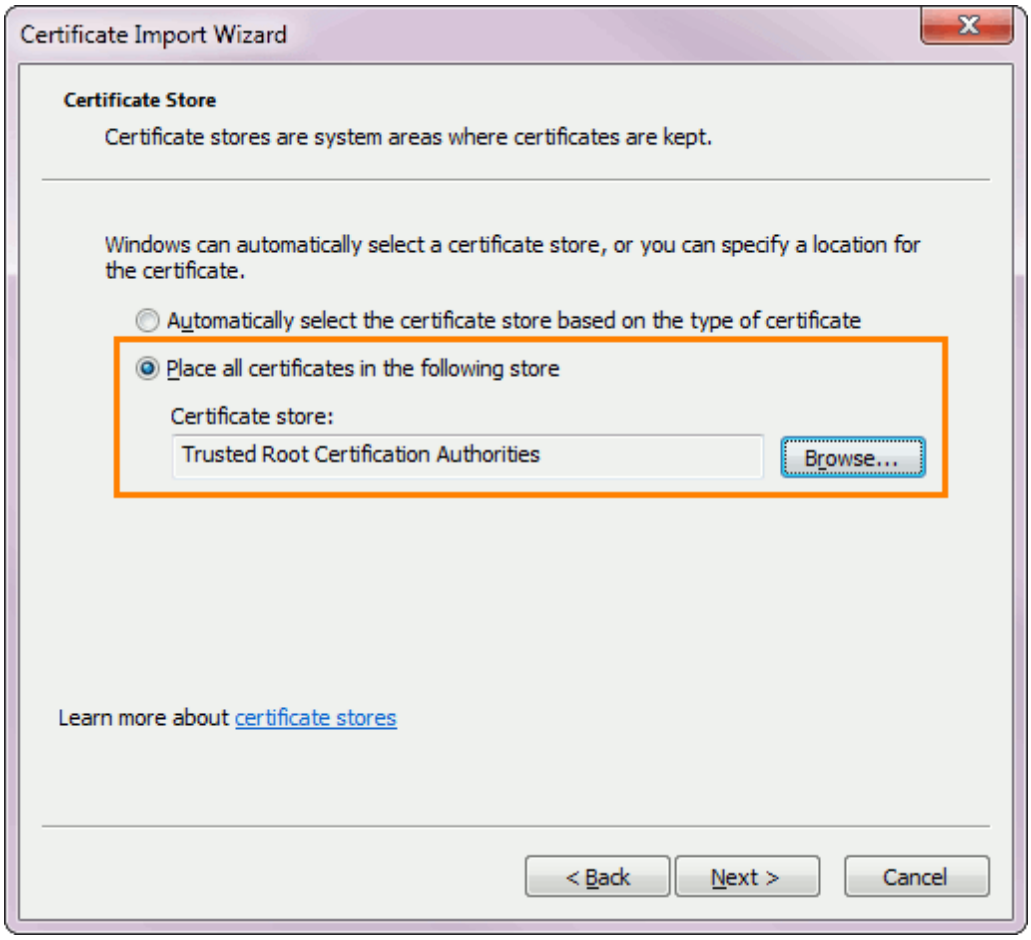
1. Double click the certificate. This will open the Certificate Information window.




2. Click **Install Certificate** button.

3. In the ensuing Import Certificate Wizard, click **Next** to proceed to the Certificate Store page.

4. Choose the "Place all certificates in the following store" option, click Browse and select the " Trusted Root Certificate Authorities" store.




5. Click **Next** and **Finish**.

6. For self-signed certificates, a Security Warning will be displayed. Press **Yes** to confirm certificate installation.

## 5.9 Tray Notifier

The Tray Notifier is a cross-platform tray notifier client interface to the Collaborator server. It can be used to provide an easy way to see what reviews you are currently participating in, and prompt notification when a review requires your attention.

### Starting the Tray Notifier

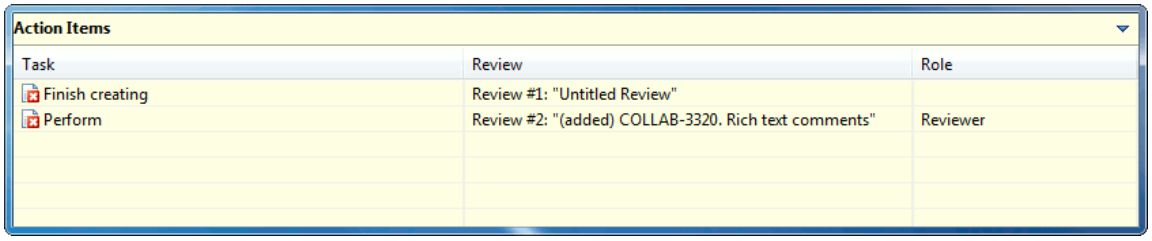
The Tray Notifier is an executable named `ccollabtray.` On Windows machines the graphical installer 487 includes an option to automatically run the Tray Notifier on startup (a shortcut to `ccollabtray` is placed in the Windows Start Menu Startup folder). Non-Windows platforms have to manually start `ccollabtray.`


## Configuring the Tray Notifier

If you used the graphical installer[487], your connection to the Collaborator Server should be configured already. Otherwise, you will be prompted when you try to connect. You can also select the *Preferences* item in the context menu[614] to open the preferences dialog. Use the *Server Connection Preferences*[494] page to configure the connection information to your Collaborator Server.

## Using the Tray Notifier

Clicking on the Tray Notifier opens a list of your current Action Items.



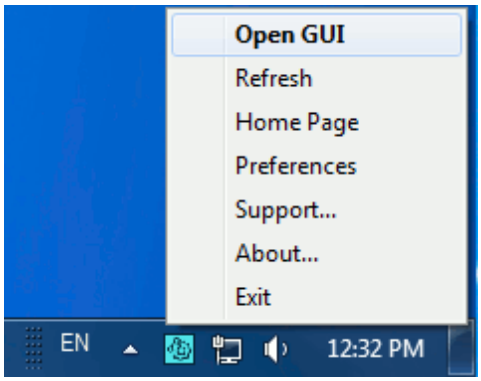

*Tray Notifier Action Items*

The Action Items list can be resized and moved by dragging it is borders. Action Items which are urgent appear with a red icon. Action Items which are not urgent appear with a green icon. Double-clicking on an Action Item opens a page in a browser window which lets you perform the action. Commit Action Items will open a wizard that will walk you through committing your files. Most changes can be committed with the tray notifier, but often changes to the directory structure will require that you use your SCM's tools directly.

The Tray Notifier icon changes to show a check-mark to alert you when there are new Action Items that require your attention.


## Context Menu

The context menu of the Tray Notifier (right-click on Windows) includes several options:




*Tray Notifier context menu*

- *Open GUI* launches GUI Client
- *Refresh* immediately checks the Collaborator Server for updated Action Items
- *Home Page* opens your Collaborator Home Page 333 in a browser window
- *Preferences* opens the client preferences dialog 494
- *About...* tells you information about your installation
- *Exit* shuts down the Tray Notifier

# 5.10 External Diff Viewer Launcher

## About

External Diff Viewer Launcher is a cross-platform helper utility that allows comparing review materials in any third-party diff viewer instead of Web Client's diff viewer 364. This utility launches the selected external diff viewer on your local computer, retrieves diff data from the Collaborator server and transforms it to the format used by the external diff viewer.

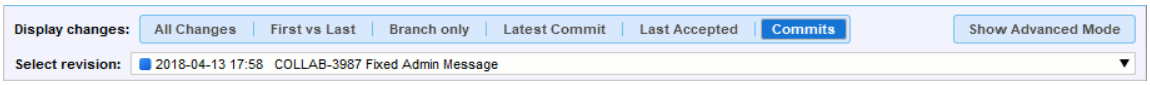
## Prerequisites

- **Install Collaborator Client** 486 - External Diff Viewer Launcher is part of the Collaborator desktop client package.
- **Install your preferable diff viewer** - External Diff Viewer Launcher can work with almost any third-party diff viewer, and has presets for 5 most popular diff viewers.
- **Configure External Diff Viewer Launcher** 617 - Specify the arguments to be used when invoking an external diff viewer.
- **Associate the launcher with Collaborator diff files** 618 - Process Collaborator diff files by using External Diff Viewer Launcher.

## Launch an external diff viewer for a single file

To compare two revisions of a single file:

- Open the desired file in Web Client's diff viewer 364.
- Select the revisions you want to compare using Select revision, Display changes or other controls of the Diff Viewer header 369.




*Diff Viewer header*

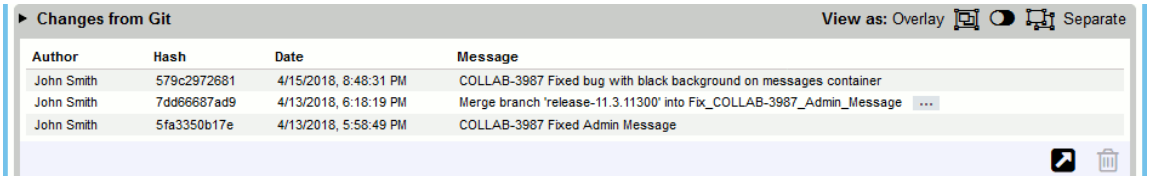
- Press the **External Diff Viewer** button on the Diff Viewer toolbar.


This will download the Collaborator diff file for the chosen revisions of the file, and, if file type association 618 was made, will launch the external diff viewer and compare the revisions in it.

## Launch an external diff viewer for multiple files

To compare the changes made to multiple files (two directory trees):

- Open the desired review in the Review Screen 345.
- Scroll to the **Review Materials** section and find the desired changelist.
- Press the **External Diff Viewer** button in the Changelist info section.



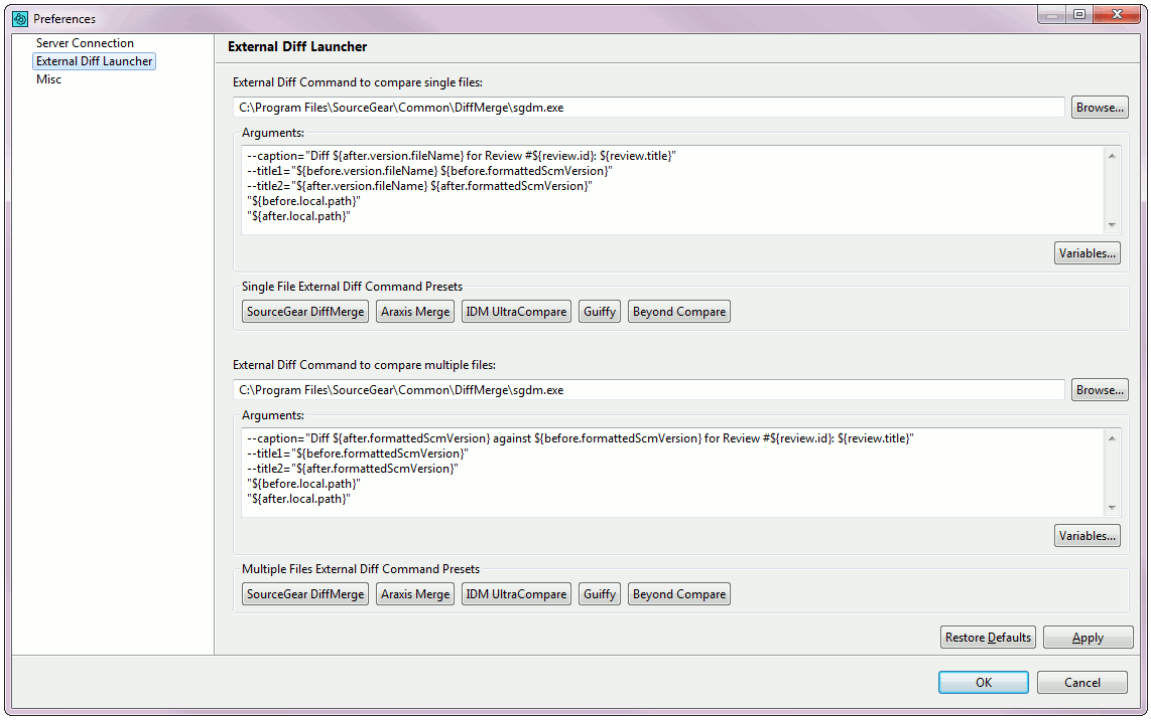

This will download the Collaborator diff file for the chosen set of files, and, if file type association 618 was made, will launch the external diff viewer and compare the files in it.

# Configure the launcher for a particular diff viewer

Since various third-party diff viewers have a different set of command-line arguments, you will have to specify what command-line arguments the launcher should use for your preferable diff viewer. To configure the launcher:

- Open Collaborator GUI Client 496.
- Click **File > Preferences** on the main screen.
- In the ensuing Preferences dialog, click the **External Diff Launcher** item in the left panel.

This will display the External Diff Viewer launcher preferences 503.




- The preferences are divided into two separate groups, one for comparing two versions of a single file, and another for comparing multiple files. Specify the path to your diff viewer executable and the command-line arguments to use for each group.

There are argument presets for 5 most popular diff viewers:

- SourceGear DiffMerge
- Araxis Merge
- IDM UltraCompare

- Guiffy
- Beyond Compare

To use one of the presets, just click the appropriate button.

- Press **OK** to close the dialog and apply the settings.

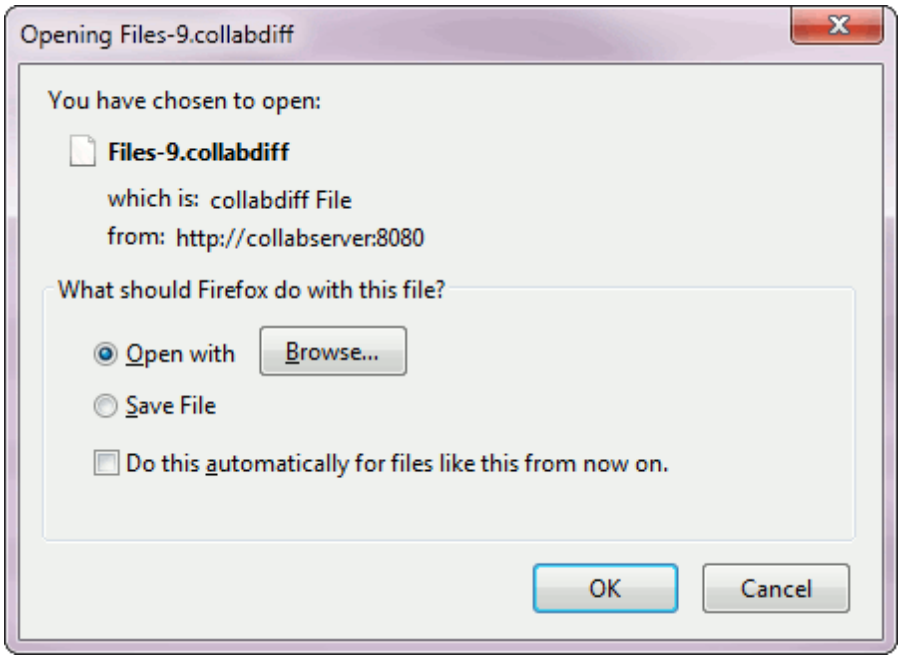
## Associate the launcher with Collaborator diff files

Collaborator server returns diff data in its own internal format. Physically, these are files that have the *.collabdiff* extension. In order to process Collaborator diff files, you need to associate this extension with the External Diff Viewer Launcher executable.

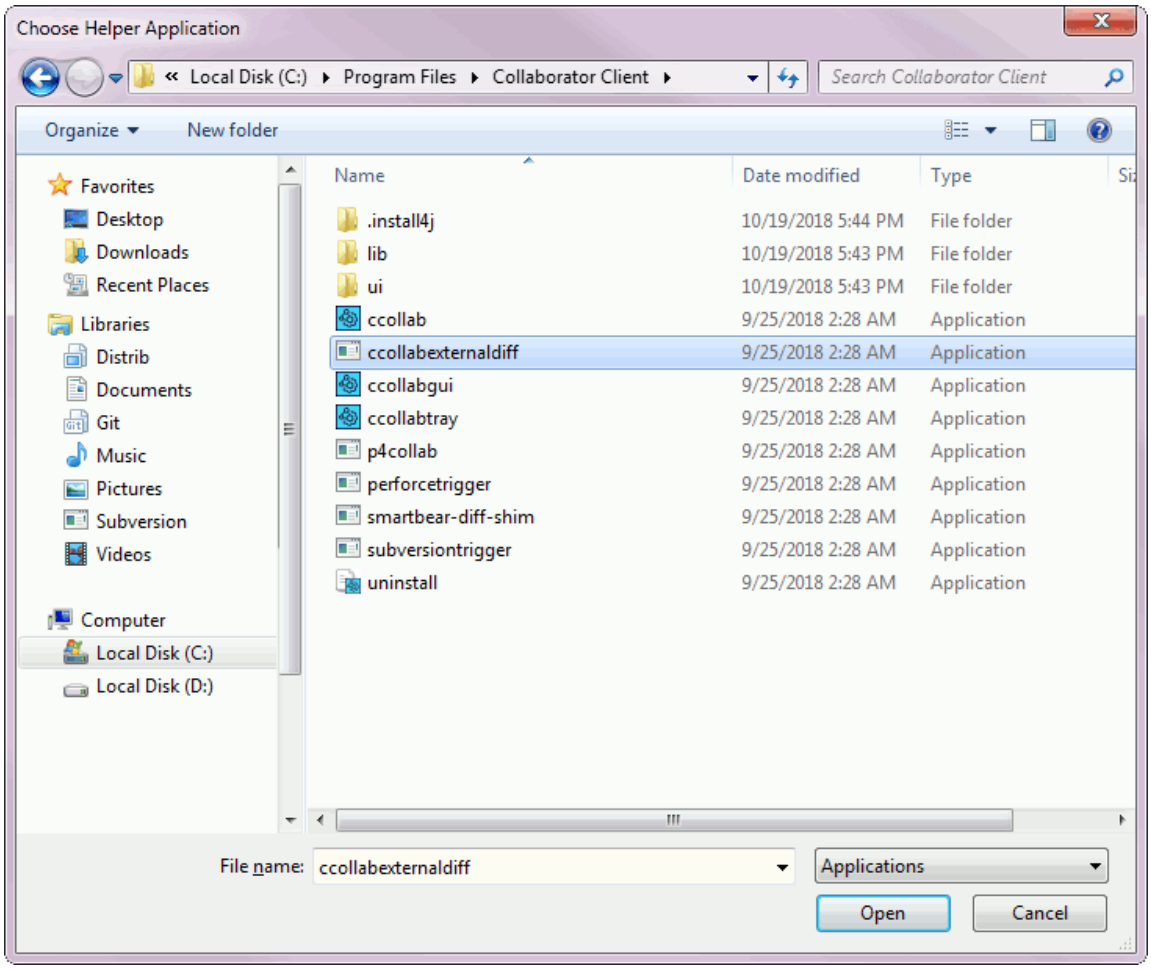
The instructions below describe how you can do this in the Mozilla Firefox browser. The steps may vary slightly in other browsers.

To associate the launcher with Collaborator diff files:

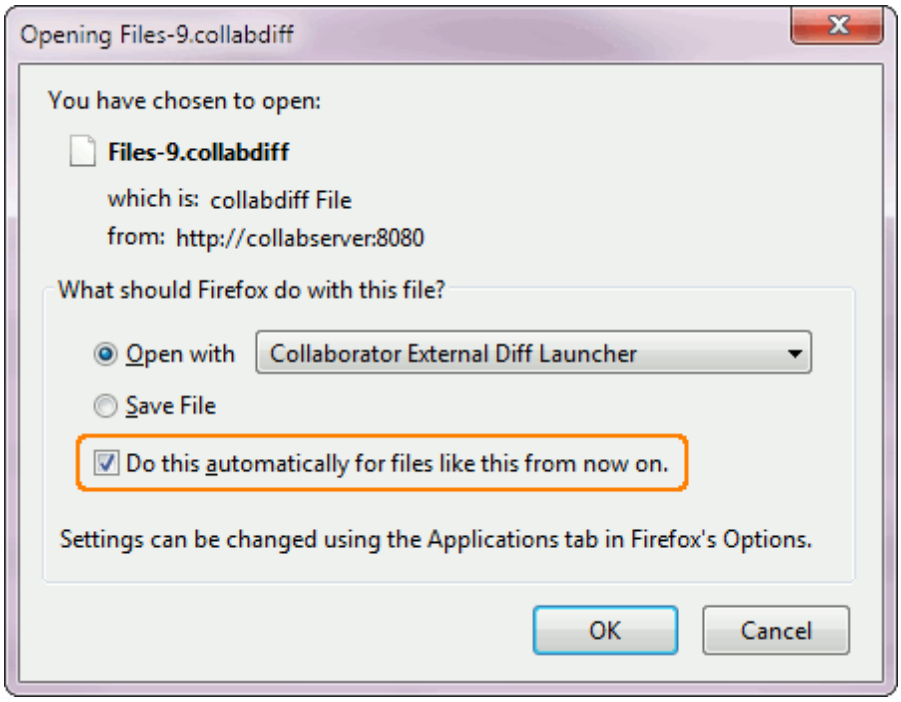
- Open any review in Collaborator Web Client
- Locate any of the  **External Diff Viewer** buttons and click it.
  The Mozilla Firefox browser will suggest that you select an application to open the file with.




- Select **Open with** and click **Browse**. You will be prompted to choose an application:




- Select `ccollabexternaldiff.exe` in the *Collaborator Client* folder and click **Open**.
- In the Opening File window, click **Do this automatically for files like this from now on**:




- Click **OK** to save these settings.

This should be performed only once. Later on, clicking the **External Diff Viewer** buttons in Web Client will automatically invoke the launcher, which in its turn will invoke the diff viewer.

# 6 Version Control Integrations

Collaborator integrates with many popular Version Control Systems. Integrations include command-line, graphical, and Eclipse plug-in clients and Version Control triggers. Server integration is supported for selected Version Control Systems and requires that a properly configured client be configured on the Collaborator server.

***Note:*** *Support of version control systems varies depending on Collaborator edition. Collaborator Community supports only Git and Subversion. Collaborator Team additionally supports CVS, Mercurial, Perforce and Team Foundation Server. Collaborator Enterprise supports all version control systems listed below. For a complete list of differences between Collaborator editions, please see the comparison page* 3.

## In This Section

- IBM Rational ClearCase Integration 666
- IBM Rational Synergy Integration 691
- IBM Rational Team Concert Integration 700
- Mercurial Integration 730
- Microsoft Team Foundation Server Integration 738
- PTC Integrity Integration 752
- Perforce Integration 764
- Subversion Integration 799

## Related Topics of Interest

- Repository Hosting Service Integrations 825
  Describes integrations with remote repository hosting services.

# 6.1 AccuRev Integration

This section describes Collaborator integration with AccuRev:

## GUI Client 623

The GUI Client can upload local changes to files 625 controlled by AccuRev before they are promoted. This includes both kept and modified files in the local Workspace. The GUI Client can also upload pending changes in a Stream 628, differences in a Transaction 626, and arbitrary AccuRev diffs 627.

## Command-Line Client 628

The Command-Line Client can upload local changes to files 629 controlled by AccuRev before they are promoted. This includes both Kept and modified files in the local Workspace. The Command-Line Client can also upload pending changes in a Stream 634, differences in a Transaction 631, and arbitrary AccuRev diffs 632.

## Eclipse Plug-in

The Eclipse Plug-in 525 integrates the AccuRev Plug-in for Eclipse from https://www.microfocus.com/documentation/accurev/ so you can upload 545 locally modified files to a Review.

## Supported Versions

Our Command-Line Client uses your own AccuRev command-line client (`accurev`) to communicate with the server. We support these versions:

- v4.5.3, 5.5

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

## Support for Copy / Move

Collaborator fully supports AccuRev's file and directory copy/move semantics. Thus, if a file is copied or moved rather than added from scratch, it will show up that way in the various user interfaces.

## Support for Directory-level New/Delete/Copy/Move

Collaborator partially supports AccuRev's concept of directories (not just files) being altered.

All files underneath the directories in question will be scanned, uploaded and represented properly in the GUI. The directories themselves will *not* be shown in any GUI, or even in the file list 509 confirmation screen presented by the command-line client.

This all works correctly even if, for example, you move a parent directory *and* alter/add/delete a file within that directory in the same changelist. You will get the correct content for the file but the directory itself will not be listed in the review.

## Support for Include / Exclude Rules

Collaborator treats files or folders that conform to AccuRev's include rules as added or modified files/folders, while files and folders that conform to exclude rules are treated as deletions. To denote files and folders that were added or removed due to include/exclude rules, Collaborator uses the following descriptions of change types:

- `INCLUDED` - The file was added because of include rule.
- `EXCLUDED` - The file was removed because of exclude rule.
- `INCLUDED_DIR` - The folder and all its files and sub-folders were added because of include rule.
- `INCLUDED_DIR_ONLY` - The folder but not any of its files or sub-folders were added because of include rule.
- `EXCLUDED_DIR` - The folder and all its files and sub-folders were removed because of exclude rule.

## Technical Details and Limitations

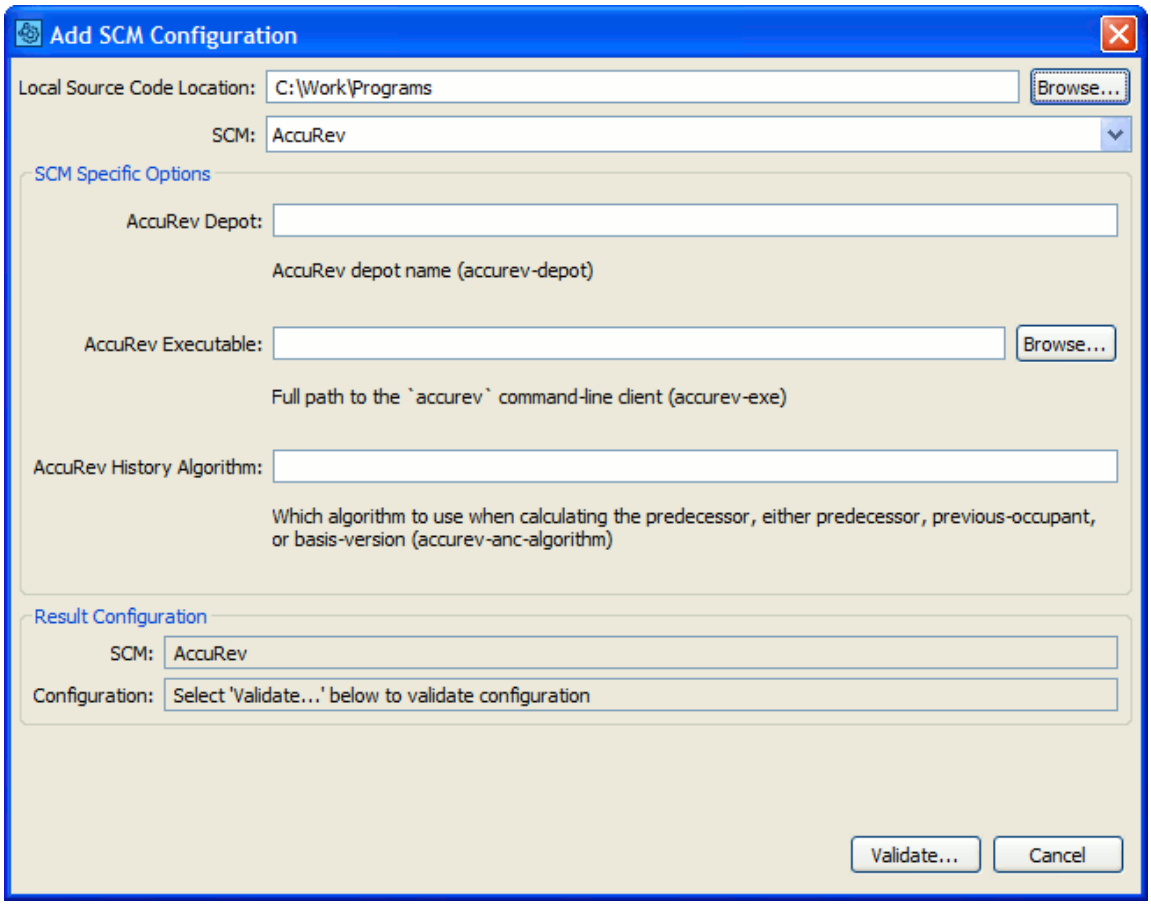
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you add several diffs (non atomic changelists) to the same review.

### 6.1.1 GUI Client

## AccuRev-specific Options

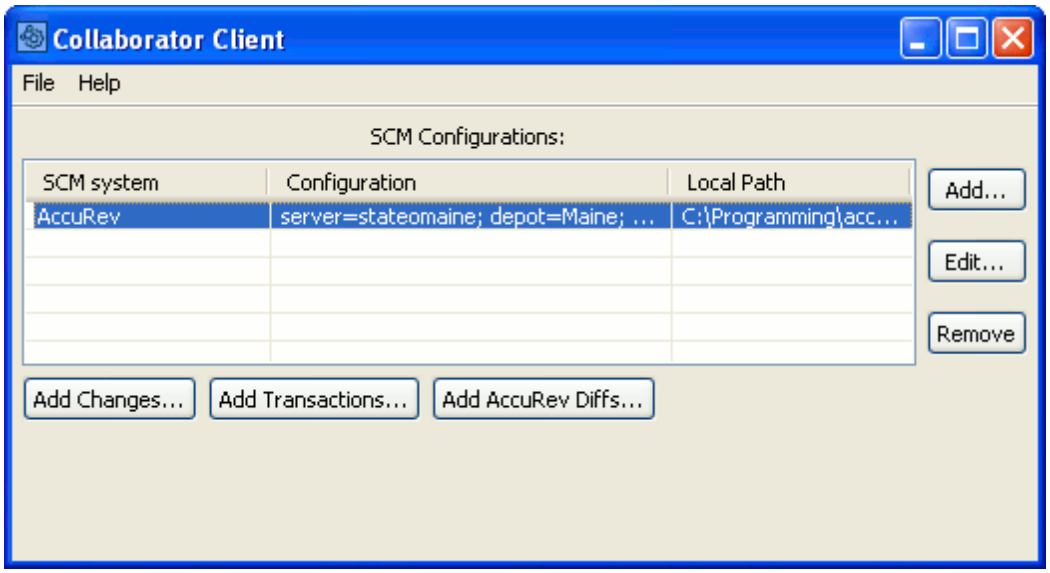
The SCM Configuration dialog 498 has several AccuRev-specific options. If there is no AccuRev Workspace on your machine, you can still review committed AccuRev Transactions by specifying the AccuRev Depot name option.




*AccuRev SCM Configuration*

## Uploading files to a Review

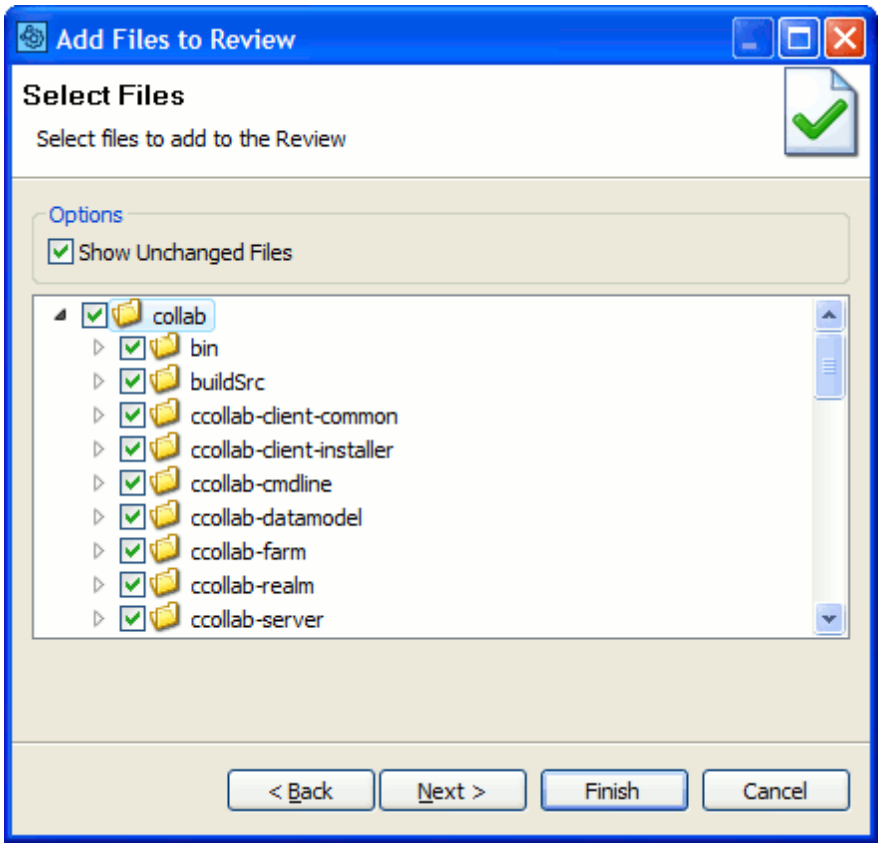
Selecting an AccuRev SCM Configuration in the GUI Client main screen[496] causes three Add to Review buttons to appear. The *Add Changes...*[625] button uploads modified files in a Workspace. The *Add Transactions...*[626] button uploads the files in committed Transactions. The *Add AccuRev Diffs...*[627] button uploads arbitrary diffs, or compares the differences between two Streams[628].




*Uploading AccuRev files to a Review*

## Add Changes

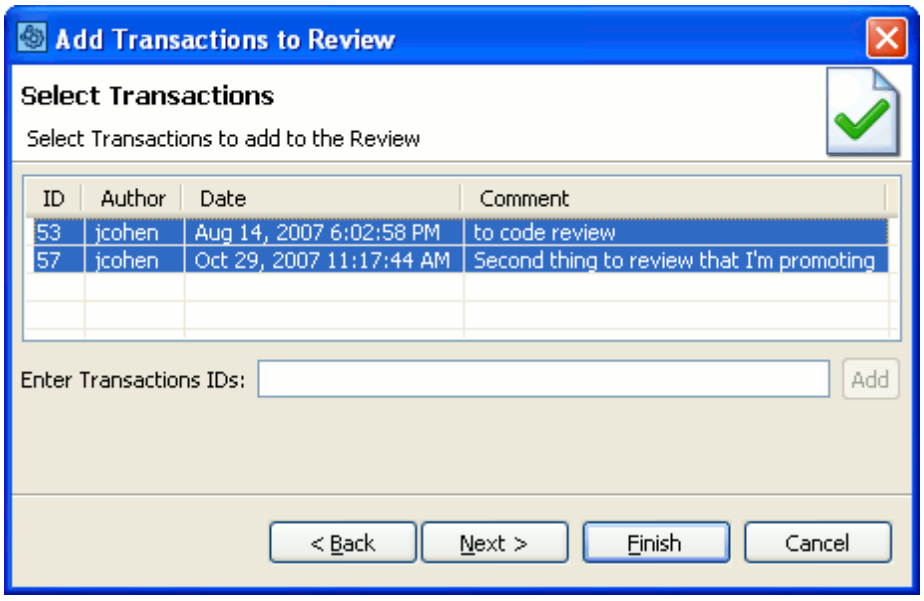
Press the *Add Changes...* button to upload modified files in a Workspace to the Collaborator Server for review.




*Add Changes*

## Add Transactions

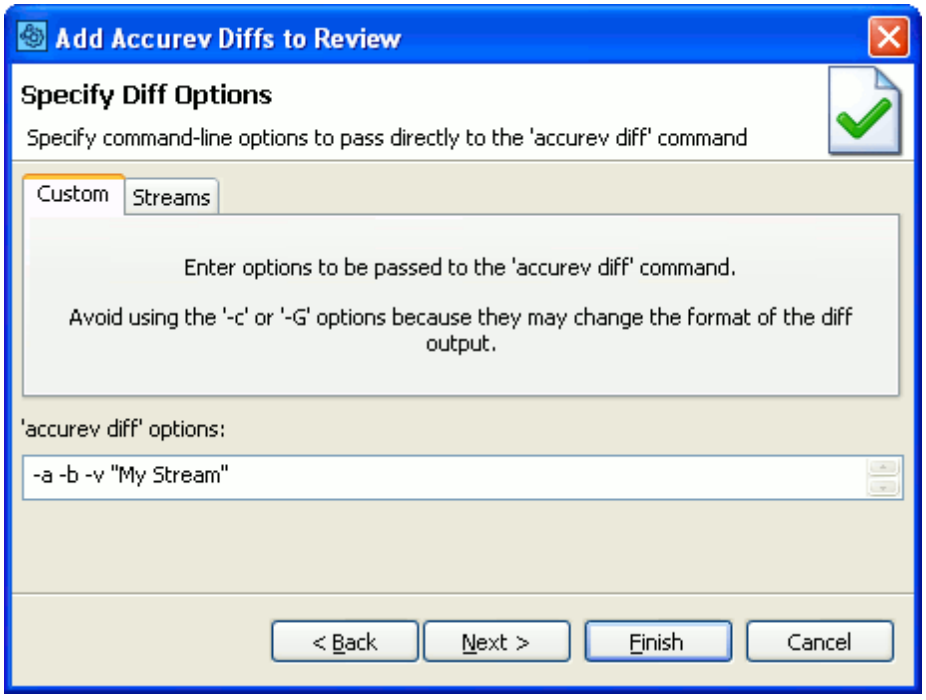
Press the *Add Transactions...* button to upload the files in an AccuRev Transaction to the Collaborator Server for review.




*Add Transactions*

## Add AccuRev Diffs

Press the *Add AccuRev Diffs...* button to upload arbitrary AccuRev diffs to the Collaborator Server for review.

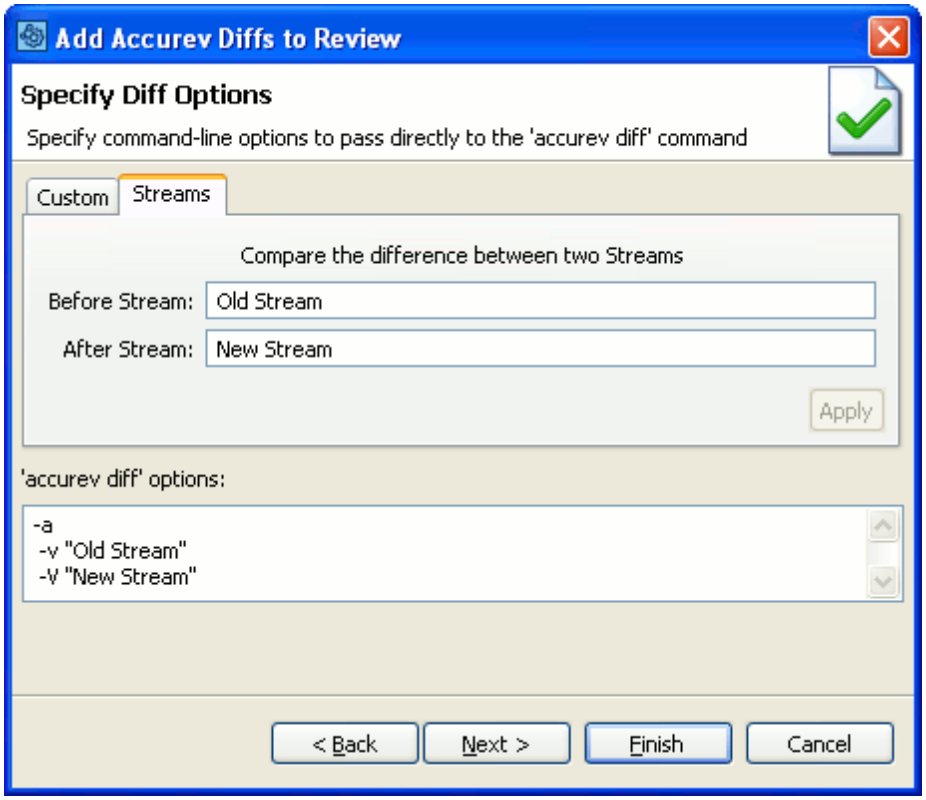



*Add AccuRev Diffs*

You can enter arbitrary AccuRev diff options, or select the Streams tab[628] to compare the difference between two Streams.

#### 6.1.1.1 Comparing two Streams

Press the *Add AccuRev Diffs...*[627] button on the main screen[496] and then select the *Streams* tab to upload the difference between two Streams.




*Upload the difference between two Streams*

### 6.1.2 Command-Line Client

## Commands recommended for AccuRev

`ccollab addchanges`[629] - Attaches locally-modified files to a review

`ccollab addchangelist`[631] - Attaches an atomic changelist to a review

`ccollab addardiffs`[632] - Uploads diffs generated from accurev diff command

`ccollab addstream`[634] - Attaches pending differences from an AccuRev stream

The addchanges 629 command will upload local changes to files controlled by AccuRev before they are promoted. This includes both kept and modified files in the local workspace.

The addchangelist 631 command will upload a list of AccuRev transactions.

## Configuration

In most cases, the Command-Line Client can automatically detect your AccuRev configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your AccuRev configuration or you want to override the detected settings, you can manually specify AccuRev settings using global options 517 .

To manually configure the Command-Line Client to use AccuRev, execute the following command:

```
ccollab set scm 517 accurev
```

## AccuRev-specific Options

| Option | Description |
|---|---|
| `--accurev-depot <value>` | AccuRev depot name |
| `--accurev-exe <value>` | Full path to the `accurev` command-line client |
| `--accurev-anc-algorithm <value>` | Which algorithm to use when calculating the predecessor, either predecessor, previous-occupant, or basis-version |

## Example:

```
ccollab set accurev-depot MyDepot
```

#### 6.1.2.1 addchanges (for AccuRev)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by AccuRev to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on. |

| Option | Required? | Description |
|---|---|---|
| | | After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

#### 6.1.2.2 addchangelist (for AccuRev)

## Description

The `ccollab addchangelist` command attaches all files from a submitted AccuRev changelist (transaction) to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or AccuRev-specific global options. See Command-line Global Options Reference 516. |

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ... ]` | Yes | Identifier(s) of the desired changeset(s) in your source control. |

## Examples:

To upload Transactions t4321 and t7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

To upload Transactions t5432 and t12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

#### 6.1.2.3 addardiffs

## Description

The `ccollab addardiffs` command uploads differences between arbitrary versions of files in AccuRev. The differences are generated using the native 'accurev diff' command of AccuRev.

## Command Line Syntax:

```
ccollab [global-options] addardiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| --upload-comment <value> | No | Comment used to upload files (defaults to command-line arguments) |
| <review> | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <user-diff-arg> [<user-diff-arg> ...] | No | Command-line arguments to pass directly to the diff command |

## Examples:

```
ccollab addardiffs 698 -a -b -v "My Stream"
```

## Remarks:

- Do not use diff arguments that affect the diff output such as '-c' or '-G'. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.
- This command does not upload added or deleted files due to AccuRev limitations.

#### 6.1.2.4 addstream

## Description

The `ccollab addstream` command uploads differences pending promotion in an AccuRev stream.

## Command Line Syntax:

```
ccollab [global-options] addstream [--upload-comment <value>]
<review> [<stream>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<stream>` | No | Specify an AccuRev stream by name, or leave blank to use the current workspace |

# 6.2 CVS Integration

This section describes Collaborator integration with CVS:

## GUI Client 636

The GUI Client can upload local changes to files 637 controlled by CVS before they are checked into version control. The GUI Client can also upload arbitrary CVS diffs 638, or the difference between two Labels 638 or dates 639.

## Command-Line Client 640

The Command-Line Client can upload local changes to files controlled by CVS before they are checked into version control. The Command-Line Client can also upload arbitrary CVS diffs.

### Eclipse Plug-in

The Eclipse Plug-in 525 integrates with the Eclipse CVS team plug-in so you can upload 545 locally modified files with full support from the CVS Eclipse code.

### Supported Versions

Our Command-Line Client 506 uses your own CVS command-line client (`cvs`) to communicate with the server. We support these versions:

- v1.11.1 and later
- v1.11.0 and before, only `ccollab addcvsdiffs` 643 is supported
- `CvsNT` (Windows-only)

The Eclipse Plug-in 525 uses the Eclipse CVS plug-in to communicate with the CVS server.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

### Technical Details and Limitations

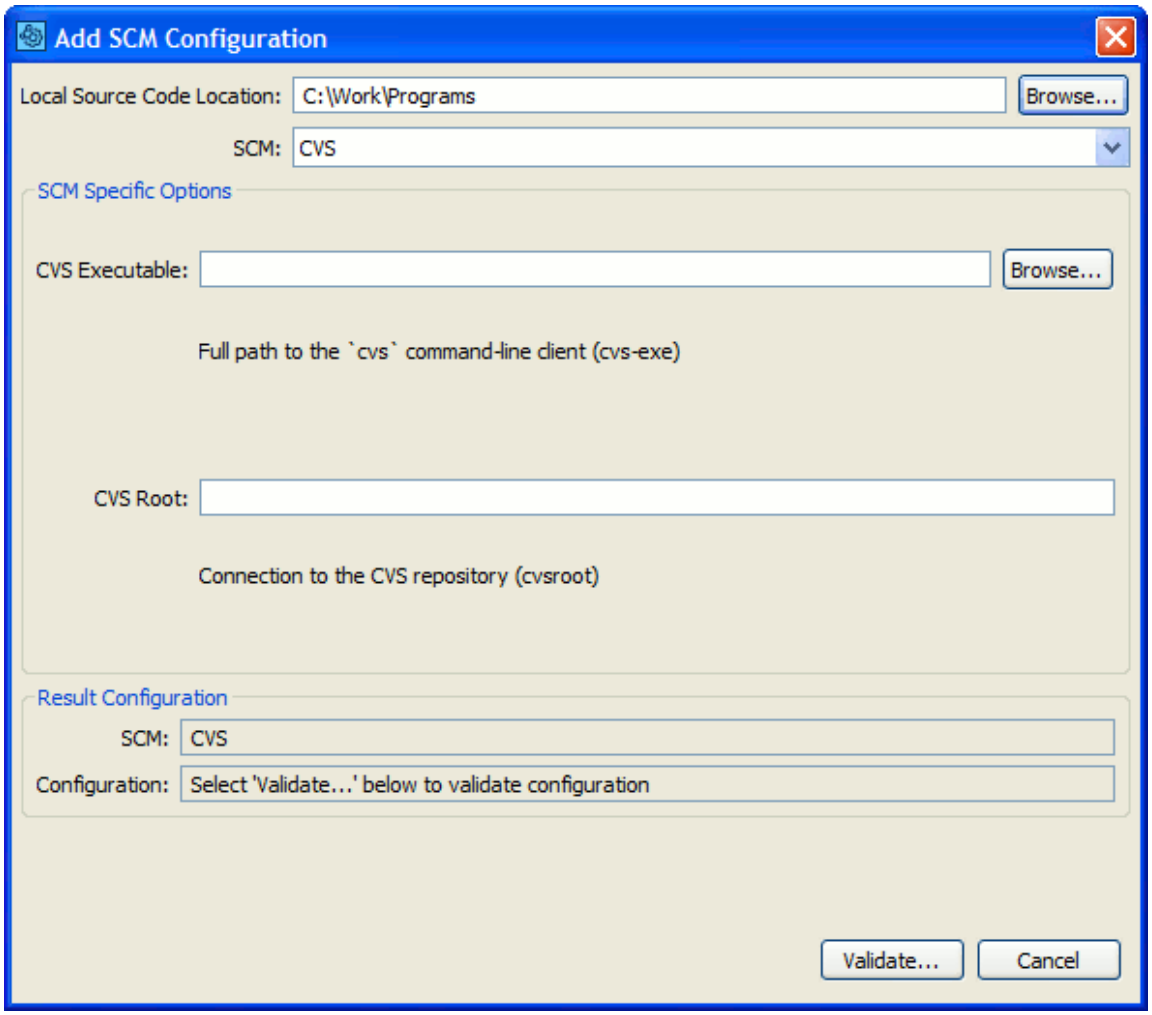
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you add several diffs (non atomic changelists) to the same review.

## 6.2.1 GUI Client

### CVS-specific Options

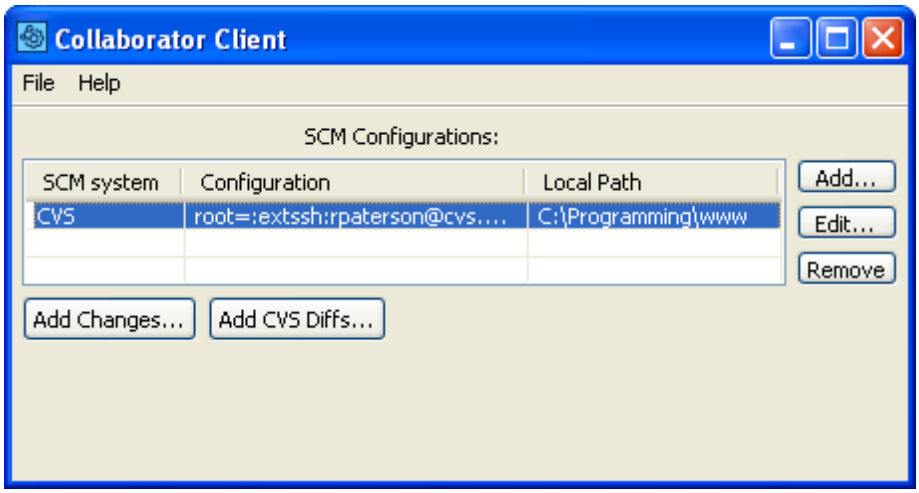
The SCM Configuration dialog[498] has several CVS-specific options.




*CVS SCM Configuration*

### Uploading files to a Review

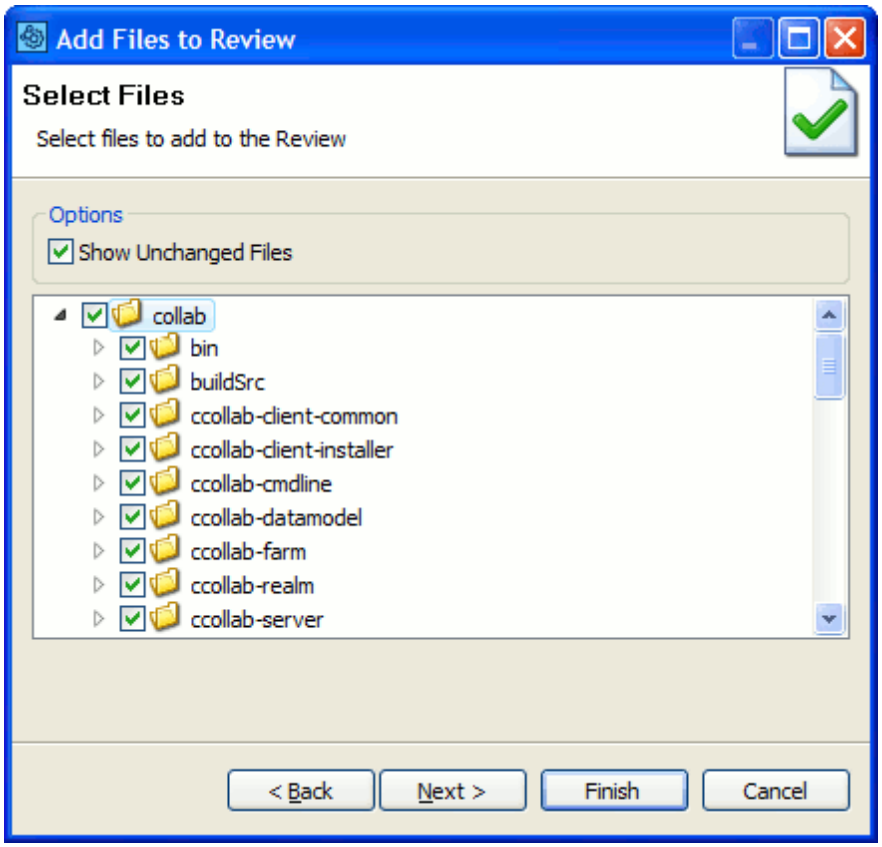
Selecting a CVS SCM Configuration in the GUI Client main screen[496] causes several Add to Review buttons to appear. The *Add Changes...*[637] button uploads modified files in a CVS checkout. The *Add CVS Diffs...*[638] button uploads arbitrary diffs, or compares the difference between two Labels[638] or dates[639].




*Uploading CVS files to a Review*

## Add Changes

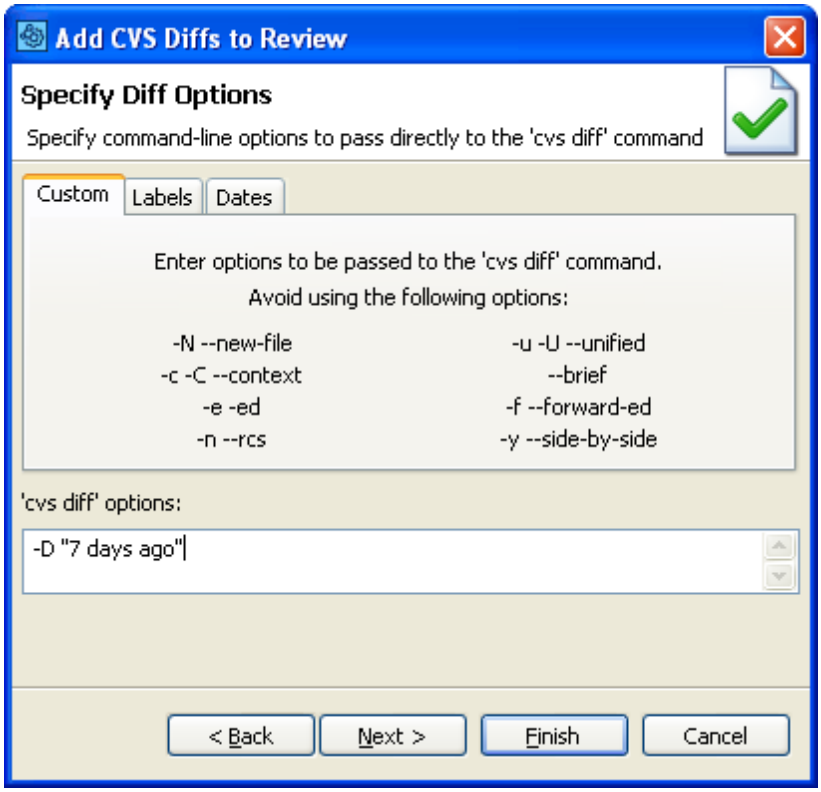
Press the *Add Changes...* button to upload modified files in a CVS checkout to the Collaborator Server for review.




*Add Changes*

## Add CVS Diffs

Press the *Add CVS Diffs...* button to upload arbitrary CVS diffs to the Collaborator Server for review.

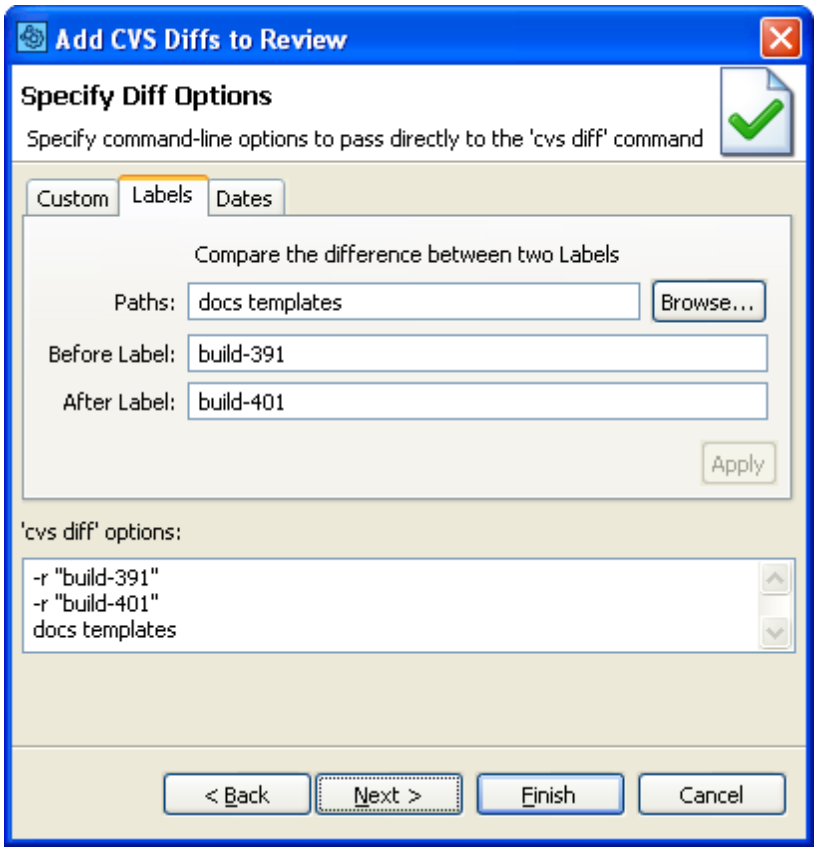



*Add CVS Diffs*

You can enter arbitrary CVS diff options, or compare the difference between two Labels 638 or dates 639.

### 6.2.1.1 Comparing two Labels

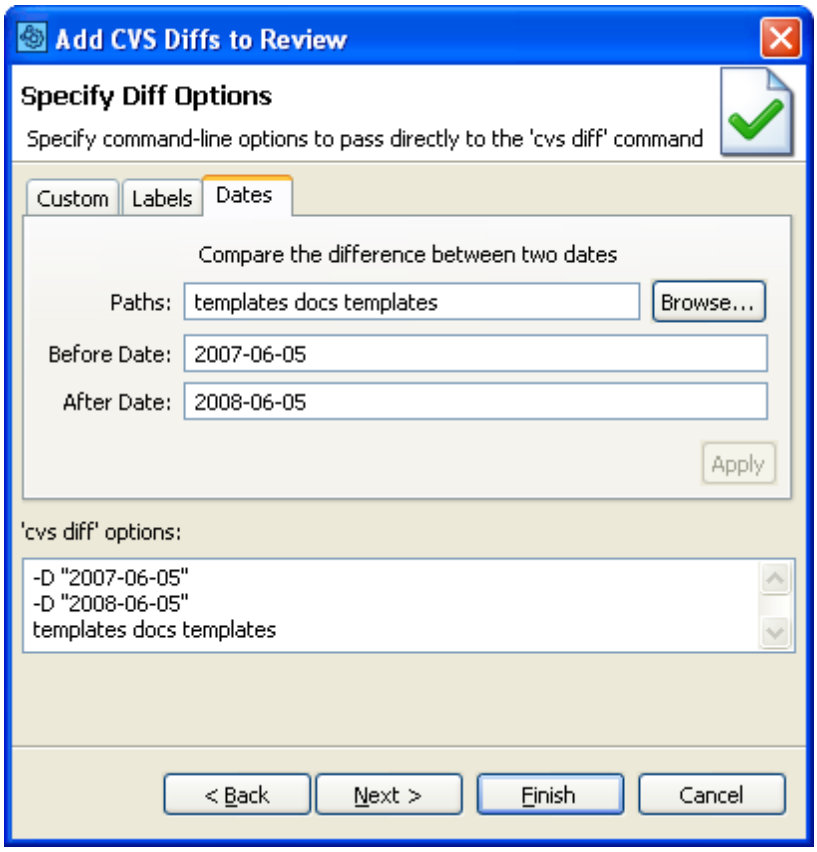
Press the *Add CVS Diffs...* 638 button on the main screen 496 and then select the *Labels* tab to upload the difference between two Labels.




*Upload the difference between two Labels*

### 6.2.1.2 Comparing two dates

Press the *Add CVS Diffs...*[638] button on the main screen[496] and then select the *Dates* tab to upload the difference between two dates.




*Upload the difference between two dates*

### 6.2.2 Command-Line Client

## Commands recommended for CVS

`ccollab addchanges` 641 - Attaches locally-modified files to a review

`ccollab addcvsdiffs` 643 - Uploads diffs generated from cvs diff command

`ccollab commit` 644 - Commit changes in the review

The addchanges 641 command will upload local changes to files controlled by CVS before they are checked into version control.

## Configuration

In most cases, the Command-Line Client can automatically detect your CVS configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your CVS configuration or you want to override the detected settings, you can manually specify CVS settings using global options[518].

To manually configure the Command-Line Client to use CVS, execute the following command:

```
ccollab set scm[517] cvs
```

## CVS-specific Options

| Option | Description |
|---|---|
| `--cvs-exe <value>` | Full path to the `cvs` command-line client |
| `--cvsroot <value>` | Connection to the CVS repository |

#### 6.2.2.1 addchanges (for CVS)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by CVS to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference[516]. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour: |

| Option | Required? | Description |
|---|---|---|
| | | • `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br><br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

#### 6.2.2.2 addcvsdiffs

## Description

The `ccollab addcvsdiffs` command uploads differences between arbitrary versions of files in CVS. The differences are generated using the native 'cvs diff' command of CVS.

## Command Line Syntax:

```
ccollab [global-options] addcvsdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files (defaults to command-line arguments) |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `-skipN` | No | Specifies whether to skip the `-N --new-file` argument of the cvs diff command.<br>By default, the diffs generated by Collaborator contain information about new files. Use -skipN argument to generate diffs for already committed files. |

| Option | Required? | Description |
|---|---|---|
| <user-diff-arg> [<user-diff-arg> ...] | No | Command-line arguments to pass directly to the diff command |

## Remarks:

Do not use diff arguments that affect the diff output such as '-u -U --unified', '-c -C --context', '--brief', '-e -ed', '-f --forward-ed', '-n --rcs', or '-y --side-by-side'. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.

## Examples:

To upload the changes between labels build-391 and build-401:

```
ccollab addcvsdiffs review -skipN -r build-391 -r build-401
```

To upload the changes between dates 2006-01-01 and 2006-02-01:

```
ccollab addcvsdiffs review -skipN -D 2006-01-01 -D 2006-02-01
```

To upload the changes in the last 7 days:

```
ccollab addcvsdiffs review -skipN -D "7 days ago"
```

### 6.2.2.3 commit (for CVS)

## Description

The `ccollab commit` command submits the changes from a pre-commit review to source control. Be sure to include a relevant comment.

## Command Line Syntax:

```
ccollab [global-options] commit [--comment <value>] [--dismiss-only]
[--force] <review>
```

### Command Options

| Option | Required? | Description |
|---|---|---|
| `--comment <value>` | No | Comment for reviewed changes |
| `--dismiss-only` | No | Just dismiss the Action Item |
| `--force` | No | Ignore potential problems |
| `<review>` | Yes | Identifier of the desired review (an integer number), or an `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

### Example:

```
ccollab commit 25 --comment "my code" --force
```

## 6.3 Git Integration

This section describes Collaborator integration with Git:

### Git Server Integration 647

The Collaborator server can pull commits directly from your Git repository for review, without users needing to install any client programs.

### GUI Client 648

The GUI Client can upload arbitrary Git diffs and provides a special user interface for specifying branches as diff arguments. The GUI Client can also upload local changes to files that have been added to the Git index.

## Command-Line Client 656

The Command-Line Client can upload arbitrary Git diffs and changes to files that have been added to the Git index.

### Supported Versions

Our integration uses your own installed Git command line client executable to generate differences for review. We require Git v1.7.4 or later.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

NOTE: If you intend to add unpushed commits, you must have an upstream tracking branch set. You can tell Git and Collaborator which upstream branch to compare against by running the command:

```
git branch --set-upstream [localbranch] [upstreambranch]
```

### Submodule Support

Currently Collaborator clients ignore submodule changes when adding changelists or diffs from the super project. Unlike clients, server side repository hosting integrations 826 can display changes in the submodules of tracked repositories. So, to add the submodule differences you must either use repository hosting integration, or navigate into the submodule itself and add the files from there.

### Technical Details and Limitations

Adding Git commits from GUI Client, Command-Line Client, Eclipse Plug-in or Visual Studio Extension will also add part of Git log information to build properly ordered list of changes on Collaborator server side. This will ensure that in Review Screen and Diff Viewer Git commits are displayed in the same order as in the Git log. To use this functionality you will need to update both server and clients to version 11.5 and later.

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists (commits in terms of Git) to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- If you add changelists from different branches or repositories to the same review.
- If you add several diffs (non atomic changelists) to the same review.

### 6.3.1 Git Server Integration

The Collaborator server can be configured to communicate directly with a Git repository. This allows users to review commits completely from the browser, without having to install any client programs. To enable this feature, first install and configure a Git client on the Collaborator server and then create an entry for your Git repository in the Version Control 204 tab of the administration interface. Version control server entries are also created automatically if one of the client programs uploads files from a repository that does not match any of the currently configured servers.


| | |
|---|---|
| **Title** | The title is displayed to users, so it should be something that everyone will understand, even if they are going through proxies, VPNs, or other such things. When a version control server entry is created automatically, this is filled in with "Git". |
| **Attach changelists from browser** | If enabled, this feature lets users select commits to review directly from the web browser, without having to install any client programs. |
| **Git Executable** | Full path to the 'git' command line client |
| **Git Repository** | Full path to a Git repository. This is only necessary if you enable "Attach changelists from browser". The repository can be "bare". |

Click the ***Test Connection*** to make sure Collaborator can contact your Git repository.

## Client Configuration Mapping

You can supply Java-style regular expressions to map changelists from this Git repository uploaded from our client tools. It is important to set up these regular expressions so that files uploaded by the various Collaborator client tools are correctly associated with this server-side Git configuration.

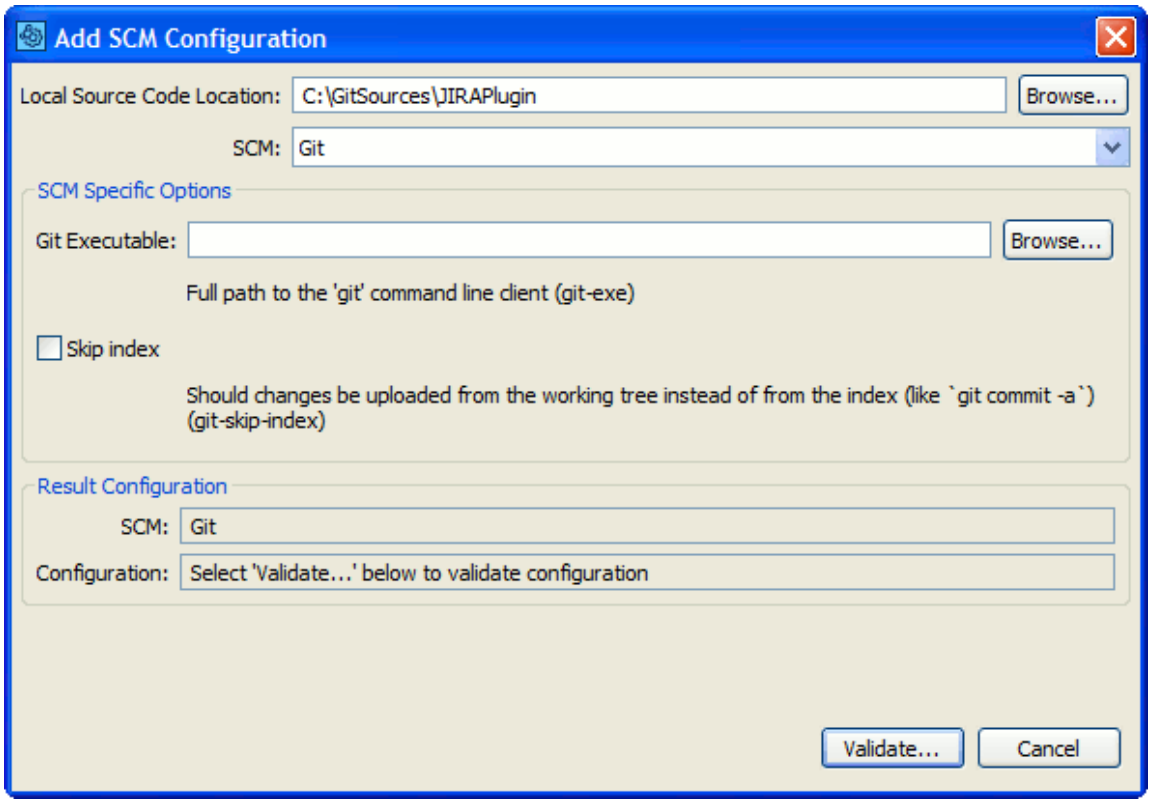

| | |
|---|---|
| **Repository Host Pattern** | Match on the client's configured remote origin URL (remote.origin.url). Note clients may have various network configurations that make the URL look different. |
| **First Commit Hash Pattern** | Match on the first commit hash returned from running "`git rev-list --max-parents=0 HEAD`". |

## 6.3.2 GUI Client

### Git Configuration in GUI Client

1. Start GUI Client.
2. Click **Add** to create a new **SCM configuration**.
3. In the subsequent SCM Configuration dialog 498 dialog, specify the local source code location. For Git, this is not optional.
4. Select *Git* in the **SCM** drop-down list.
5. Several options will become available. These are specifically tailored to configure Git integration.

| Option | Description |
|---|---|
| **Git Executable** | A full path referring the Git command line client (git.exe). |
| **Skip index** | Defines whether the changes you upload via the "Add Changes" command should be taken from the working tree (including not-yet-committed changes) instead of from the index. This is an analogue of the `git commit -a` command. |



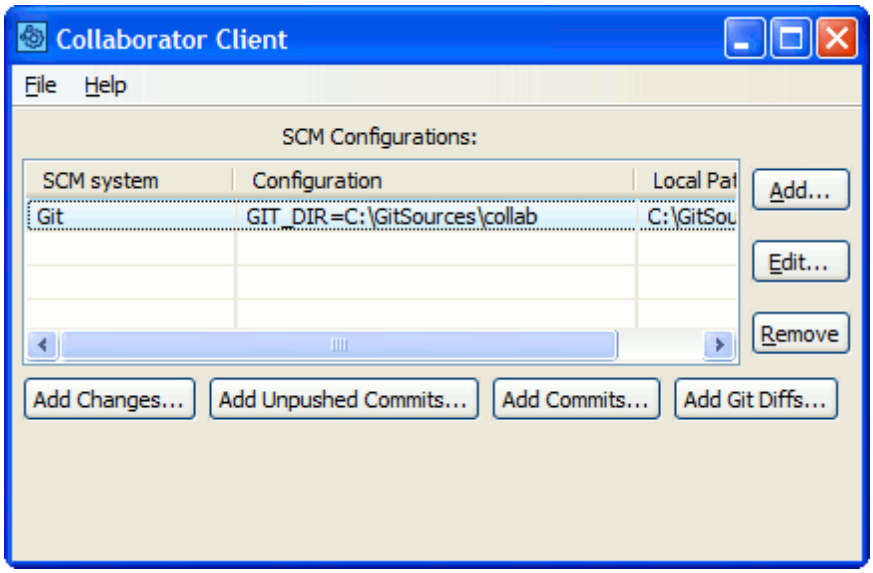

*Configure client to work with Git*

6. Once ready, click **Validate** to make sure the integration is operational.

After that, a new configuration for Git will appear on the main screen of the GUI Client.

## Uploading files to a Review

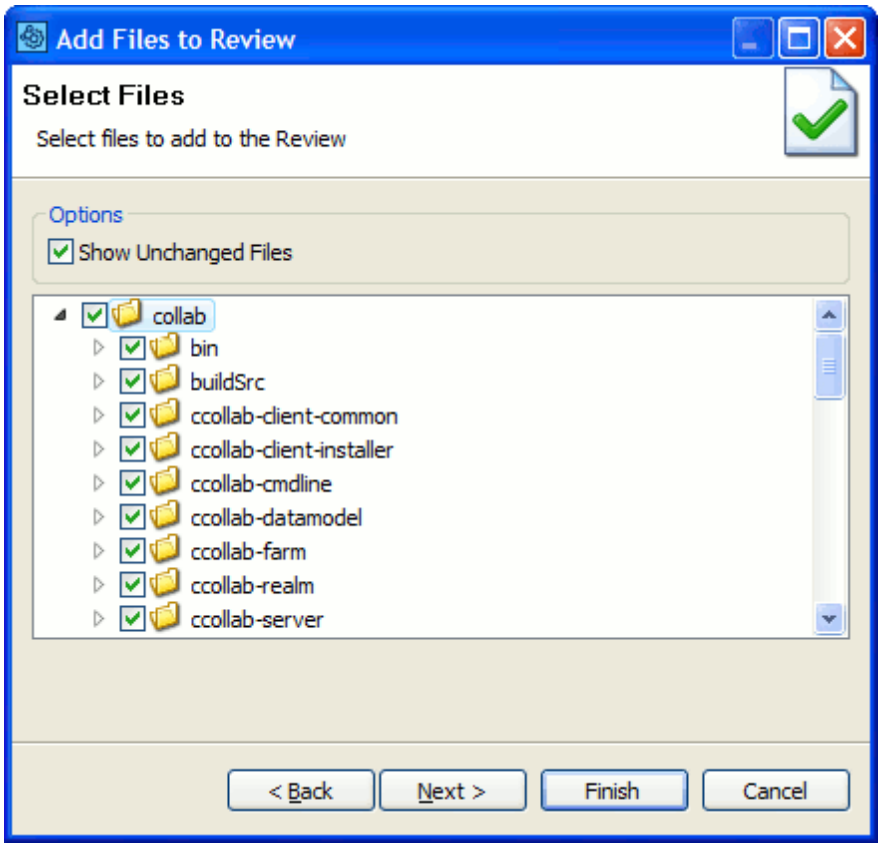
Selecting a Git SCM Configuration in the GUI Client main screen[496] causes *Add to Review buttons* to appear.




*Uploading Git files to a review*

## Add Changes

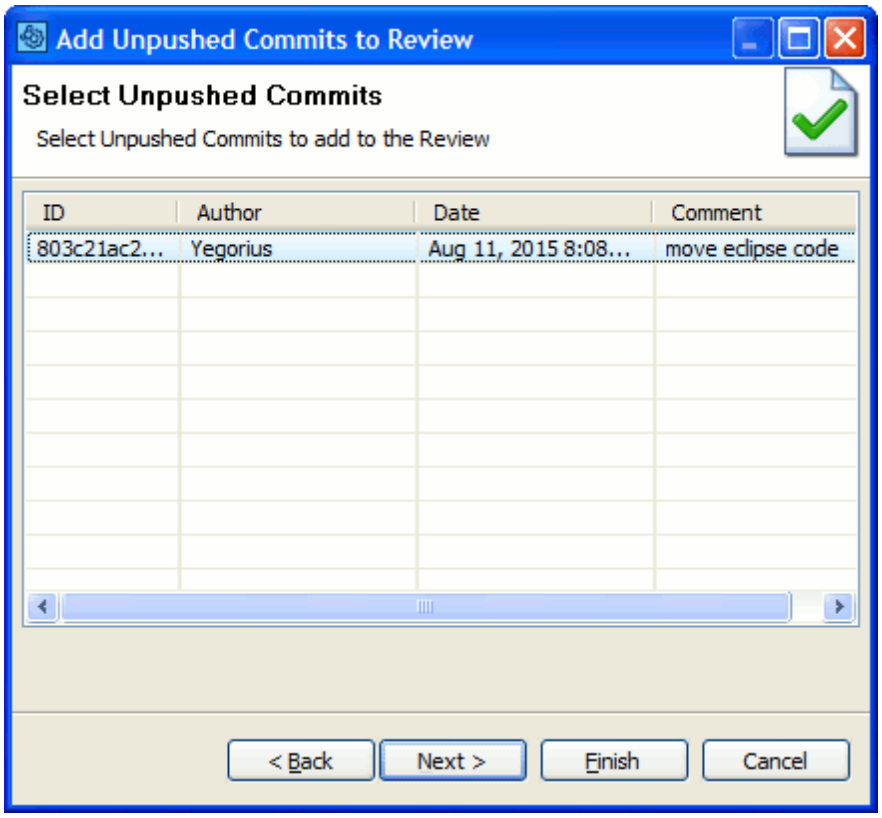
Selecting *Add Changes...* allows you to upload the modifications that are currently in the index. These are the modifications that would be committed if you typed 'git commit' from a command line.




*Add Changes*

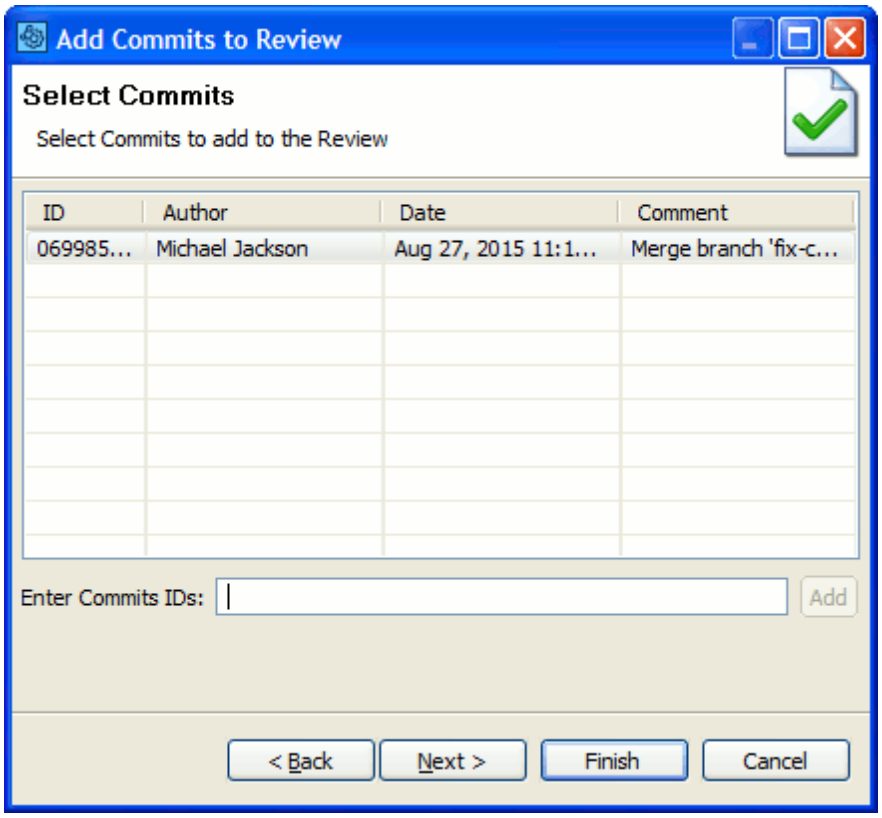
## Add Unpushed Commits

Selecting *Add Unpushed Commits...* selects all commits in your local branch that have not been pushed to its tracking branch. NOTE: This assumes that you have set up branch tracking in Git. If you see an error when running *Add Unpushed Commits...* (like, "Error initializing local changelists") make sure that your current branch has a tracking branch set. You can set this up, initially by running "git config branch.autosetupmerge always". You can set this up on an existing branch by running "git branch --set-upstream name-of-branch name-of-upstream".



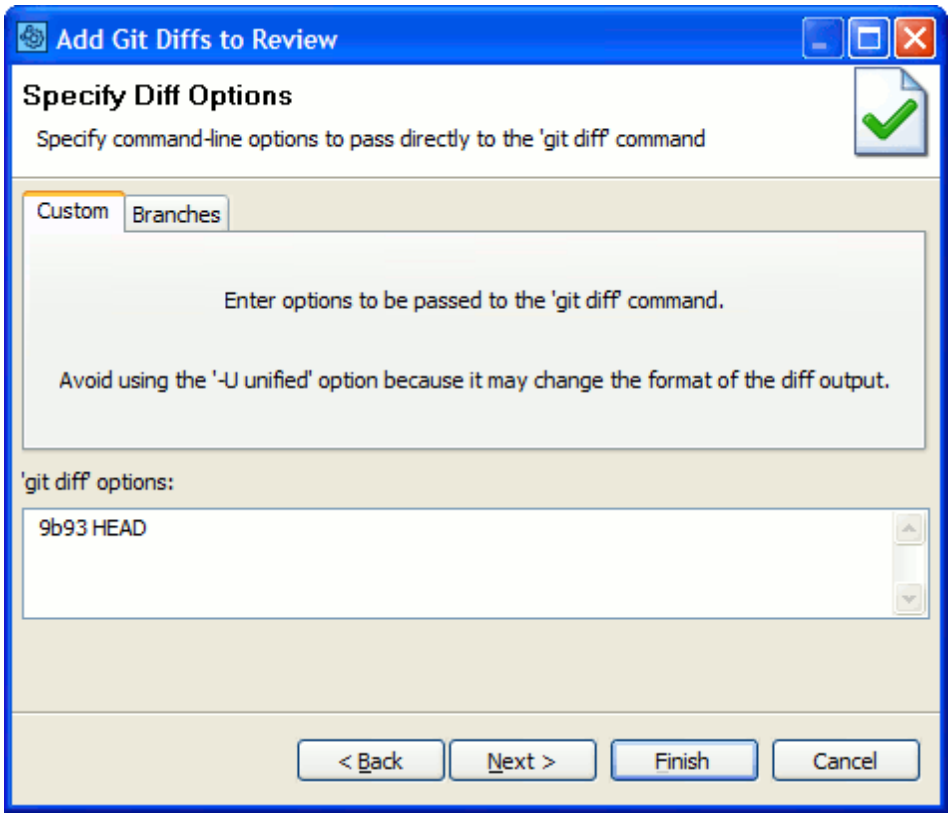

## Add Commits

Selecting *Add Commits...* allows you to upload commits, whether they have been pushed or not. You can add a specific commit by adding the commit ID and clicking "Add":



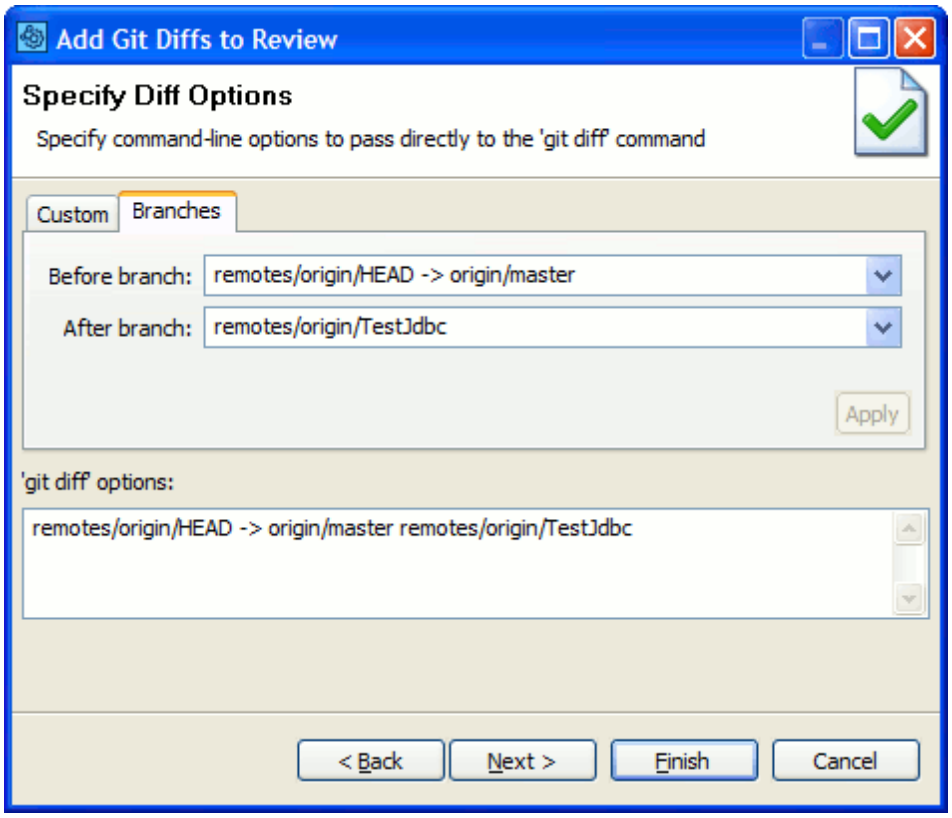

## Add Git Diffs

Press the *Add Git Diffs...* button to upload arbitrary Git diffs to the Collaborator Server for review.




*Upload arbitrary Git diffs*

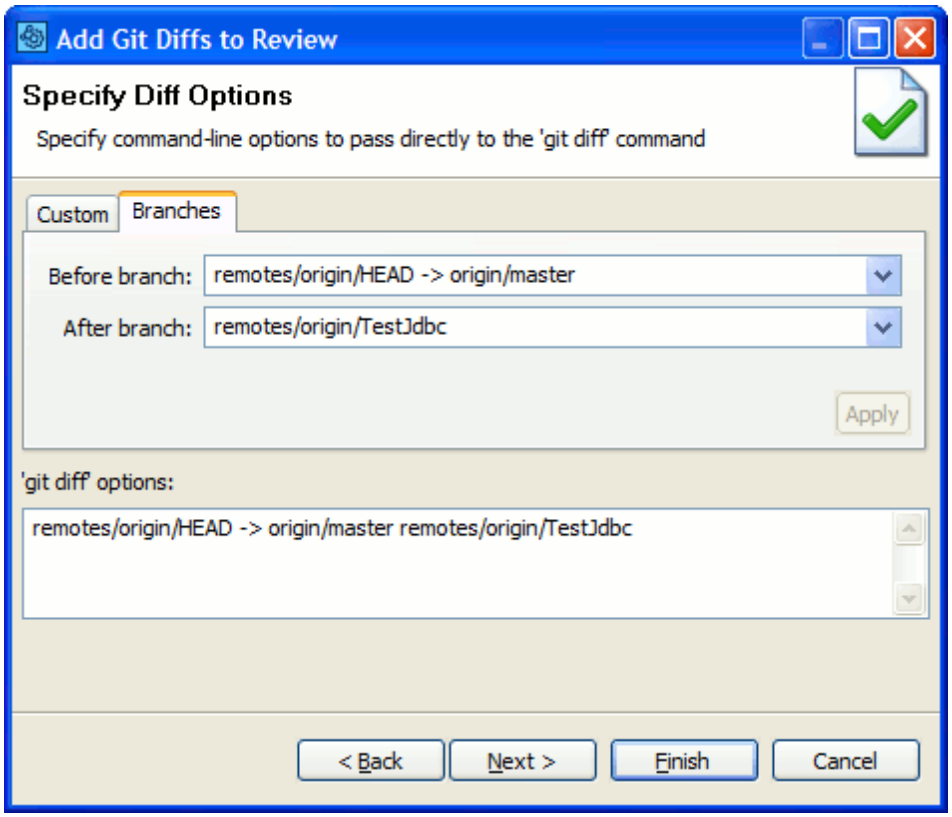
Press the *Add Git Diffs... button* and then click on the *Branches* tab to upload the differences between two branches to a review. Specify the branches by selecting from the *before* and *after* dropdown menus.




*Upload the difference between two branches*

#### 6.3.2.1 Comparing two branches

Press the *Add Git Diffs...* 648 button on the main screen 496 and then select the *Branches* tab to upload the differences between two branches to a review. Specify the branches by selecting from the *before* and *after* dropdown menus.




*Upload the difference between two branches*

### 6.3.3 Command-Line Client

## Commands recommended for Git

`ccollab addchanges` 657 - Attaches locally-modified files to a review.

`ccollab addchangelist` 659 - Attaches an atomic changelist to a review.

`ccollab addgitdiffs` 660 - Uploads differences generated by git diff command.

`ccollab gitaddbranch` 661 - Uploads all differences between the specified branch and the remote-tracking branch.

## Configuration

In most cases, the Command-Line Client can automatically detect your Git configuration. Try testing your configuration[507] to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Git configuration or you want to override the detected settings, you can manually specify Git settings using global options[517].

To manually configure the Command-Line Client to use Git, execute the following command:

```
ccollab set scm[517] git
```

## Git-specific Options

| Option | Description |
|---|---|
| `--git-exe <value>` | Full path to the 'git' command line client |
| `--git-skip-index` | Defines whether the changes you upload via the `ccollab addchanges` command should be taken from the working tree (including not-yet-committed changes) instead of from the index. This is an analogue of the `git commit -a` command. |

#### 6.3.3.1 addchanges (for Git)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by Git to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br><br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

### 6.3.3.2 addchangelist (for Git)

## Description

The `ccollab addchangelist` command attaches all files from an unpushed or committed Git changelist to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or Git-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review, |

| Option | Required? | Description |
|---|---|---|
| | | • `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ... ]` | Yes | Identifier(s) of the desired changeset(s) in your source control. |

## Examples:

To upload Unpushed Commits 4321 and 7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

To upload Commits 5432 and 12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

### 6.3.3.3 addgitdiffs

## Description

The `ccollab addgitdiffs` command uploads differences between arbitrary versions of files in Git. The differences are generated using the native 'git diff' command of Git.

## Command Line Syntax:

```
ccollab [global-options] addgitdiffs [--upload-comment <value>]
<review> [<git-diff-arg> [<git-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files (defaults to command-line arguments) |

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<git-diff-arg> [<git-diff-arg> ...]` | No | Options which should be passed to the git diff command |

## Remarks:

Do not use diff arguments that affect the diff output such as '-U unified'. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.

## Examples:

To upload all changes between the revision 8 revisions ago and the revision 4 revisions ago:

```
ccollab addgitdiffs review master~8 master~4
```

To upload all changes in your local working directory:

```
ccollab addgitdiffs review
```

#### 6.3.3.4 gitaddbranch

## Description

The `ccollab gitaddbranch` command uploads all differences between the given branch and the remote branch being tracked for changes.

## Command Line Syntax:

```
ccollab [global-options] gitaddbranch <review> [<branch>]
[<upstream>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| <review> | Yes | Identifier of the desired review (an integer number), or a new, ask, or last keyword. Where keywords define the following behaviour:<br>• new - the command will create a new review,<br>• ask - the command will pause execution and prompt for the identifier of the desired review,<br>• last - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <branch> | No | Name of a branch whose changes should be added. Default is the current checkout branch. |
| <upstream> | No | Name of the remote-tracking branch to be compared against. If omitted Collaborator will try the default upstream branch (which was set via the git branch --set-upstream-to command-line key). |

## Examples:

To upload all differences between the "foo_feature" branch and the "origin/main" repository:

```
ccollab gitaddbranch new foo_feature origin/main
```

### 6.3.4 Git Server Hooks

The ensure-review-started 664 and ensure-reviewed 665 hooks ensure that files cannot be committed unless certain conditions are met. If a user attempts to commit files that do not meet those conditions, an error message describing the unfulfilled conditions will be displayed and the files will not be committed. If the conditions are met, the commit will be allowed to continue normally. The ensure-review-started hook requires that the review exist; ensure-reviewed requires that the review be completed.

If you are familiar with Gerrit (an open source Git-focused review package), you can implement a similar workflow for your engineering staff using Collaborator and Git triggers. Further information may be found on the SmartBear blog in the "Gerrit-Style Code Review with Collaborator" post.

## Linking reviews with commits

To use the ensure-review-started 664 and ensure-reviewed 665 hooks, you must require developers to put the review ID somewhere in the Git commit message. The format of this text is completely up to you; you will need to supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text. The regular expression is specified using the --review-id-regex hook command option.

Here are some common ways of specifying the review ID and the corresponding regular expressions. Note that regular expressions are *case-insensitive* and you must *identify the review ID portion with parenthesis*:

| Text | --review-id-regex |
|---|---|
| Review: 4233 | review:\s*(\d+) |
| rID4233 | rid(\d+) |
| (review 4233) | \(review (\d+)\) |

This text can appear in-line with other text or in a more formal "form-style" layout. Because you control the regular expression, you can control exactly what this looks like.

For more information about Git hooks in general, see the Git documentation.

#### 6.3.4.1 ensure-review-started (for Git)

## Description

Use the `ensure-review-started` trigger to ensure that a review for the specified changelist has been started. The trigger blocks the submit operation if there is no review for the changelist in Collaborator, and displays an error message telling that the changes need to be reviewed before submitting them to the Git repository.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-review-started [--
review-id-regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<changelist-id>` | Yes | The changelist identifier. |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install this trigger you will typically create an update hook. If you already have an update hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Git documentation. The update hook should iterate over the commits being pushed and call this trigger for each one of them, and then exit with a non-zero exit code if any of the trigger invocations failed.

Example shell script:

```
#!/bin/sh
refname="$1"
oldrev="$2"
newrev="$3"
for commit in `git rev-list $oldrev..$newrev -- ''`
do
```

```
 /collab/install/ccollab --url <collabUrl> --user <collabUser> --
password <collabPasswd> --scm git admin trigger ensure-review-started
$commit
 exitcode=$(($exitcode + $?))
done
exit $exitcode
```

Note our use of "exit" to ensure that the hook script terminates with a non-zero exit code if our trigger rejects one of the commits.

#### 6.3.4.2 ensure-reviewed (for Git)

## Description

Use the `ensure-reviewed` trigger to ensure that the review that was created for the specified changelist has been completed by the time you submit the changelist to the Git repository. If the review has not been completed, the trigger blocks the submit operation and displays an error message informing the user about the problem.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-reviewed [--review-id-
regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<changelist-id>` | Yes | The changelist identifier. |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install this trigger you will typically create an update hook. If you already have an update hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Git documentation. The update hook should iterate over the commits being pushed and call this trigger for each one of them, and then exit with a non-zero exit code if any of the trigger invocations failed.

Example shell script:

```
#!/bin/sh
refname="$1"
oldrev="$2"
newrev="$3"
for commit in `git rev-list $oldrev..$newrev -- ''`
do
 /collab/install/ccollab --url <collabUrl> --user <collabUser> --
password <collabPasswd> --scm git admin trigger ensure-reviewed
$commit
 exitcode=$(($exitcode + $?))
done
exit $exitcode
```

Note our use of "exit" to ensure that the hook script terminates with a non-zero exit code if our trigger rejects one of the commits.

## 6.4 IBM Rational ClearCase Integration

Topics of this section describe Collaborator integration with IBM Rational ClearCase.

### ClearCase Server Integration 668

To take advantage of ClearCase integration, configure the Collaborator server so that it communicates with a ClearCase repository directly, without any client software.

### GUI Client 670

The GUI Client can scan ClearCase for checked-out files and upload them before they are checked in 672. It uploads changes from your activities 673 or changes from an activity or activities that you name 674. It can also upload any versions of ClearCase files after they are checked in.

### Command-Line Client 679

The Command-Line Client can scan for checked-out files and upload them before they are checked in, upload any versions of ClearCase files after they are checked in, or upload UCM change sets for review.

## Eclipse Plug-in

The Eclipse Plug-in 525 integrates with the IBM Rational ClearCase SCM Adapter plug-in or with the open source ClearCase plug-in for Eclipse so you can upload 545 locally modified files with full support from your ClearCase eclipse plug-in.

## Supported Versions

Our integration uses your own ClearCase command-line client (`cleartool`) to communicate with the server. We support these versions:

- Rational ClearCase and Rational ClearCase LT, versions 4.1 through 9.x
- Rational ClearCase Remote Client (CCRC), version 9.x

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

## Support for ClearCase UCM

Collaborator supports ClearCase UCM activities. The change set associated with a ClearCase activity can be added to a review using the Command-Line Client 506 addactivity 688 command. For each file in the activity, the latest version in the activity and the predecessor of the earliest version in the activity are uploaded to the review.

## A Note about ClearCase Activities

Uploading activities for review generates file differences based on Multiple Version Changelists 970. Uploading activities is not supported by Rational ClearCase Remote Client.

## Support for ClearCase Remote Client (CCRC 9.x only)

Collaborator integration with CCRC uses the Rational Change Management Server API. When running the Collaborator client installer, there is an edit field allowing you to select the location of the Rational CM API jar files on your system. The required files are:

```
remote_core.jar

stpcc.jar

stpcmmn.jar

stpwvcm.jar
```

These files are installed by the ClearCase Remote Client under the Rational Shared Resource Directory. The directory should look similar to one of the following:

```
C:\Program Files\IBM\IMShared\plugins\com.ibm.rational.teamapi_9.0.0.
vNNNNNNNN, or
/opt/eclipse/plugins/com.ibm.rational.teamapi_9.0.0.vNNNNNNNN
```

Launch Collaborator client installer with administrator privileges, find the location of the Rational CM API jar files on your system and specify it in the "CM API Directory" field during installation. The installer will copy the abovementioned jar files to the following subdirectory under the Collaborator client installation directory:

```
<Collaborator Client>/ui/plugins/com.smartbear.collaborator_13.6.xxxx
```

After that, the Collaborator client would become able to use the Rational Change Management Server API.

If using the RPM installer, you will have to manually copy these jar files to the correct subdirectory.

Once installed, create a ClearCase SCM configuration using the the same connection parameters to the ClearCase CM server as your ClearCase Remote Client. These are your username and password, and the ClearCase TeamWeb services URL. For example, if your ClearCase server host name is cc-server, the default server URL would be:

```
http://cc-server:16080/ccrc
```

Be careful to specify your connection parameters correctly. These parameters are not authenticated up front, instead calling back when server activity must take place. If the authentication is wrong, a cascade of errors can occur which can exhaust the RPC connections to the CM server.

## Troubleshooting CCRC Views

ClearCase Remote Client views can get corrupted in a way that causes the CM API to generate exceptions when trying to upload CCRC files to a review. We do not know the cause of these exceptions, but selecting "Refresh->Repair->Vob Tag Directories" from the view context menu in the CCRC client, or deleting and recreating the view, fixes this condition.

## Technical Details and Limitations

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you add several diffs (non atomic changelists) to the same review.

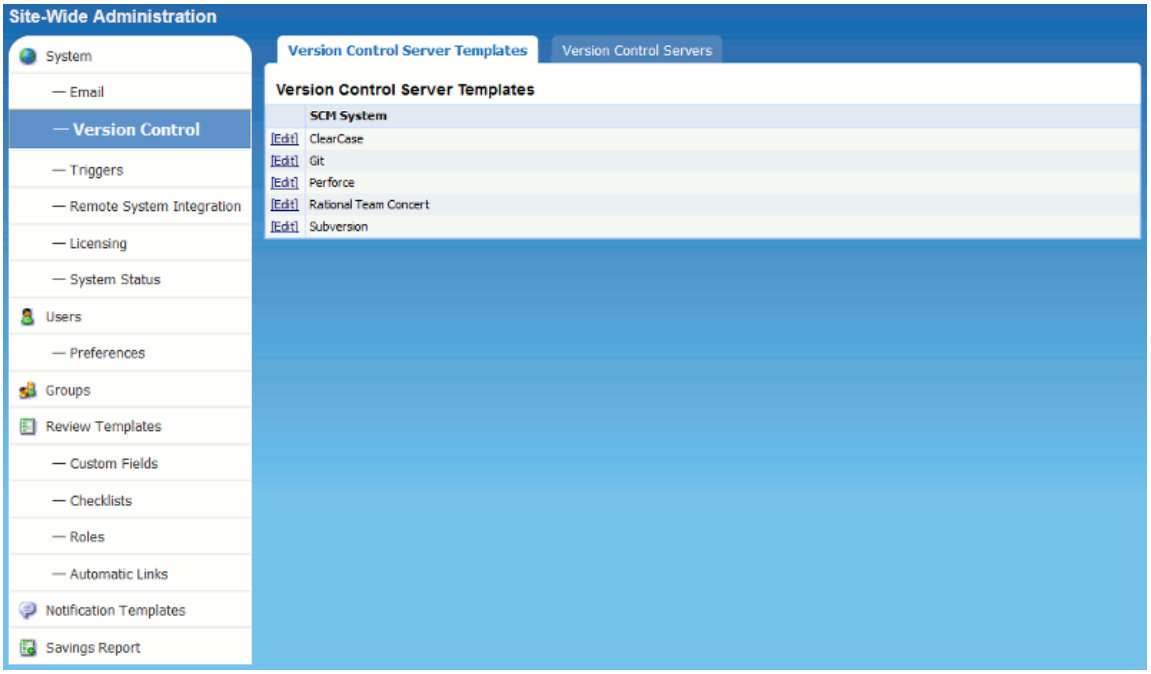
### 6.4.1 ClearCase Server Integration

The Collaborator server can be configured to communicate directly with a ClearCase repository. This allows users to review commits completely from the browser, without having to install any client programs.

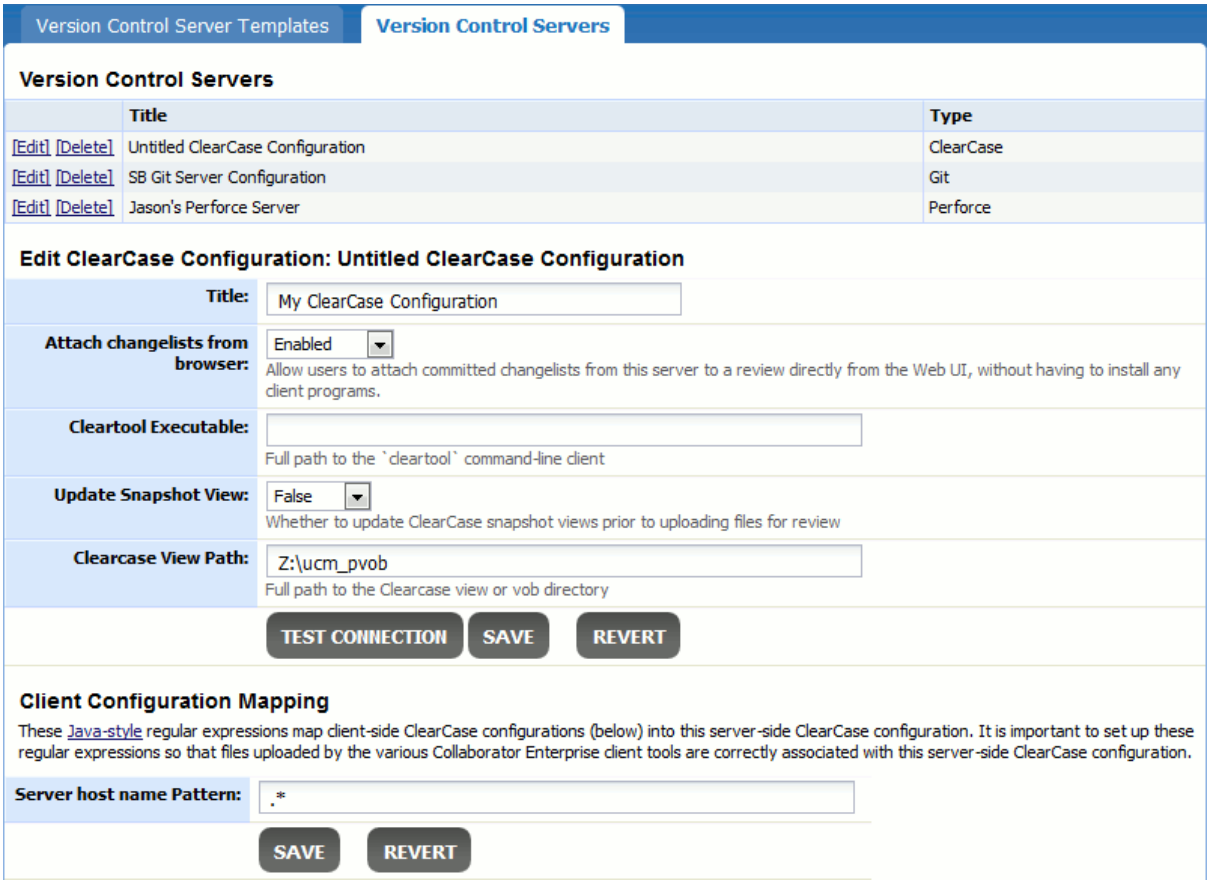
To enable this feature:

1. Install and configure a ClearCase client on the Collaborator server.

2. In Collaborator, click **Admin** on the home page and then select **Version Control** from the menu on the left:




3. Add new ClearCase configuration repository to Collaborator. To do this, switch to **Version Control Servers** tab, in the **SCM System** drop-down list select "ClearCase" and click **Create**. The following edit form will appear. Enter your repository details there:




4. Save the changes.

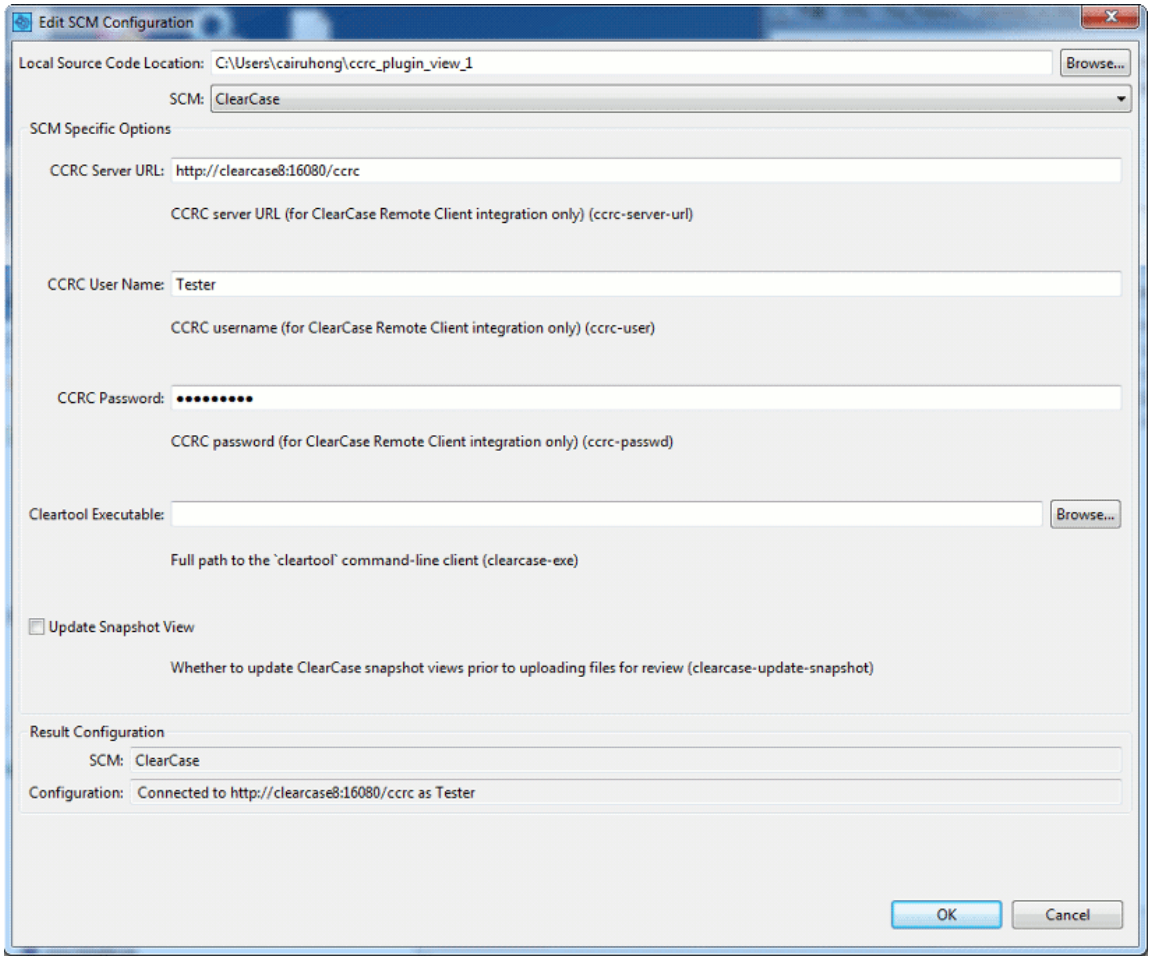
## 6.4.2 GUI Client

### ClearCase-specific Options

The SCM Configuration dialog 498 has several options tailored to configure ClearCase integration.

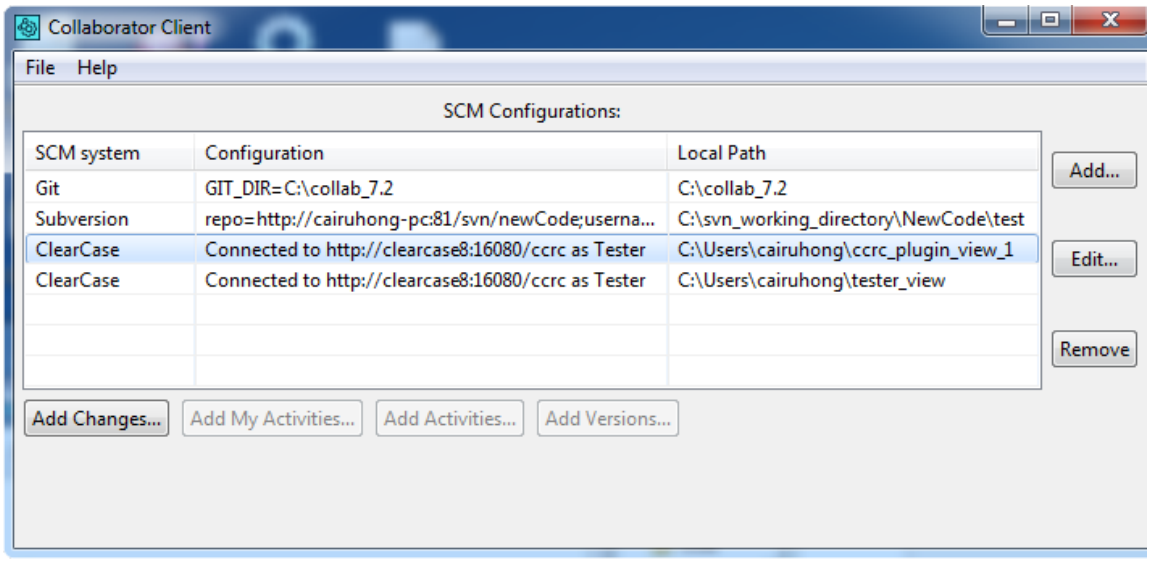
| Option | Description |
|---|---|
| **CCRC Server URL** | An URL of the ClearCase server. Required for remote connections. |
| **CCRC User Name** | A user-name you want to use to authorize on the ClearCase server. Required for remote connections. |
| **CCRC Password** | A password you want to use to authorize on the ClearCase server. Required for remote connections. |

| | |
|---|---|
| **Cleartool Executable** | A full path to the **cleartool** command line client (`clearcase.exe`). |
| **Update Snapshot View** | If selected, updates ClearCase snapshot views before uploading files to a Collaborator review. |




## Uploading files to a Review

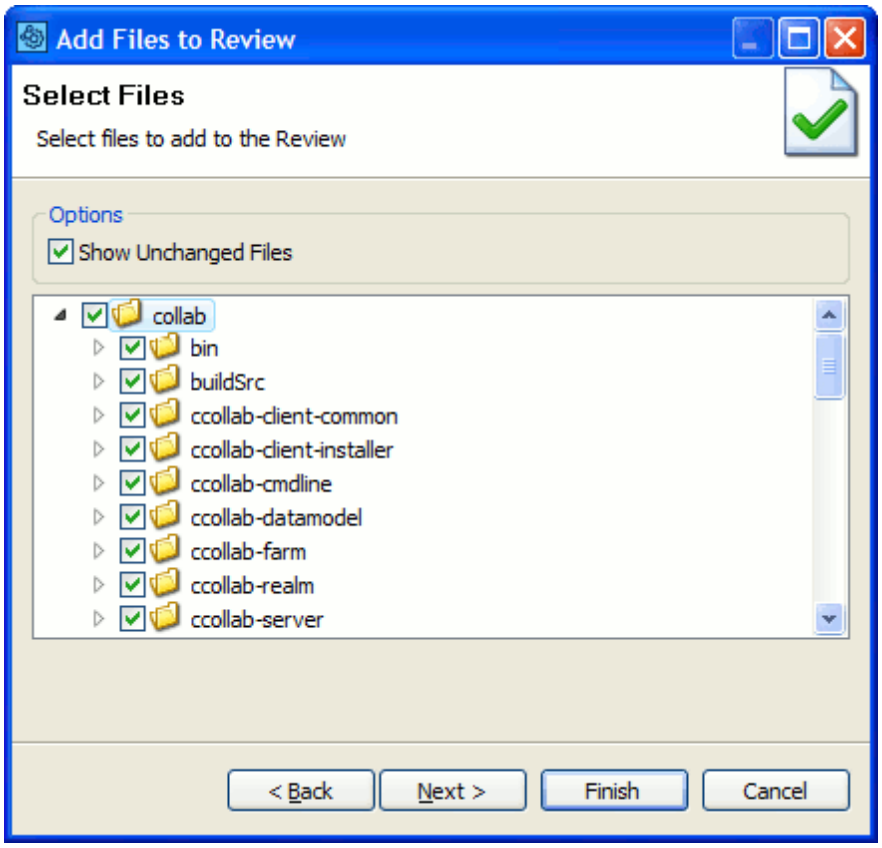
Selecting a ClearCase SCM Configuration in the GUI Client main screen[496] enables several options for adding files for review. The *Add Changes...*[672] button finds modified files or allows selection of unmodified files, for uploading to a review. The *Add My Activities...*[673] button uploads changes from your activities. The *Add Activities...*[674] button uploads changes from an activity or activities that you name. The *Add Versions...*[674] button uploads arbitrary versions of files stored in ClearCase.




*Uploading ClearCase files to a Review*

## Add Changes

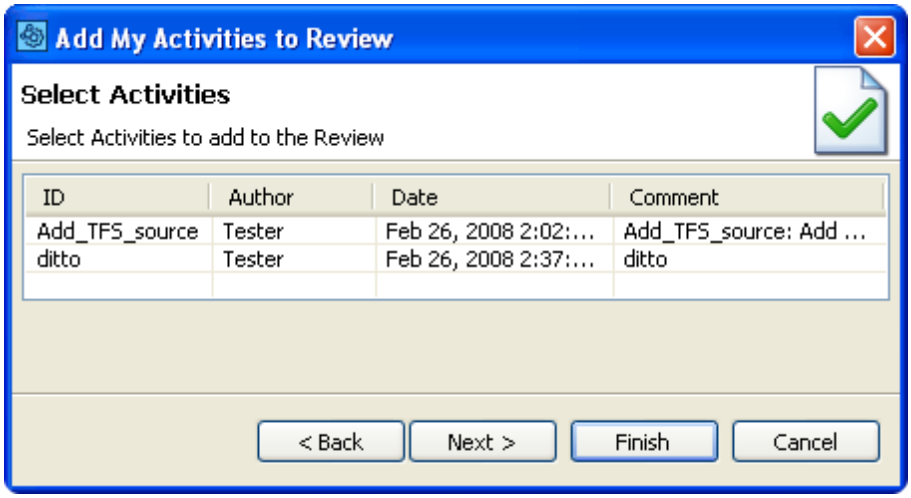
Press the *Add Changes...* button to upload checked-out files to the Collaborator Server for review.




*Add Changes*

## Add My Activities

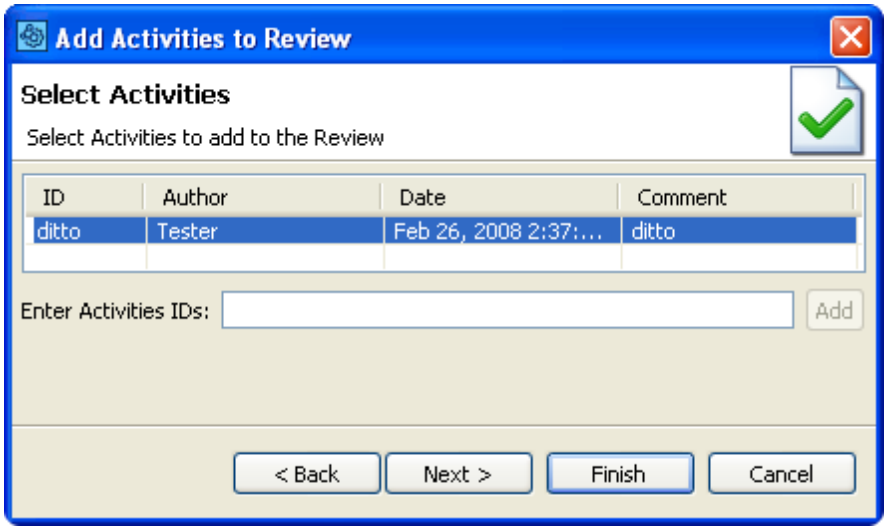
Press the *Add My Activities...* button to upload files from selected activities which you own to the Collaborator Server for review.




*Add My Activities*

## Add Named Activities

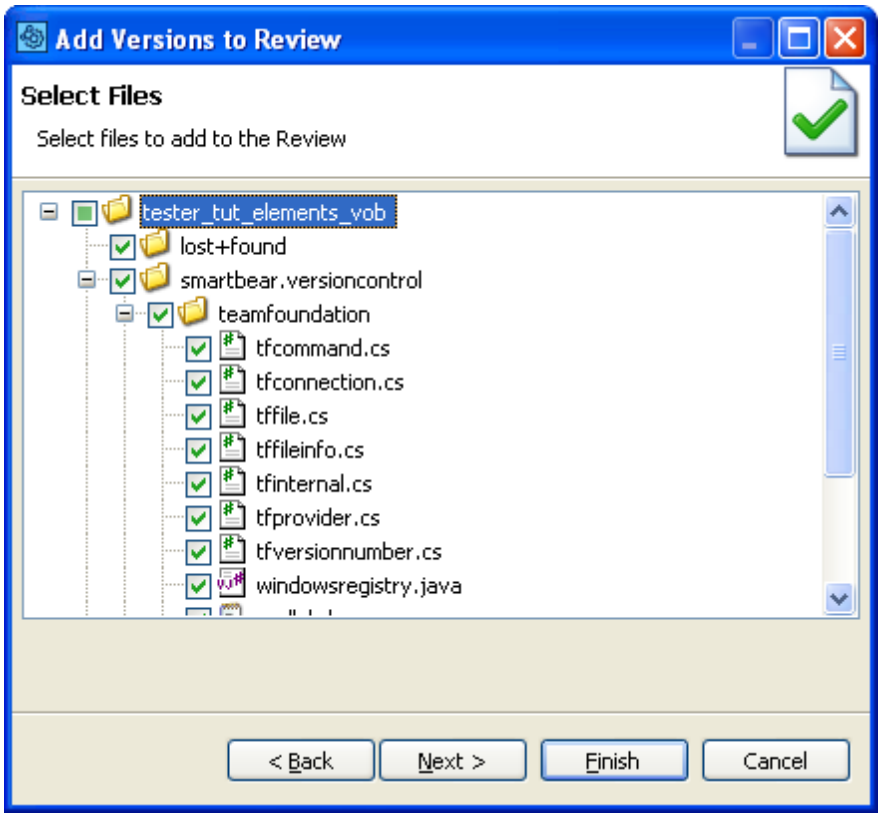
Press the *Add Activities...* button to upload changes from a named ClearCase activity to the Collaborator Server for review. You cannot upload activities that have CHECKEDOUT versions if you are not the activity owner.




*Add Named Activities*

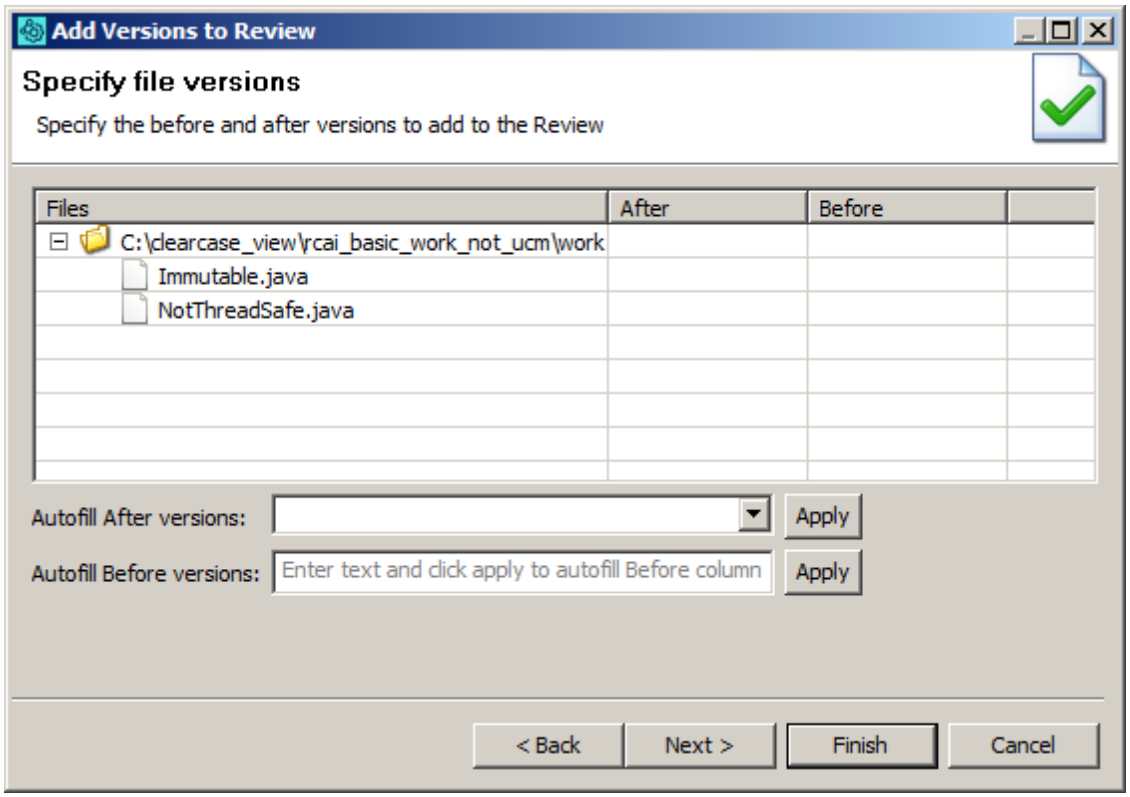
## Add Versions

Press the *Add Versions...* button to upload versions of files in your working copy to the Collaborator Server for review. Specify a name for the review or select an existing review and then click Next. You will be prompted to select one or more files in your checkout path (which was specified in the SCM Configuration dialog) 498.




*Select files*

Click Next. Do not press Finish yet.

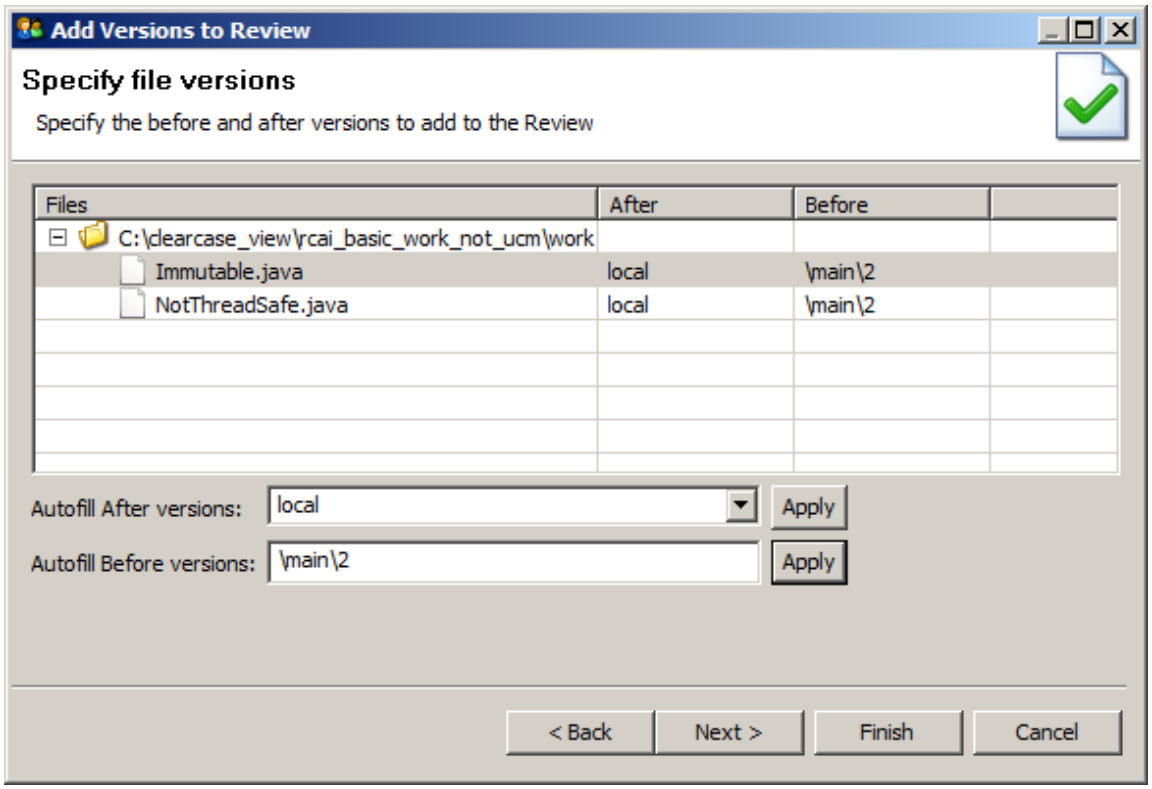



*Specify versions*

The next page of the wizard lists the files that you have selected. You need to specify which versions of these files should be added to the review.

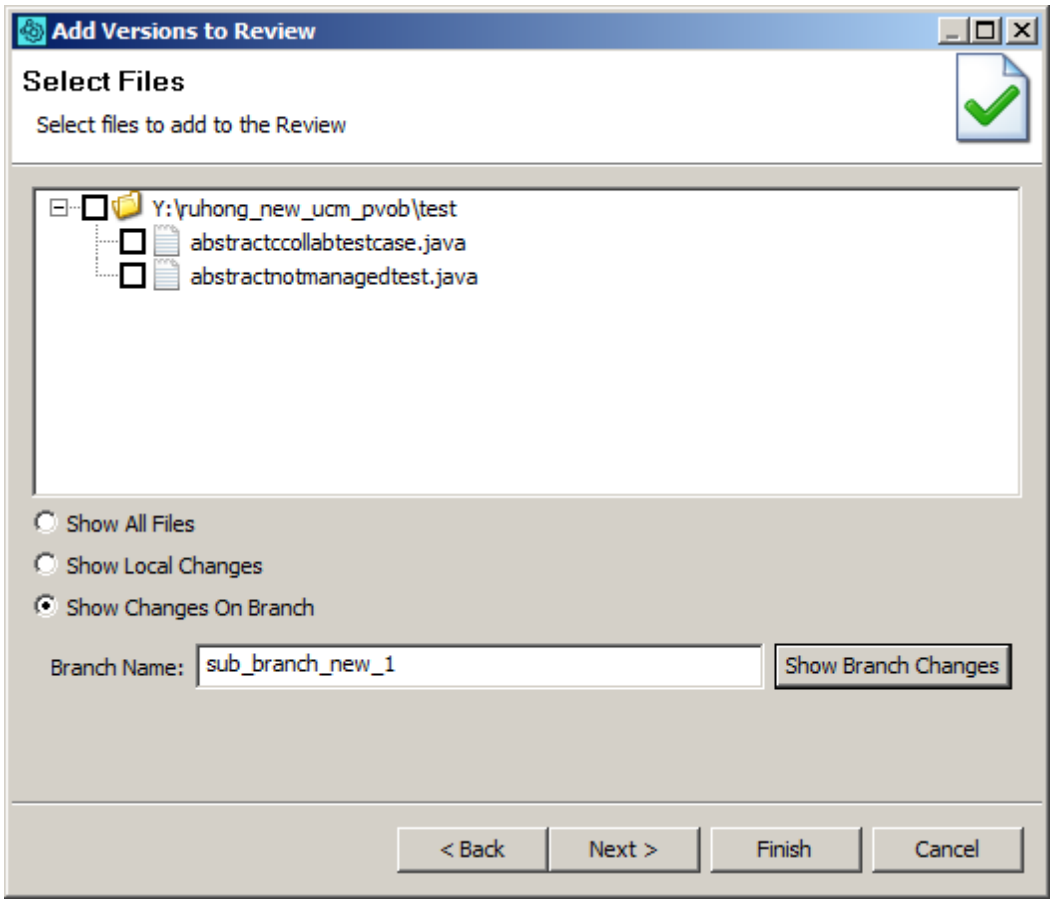
Enter the ClearCase versions of each file to Before and After columns. Rows corresponding to directories are not editable. To specify a ClearCase version you can use the 'local' and 'LATEST' keywords. To learn about these keywords, see ccollab addversions 684 command-line reference.

To fill the before and after versions for all listed files automatically, specify the needed version in the Autofill After versions or Autofill Before versions edit boxes and click Apply. To clear previously specified versions for all listed files, enter empty string to the Autofill edit boxes and click Apply.



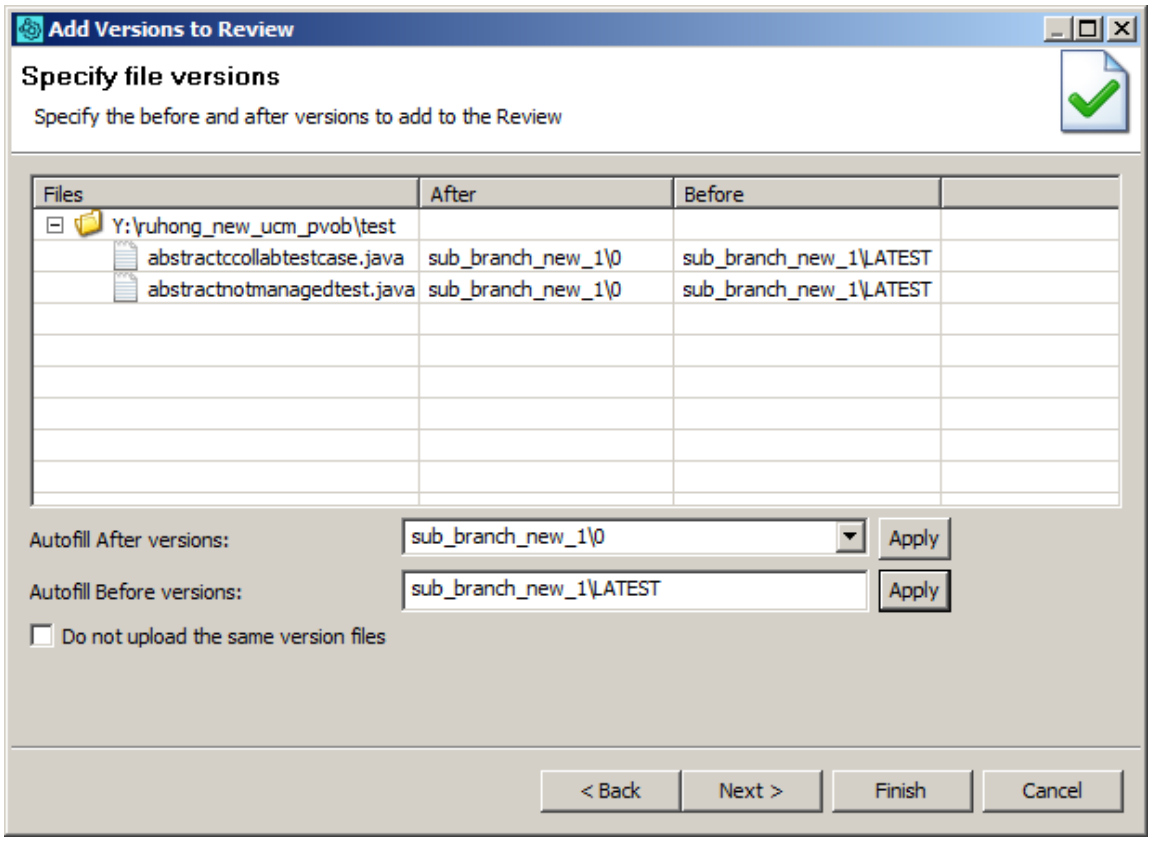

*Automatically fill in before/after values*

You can also specify a branch you want to work with:




*Specifying branch name*

You can input the absolute branch version path (for instance, "\main\branch_new_1 \sub_branch_new_1\1") or a relative path (like, "\sub_branch_new_1\1").




### 6.4.3 Command-Line Client

## Commands recommended for ClearCase

ccollab addchanges 681 - Attaches locally-modified files to a review

ccollab addchangelist 683 - Attaches an atomic changelist to a review

ccollab addversions 684 - Attaches any 2 given versions to a review

ccollab addactivity 688 - Attaches file versions in a ClearCase activity to a review

ccollab commit 690 - Commit changes in the review

The addchanges 629 command will scan for checked-out files and upload them before they are checked in.

There is an option for the addchanges 681 command that allows local changes to be uploaded with a predecessor version in another branch as the previous version. The --diffbranch 681 option is used for this:

```
ccollab addchanges --diffbranch branch-name review file-or-dir [file-or-dir] ...
```

For example, to create a new review and add all changes in the current directory and below, and compare them to the predecessor versions in the \main branch, you would use:

```
ccollab addchanges --diffbranch main new .
```

You can also use the special branch name pre to refer to the predecessor branch. For example, to add the local changes to hello.c compared to the most recent predecessor version in the previous branch to review 45, you would use:

```
ccollab addchanges --diffbranch pre 45 hello.c
```

The addversions command will upload arbitrary versions of ClearCase files after they are checked in.

The addactivity 688 command will upload UCM change sets for review.

## Configuration

In most cases, the Command-Line Client can automatically detect your ClearCase configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your ClearCase configuration or you want to override the detected settings, you can manually specify ClearCase settings using global options 516.

To manually configure the Command-Line Client to use ClearCase, execute the following command:

```
ccollab set scm 517 clearcase
```

## ClearCase-specific Options

| Option | Description |
|---|---|
| `--ccrc-server-url <value>` | CCRC server URL (for ClearCase Remote Client integration only) |

| Option | Description |
|---|---|
| `--ccrc-user <value>` | CCRC username (for ClearCase Remote Client integration only) |
| `--ccrc-passwd <value>` | CCRC password (for ClearCase Remote Client integration only) |
| `--clearcase-exe <value>` | Full path to the `cleartool` command-line client |
| `--clearcase-update-snapshot` | Whether to update ClearCase snapshot views prior to uploading files for review |

#### 6.4.3.1 addchanges (for ClearCase)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by ClearCase to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--diffbranch <value>] [--upload-comment <value>] <review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--diffbranch <value>` | No | (ClearCase only) Specify either branch-name or "pre" to determine the earlier version for upload. See Remarks. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Remarks:

The '--diffbranch' option allows local changes to be uploaded with a predecessor version in another branch as the previous version.

For example, to create a new review and add all changes in the current directory and below, and compare them to the predecessor versions in the \main branch, you would use:

```
ccollab addchanges --diffbranch main new .
```

You can also use the special branch name 'pre' to refer to the predecessor branch. For example, to add the local changes to hello.c compared to the most recent predecessor version in the previous branch to review 45, you would use:

```
ccollab addchanges --diffbranch pre 45 hello.c
```

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

### 6.4.3.2 addchangelist (for ClearCase)

## Description

The `ccollab addchangelist` command attaches all files from a submitted ClearCase changelist (activity) to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or ClearCase-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour: |

| Option | Required? | Description |
|---|---|---|
| | | • new - the command will create a new review,<br>• ask - the command will pause execution and prompt for the identifier of the desired review,<br>• last - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <changelist> [<changelist> ... ] | Yes | Identifier(s) or names of the desired activity (ies) in your source control. |

## Examples:

To upload Activities add_new_scm_support and add_extended_version_parsing to a new review:

```
ccollab addchangelist new add_new_scm_support
add_extended_version_parsing
```

To upload Activities AnActivity and AnotherActivity to review 111:

```
ccollab addchangelist 111 AnActivity AnotherActivity
```

#### 6.4.3.3 addversions (for ClearCase)

## Description

The ccollab addversions command appends the specified versions (revisions) of a file controlled by ClearCase on your computer to a review.

## Command Line Syntax:

```
ccollab [global-options] addversions [--upload-comment <value>] [--
version-spec <value> [<value> ...]] <review> [<file-path>]
[<version>] [<predecessor-version>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `--version-spec <value> [<value> ...]` | No | The version to be added to a review. A version-spec value consist of three components:<br><br>*path version* [*previous-version*],<br><br>where *path* is the file name or server path of the file, *version* is the file version to be reviewed, and *previous-version* is an optional version, against which *version* should be compared.<br><br>If any of these arguments contains spaces, enclose it in quotes.<br><br>Typically a version-spec is not used in the command line. We recommend specifying the file and version using the <file-path>, <version> and the <predecessor-version> arguments (see below). |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br><br>• `new` - the command will create a new review,<br><br>• `ask` - the command will pause execution and prompt for the identifier of the desired review, |

| Option | Required? | Description |
|---|---|---|
| | | • `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-path>` | No | The path of the file whose versions are to be added to the review. If filename is omitted, entire directory will be added.<br><br>**Important:** If you use this option, you should also specify <version> (see below). |
| `<version>` | No | Required, if <file-path> is specified.<br><br>The version (revision) of the file to be added to the review. You can specify the keyword *local* to tell the command to use the local version of the file. |
| `<predecessor-version>` | No | Preceding file version to be added to the review. If you skip this argument, Collaborator will attempt to determine the preceding version based on the information from the source control. |

## Remarks

- If you skip the predecessor version, Collaborator will generate diffs using the predecessor version reported by your source control system.
- By default, the command lets you add versions of one file only. To add versions of multiple files, create a text file and specify this file in the command line as the standard input stream (stdin):

  ```
  ccollab addversions last < versionlist.txt
  ```

  Each line in the file must consist of the following components: *path version* [*predecessor-version* ].

  For information on them, see description of the `version-spec` arguments.
- If you skip the file name and versions in the command line, the command will expect to read them from the standard input stream (stdin). Below are some examples for reading versions from the standard input:

  ```
  ccollab addversions 86753
  ```

```
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

- When specifying the version in the command line or in an input file, you can use the keyword *local* to denote the version corresponding to the local version of the file. The *local* keyword can only be used for the first version argument, not for the predecessor version.

## Examples:

Some examples of specifying versions on the command line for ClearCase:

```
ccollab addversions new ./hello.c /main/mydev/6 /main/8
ccollab addversions last ./Main.java /main/dev/31
```

To compare the local version of the ClearCase file/directory to a previous version in an edit list or input list, use 'local' as the initial version. If the file/directory is checked out as '/main/CHECKEDOUT', then 'local' is equivalent to using '/main/CHECKEDOUT' as the initial version. Otherwise 'local' would refer to the latest version of the file/directory. For example, if the local file hello.c is at version '/main/9' and has predecessor '/main/8' then all of the following lines would be equivalent:

```
./hello.c /main/9
./hello.c local
./hello.c /main/9 /main/8
./hello.c local /main/8
```

You can also reference a LATEST version, for example:

```
ccollab addversions new ./hello.c /main/ga_1.0/6 /main/LATEST
```

To upload a version with no predecessor version, use '/main/0':

```
ccollab addversions new ./hello.c /main/ga_1.0/6 /main/0
```

To compare the version "/main/2" of the directory "W:/user_name_ccrctut_pvob/server" to the version "/main/3" , please run the following command and the difference will be uploaded to Collaborator.

```
ccollab addversions new W:/user_name_ccrctut_pvob/server /main/2 /
main/3
```

#### 6.4.3.4 addactivity

## Description

The `ccollab addactivity` command attaches file versions from one or more ClearCase activities to a review.

## Command Line Syntax:

```
ccollab [global-options] addactivity [--diffintegration] [--upload-
comment <value>] <review> <activity-name> [<activity-name> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--diffintegration` | No | Use LATEST version from default integration stream, if available, as the predecessor version |
| `--upload-comment <value>` | No | Comment used to upload files (default is "Local changes") |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<activity-name> [<activity-name> ...]` | Yes | Specify one or more ClearCase activities by name, or use 'all' to include all activities in the current view |

## Remarks:

- For each file in the activity, the latest version in the activity and the predecessor of the earliest version in the activity are uploaded to the review.
- You can specify multiple activity names on the command line. You can also use the word 'all' as the activity name to include all activities in the current view. If multiple activities are given, the changeset uploaded is the union of all changes in each of the activities.

- You can use the --diffintegration option to include the default integration stream in determining version content for review. For the most recent version, this option will scan the versions of each file in the given activity as well as the versions of those files in any associated rebase activities, to determine the most recent version. For the previous version for each file in the activity, this option will use the LATEST version in the default integration branch (if available) as the previous version.
- IBM Rational ClearCase Remote Client (CCRC) does not support the `addactivity` command.

#### 6.4.3.5 commit (for ClearCase)

## Description

The `ccollab commit` command submits the changes from a pre-commit review to source control. Be sure to include a relevant comment.

## Command Line Syntax:

```
ccollab [global-options] commit [--comment <value>] [--dismiss-only]
[--force] <review>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--comment <value>` | No | Comment for reviewed changes |
| `--dismiss-only` | No | Just dismiss the Action Item |
| `--force` | No | Ignore potential problems |
| `<review>` | Yes | Identifier of the desired review (an integer number), or an `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

### Example:

```
ccollab commit 25 --comment "my code" --force
```

## 6.5 IBM Rational Synergy Integration

This section describes Collaborator integration with Rational Synergy:

### GUI Client 691

The GUI Client can upload Tasks into Collaborator. You can upload Pending 693 or Completed 694 Tasks

### Command-Line Client 695

The Command-Line Client can upload Tasks into Command-Line Client.

### Supported Versions

Our integration uses your own Rational Synergy command-line client (ccm) to communicate with the server. The integration was developed against the Rational Synergy 7.1 toolset, and should work with the latest 6.x and 7.x releases.

### Technical Details and Limitations

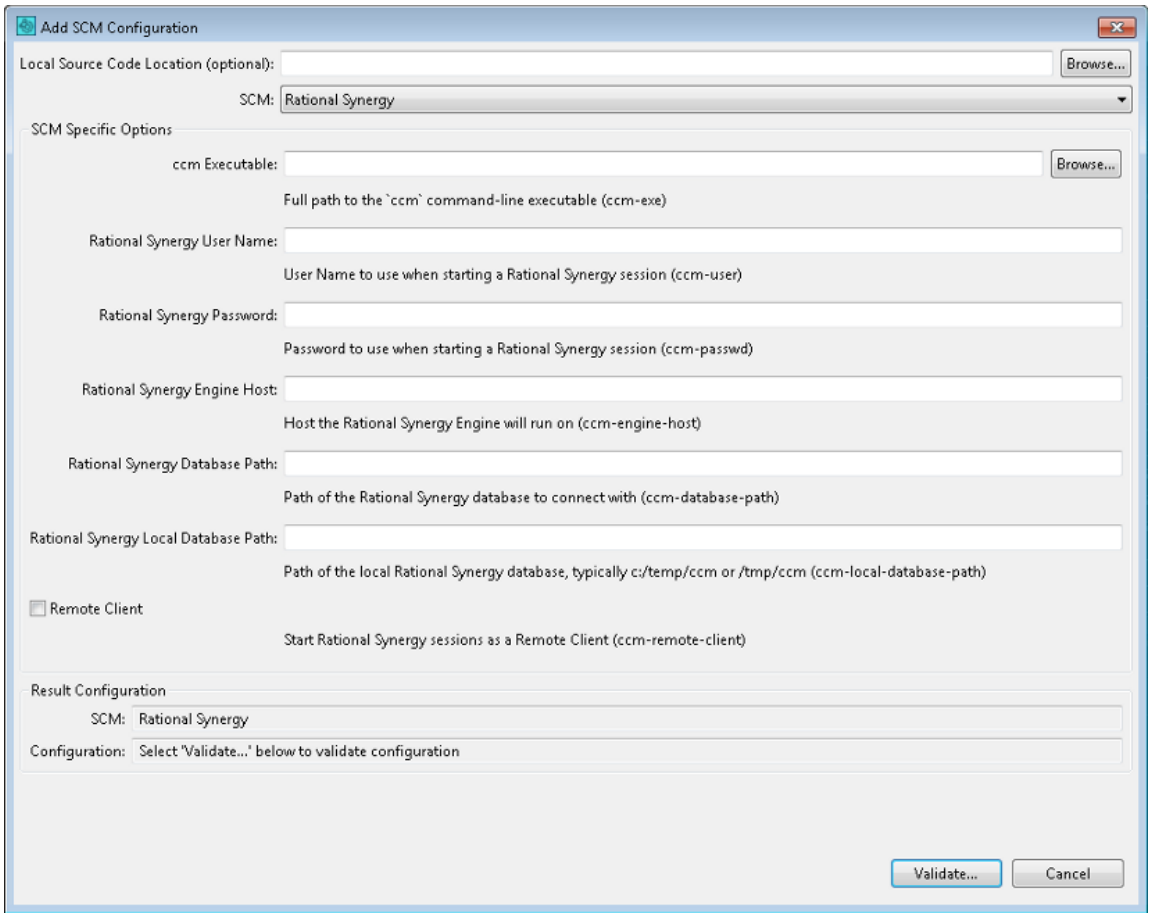
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists (tasks in terms of Synergy) to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.

### 6.5.1 GUI Client

### Rational Synergy-specific Options

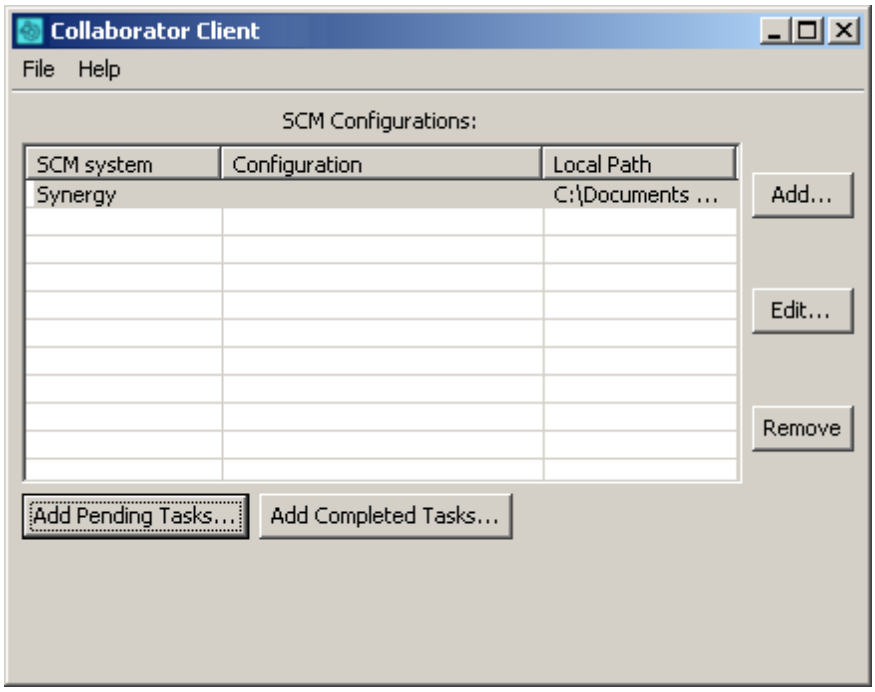
The SCM Configuration dialog 498 has several Rational Synergy-specific options. These can be set as necessary to override Rational Synergy options derived from the environment.




*Synergy SCM Configuration*

## Uploading files to a Review

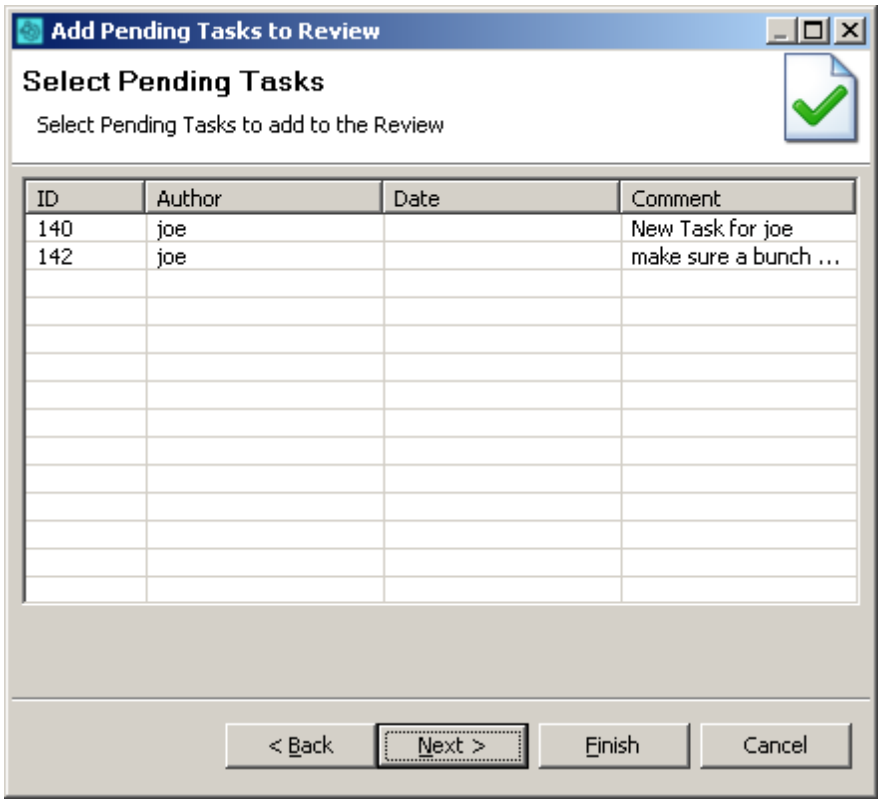
Selecting a Rational Synergy SCM Configuration in the GUI Client main screen[496] causes several Add to Review buttons to appear. The *Add Pending Tasks...*[693] button uploads pending changelists. The *Add Submitted Changelists...*[772] button uploads submitted changelists.




*Uploading Synergy Files to a Review*

## Add Pending Tasks

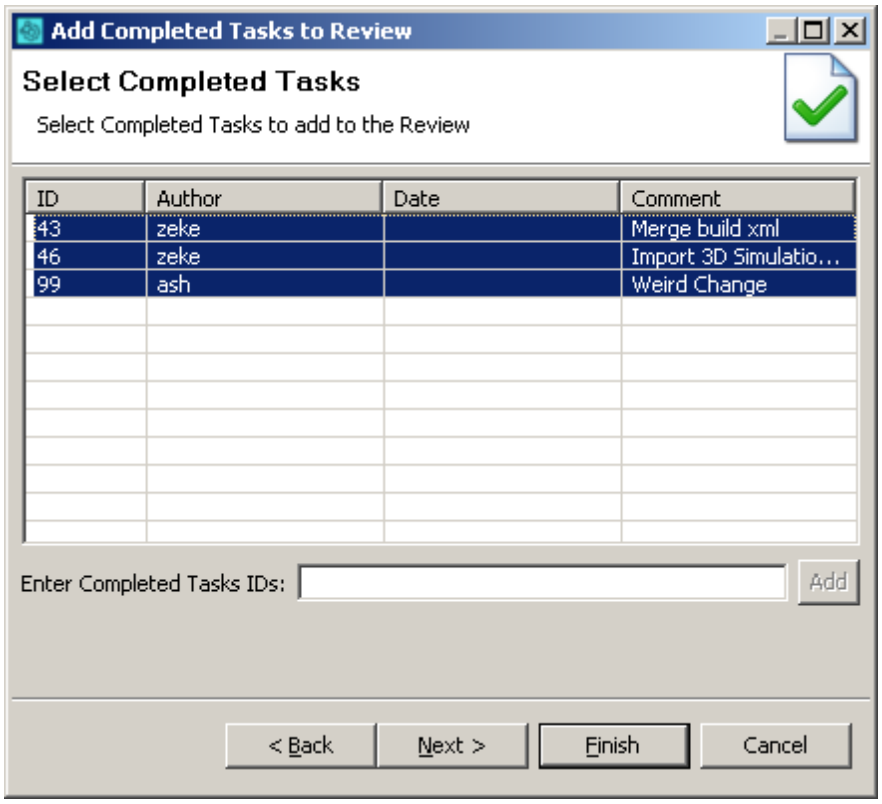
Press the *Add Pending Tasks...* button to upload the files in a pending Task to the Collaborator Server for review.




*Add Rational Synergy Pending Tasks*

## Add Rational Synergy Completed Tasks

Press the *Add Completed Tasks...* button to upload the files in a completed Task to the Collaborator Server for review.




*Add Rational Synergy Completed Task*

### 6.5.2 Command-Line Client

## Commands recommended for Rational Synergy

`ccollab addchangelist` 696 - Attaches an atomic changelist to a review

`ccollab addversions` 698 - Attaches any 2 given versions to a review

The addchangelist 778 command will upload Rational Synergy tasks into Collaborator. You can upload tasks either before or after they are completed.

## Configuration

The Command-Line Client may need some configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Rational Synergy configuration or you want to override the detected settings, you can manually specify Rational Synergy settings using global options 519.

To manually configure the Command-Line Client to use Rational Synergy, execute the following command:

```
ccollab set scm 517 synergy
```

## Rational Synergy-specific Options

| Option | Description |
|---|---|
| `--ccm-exe <value>` | Full path to the `ccm` command-line executable |
| `--ccm-user <value>` | User Name to use when starting a Rational Synergy session |
| `--ccm-passwd <value>` | Password to use when starting a Rational Synergy session |
| `--ccm-engine-host <value>` | Host the Rational Synergy Engine will run on |
| `--ccm-database-path <value>` | Path of the Rational Synergy database to connect with |
| `--ccm-local-database-path <value>` | Path of the local Rational Synergy database, typically c:/temp/ccm or /tmp/ccm |
| `--ccm-remote-client` | Start Rational Synergy sessions as a Remote Client |
| `--ccm-server-url <value>` | Server URL for Web-Mode Rational Synergy servers |

#### 6.5.2.1 addchangelist (for Rational Synergy)

## Description

The `ccollab addchangelist` command attaches all files from a pending or a completed Rational Synergy changelist (tasks) to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ...]` | Yes | Identifier(s) of the desired changeset(s) in your source control. |

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Tasks or Completed Tasks to upload.

## Examples:

To upload Pending Tasks 4321 and 7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

To upload Completed Tasks 5432 and 12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

#### 6.5.2.2 addversions (for Rational Synergy)

## Description

The `ccollab addversions` command appends the specified versions (revisions) of a file controlled by Rational Synergy on your computer to a review.

## Command Line Syntax:

```
ccollab [global-options] addversions [--upload-comment <value>] [--
version-spec <value> [<value> ...]] <review> [<file-path>]
[<version>] [<predecessor-version>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `--version-spec <value> [<value> ...]` | No | The version to be added to a review. A version-spec value consist of three components:<br><br>*path version* [*previous-version*],<br><br>where *path* is the file name or server path of the file, *version* is the file version to be reviewed, and *previous-version* is an optional version, against which *version* should be compared.<br><br>If any of these arguments contains spaces, enclose it in quotes. |

| Option | Required? | Description |
|---|---|---|
| | | Typically a version-spec is not used in the command line. We recommend specifying the file and version using the <file-path>, <version> and the <predecessor-version> arguments (see below). |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-path>` | No | The path of the file whose versions are to be added to the review. If filename is omitted, entire directory will be added.<br>**Important:** If you use this option, you should also specify <version> (see below). |
| `<version>` | No | Required, if <file-path> is specified.<br>The version (revision) of the file to be added to the review. You can specify the keyword *local* to tell the command to use the local version of the file. |
| `<predecessor-version>` | No | Preceding file version to be added to the review. If you skip this argument, Collaborator will attempt to determine the preceding version based on the information from the source control. |

## Remarks

- If you skip the predecessor version, Collaborator will generate diffs using the predecessor version reported by your source control system.
- By default, the command lets you add versions of one file only. To add versions of multiple files, create a text file and specify this file in the command line as the standard input stream (stdin):

  ```
  ccollab addversions last < versionlist.txt
  ```

  Each line in the file must consist of the following components: *path version* [*predecessor-version* ].

  For information on them, see description of the `version-spec` arguments.
- If you skip the file name and versions in the command line, the command will expect to read them from the standard input stream (stdin). Below are some examples for reading versions from the standard input:

  ```
  ccollab addversions 86753
  ccollab addversions last < versionlist.txt
  cat versionlist.txt | ccollab addversions new
  ```
- When specifying the version in the command line or in an input file, you can use the keyword *local* to denote the version corresponding to the local version of the file. The *local* keyword can only be used for the first version argument, not for the predecessor version.

## Example:

```
ccollab addversions new ./hello.c:csrc:1 3.2.5 3.2
```

# 6.6 IBM Rational Team Concert Integration

Collaborator integrates with Rational Team Concert allowing you to automatically synchronize the approver lists of Collaborator reviews and Team Concert work items, and automatically create reviews for change sets. Topics of this section provide detailed information on the integration.

## In This Section

## 6.6.1 Overview

### About Integration

Collaborator tightly integrates with Rational Team Concert. The integration helps you easily create reviews for Team Concert's work items, synchronize the list of reviewer participants and work item's approvers, and update the state attribute of users in the Approvals list automatically when the review status changes. For detailed information on the features, see **How Integration Features Work** 704.

### Supported Versions

Collaborator 12.0 and later supports Rational Team Concert versions 6.0.5 - 7.0.1.

Collaborator 11.1 - 11.5 supports Rational Team Concert versions 4.x - 6.0.5.

If you have Rational Team Concert versions 2.x - 3.x, you will need to downgrade your Collaborator (both Server and Client components) to CodeCollaborator/PeerReview Complete version 6.5.x (as only these versions supported Java 5 required by RTC 2.x - 3.x).

For more details, see Version Compatibility Table section below.

### Required Plug-Ins

The integration functionality is provided by two plug-ins:

- One of them is installed on the Rational Team Concert server machine (Collaborator Plug-In for Rational Team Concert Server).
- Another plug-in can be installed on client computers that have the Eclipse IDE (Collaborator Plug-In for Eclipse).

Both plug-ins can be downloaded from our web site:

http://support.smartbear.com/downloads/collaborator/

For information on installing and configuring the plug-ins and servers, see topics of the **Configuring Servers, Clients and Plug-Ins** 713 section. We strongly recommend that you read it.

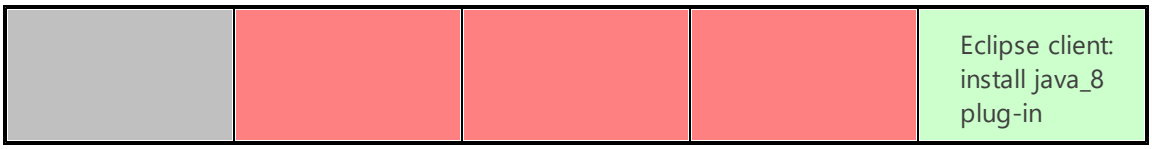
## Version Compatibility Table

Different versions of Collaborator servers, Rational Team Concert servers and Eclipse IDEs require different versions of Java environment. To integrate Collaborator and Rational Team Concert you will need to install a specific version of Rational Team Concert server side plug-in **and** a specific version of Eclipse client side plug-in.

A table below lists the possible combinations:

| **Rational Team Concert Version**<br>**Collaborator Server Version** | **2.x - 3.x (Java 5)** | **4.x - 6.0 (prior to 6.0.1) (Java 6)** | **6.0.1 - 6.0.5 (Java 7)** | **6.0.5 - 7.x (Java 8)** |
|---|---|---|---|---|
| **6.5.x and earlier (Java 5)** | RTC server: install 6.5.6511 plug-in<br><br>Eclipse client: install 6.5.6511 plug-in | Incompatible | Incompatible | Incompatible |
| **6.5.x - 8.4.8406 (Java 6)** | Incompatible | RTC server: install 8.4.8406 plug-in<br><br>Eclipse client: install 6.5.x - 8.4.8406 plug-in | Incompatible | Incompatible |
| **8.5.8500 - 9.5.9501 (Java 7)** | Incompatible | RTC server: install 8.4.8406 plug-in | Incompatible | Incompatible |

| | | | | |
|---|---|---|---|---|
| | | Eclipse client: install 8.5.8500 - 9.5.9501 plug-in | | |
| **10.0.10000 - 11.0.11000 (Java 7)** | Incompatible | Incompatible | RTC server: install 10.0.10000 - 11.0.11000 plug-in<br><br>Eclipse client: install 10.0.10000 - 11.0.11000 plug-in | Incompatible |
| **11.1.11100 - 11.2.11201 (Java 7)** | Incompatible | RTC server: install 8.4.8406 plug-in<br><br>Eclipse client: install 11.1.11100 - 11.2.11201 plug-in | RTC server: install 11.1.11100 - 11.2.11201 plug-in<br><br>Eclipse client: install 11.1.11100 - 11.2.11201 plug-in | Incompatible |
| **11.3.11300 - 11.5.11504 (Java 8)** | Incompatible | RTC server: install 8.4.8406 plug-in<br><br>Eclipse client: install 11.1.11100 - 11.2.11201 plug-in | RTC server: install java_7 plug-in<br><br>Eclipse client: install java_8 or java_7 plug-in, depending on IDE requirement | Incompatible |
| **12.0.12000 and later** | Incompatible | Incompatible | Incompatible | RTC server: install java_8 plug-in |




## Technical Details and Limitations

Review Screen, Diff Viewer, Eclipse Plug-in and Visual Studio Extension display atomic changelists (changesets in terms of RTC) in chronological order (from older to newer), regardless the order in which they have been uploaded to review.

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists (changesets in terms of RTC) to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- If you add changelists from different workspaces or streams to the same review.
- If you add several diffs (non atomic changelists) to the same review.

### 6.6.2 How Integration Features Work

**Important:** This topic describes how Collaborator integrates with Rational Team Concert and Eclipse. In order for the described integration features work, you need to configure the Rational Team Concert and Collaborator servers and clients properly. See Configuring Servers, Clients and Plug-Ins 713.

## Creating Reviews Automatically

Collaborator's plug-in for Rational Team Concert tracks the changes of a work item state. If you change the state of a work item, the plug-in creates a review for that work item in Collaborator and attaches the changelist to this review. The plug-in also creates a review, if you add a changelist to a work item that has no changelist associated with it.

If a review was created in this way, Collaborator will track changes in the work item's Approvals list and update the review's Participants list appropriately. Also, you will see the status of a user work on the review in the **State** column of the Approvals list.

**Note:** The plug-in does not update a review if you update an existing changelist.

**Requirements:**

- To use automatic review creation, you need to --
  - Configure 713 the Collaborator server.
  - Install 717 and configure 721 the Collaborator plug-in for Rational Team Concert.
  - Set up 722 follow-up actions in Rational Team Concert.
- To take advantage of user list and state synchronization, you also need to configure 713 the Collaborator server.

## Creating Reviews From Eclipse Manually

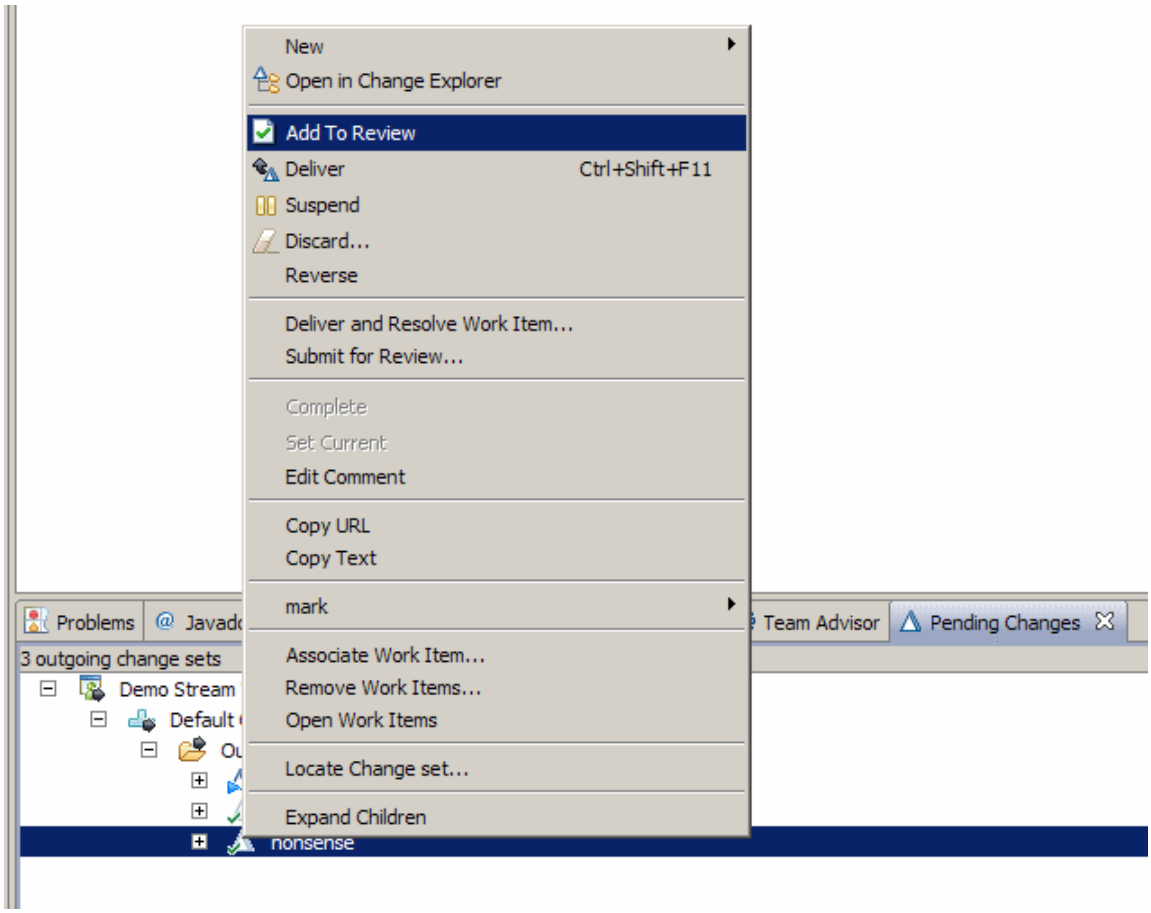
Collaborator's Eclipse plug-in inserts new **Add To Review** items to the context menus of these views:

1. Pending changes view

2. History view

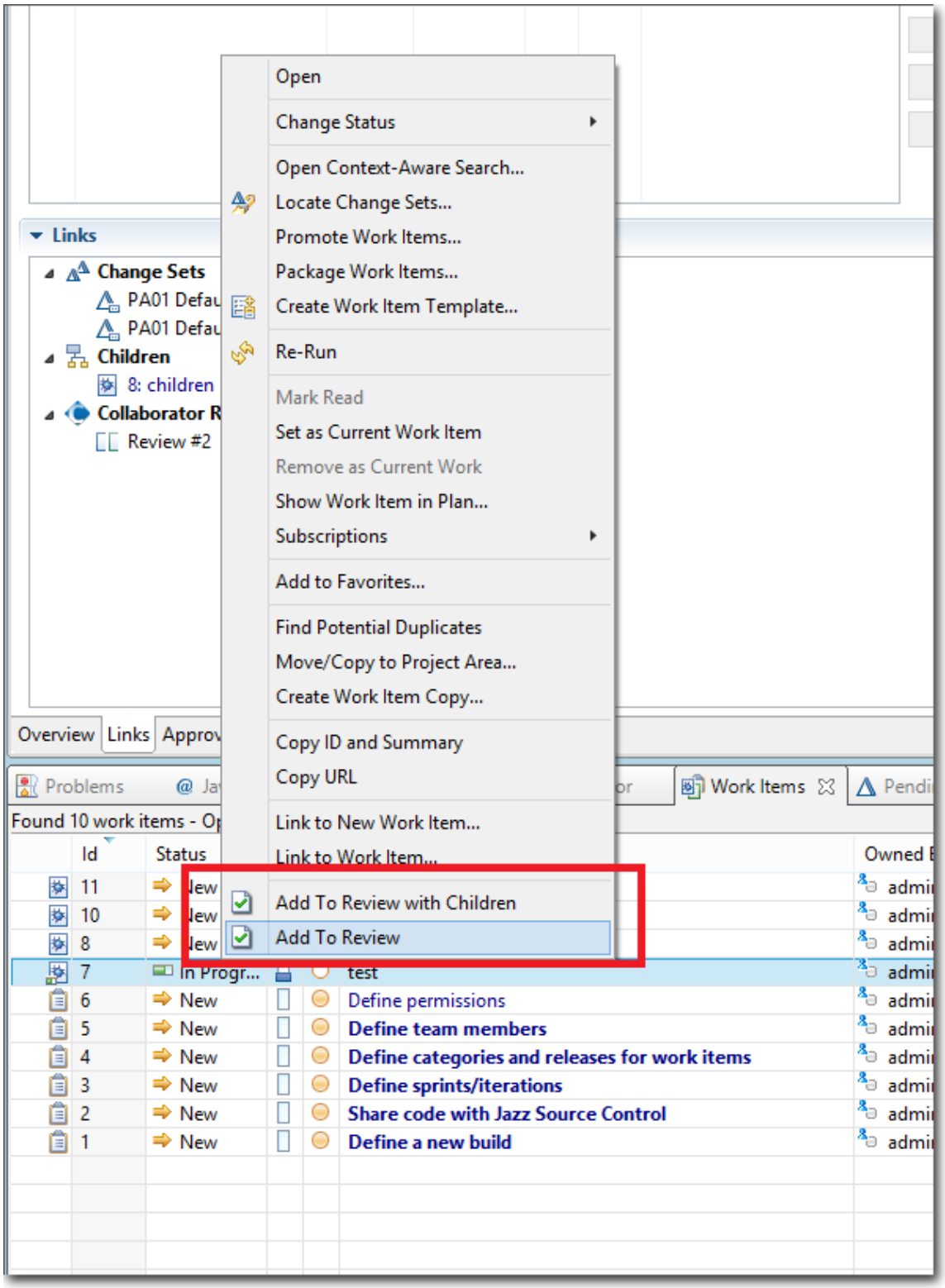
3. Work Items view

Right-click any item (for example, a changelist) on these views and select an **Add to Review** menu item. Follow instructions of the subsequent wizard to create a new review or add the changelist to an existing review.



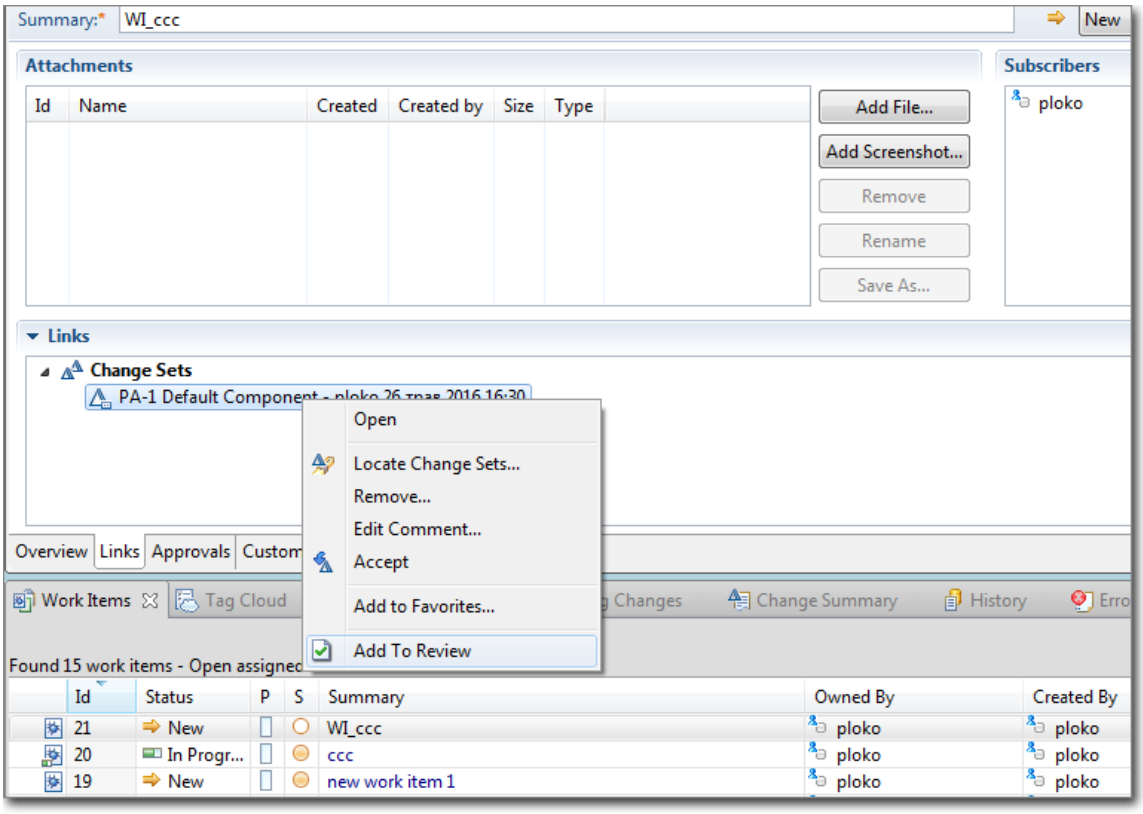

*Creating a review from a changeset*

On Work Items view, an **Add to Review with Children** menu item will also appear. When you use this function, the Collaborator will try to fetch changelist from selected Work Item and its children.




*Creating a review from the Work Items context menu*

Additionally, you can use an **Add to Review** function from the Links tab of the Work Items view.




*Creating a review from the Links context menu*

**Note**: You have to save the work item state before trying to add changelists to review.

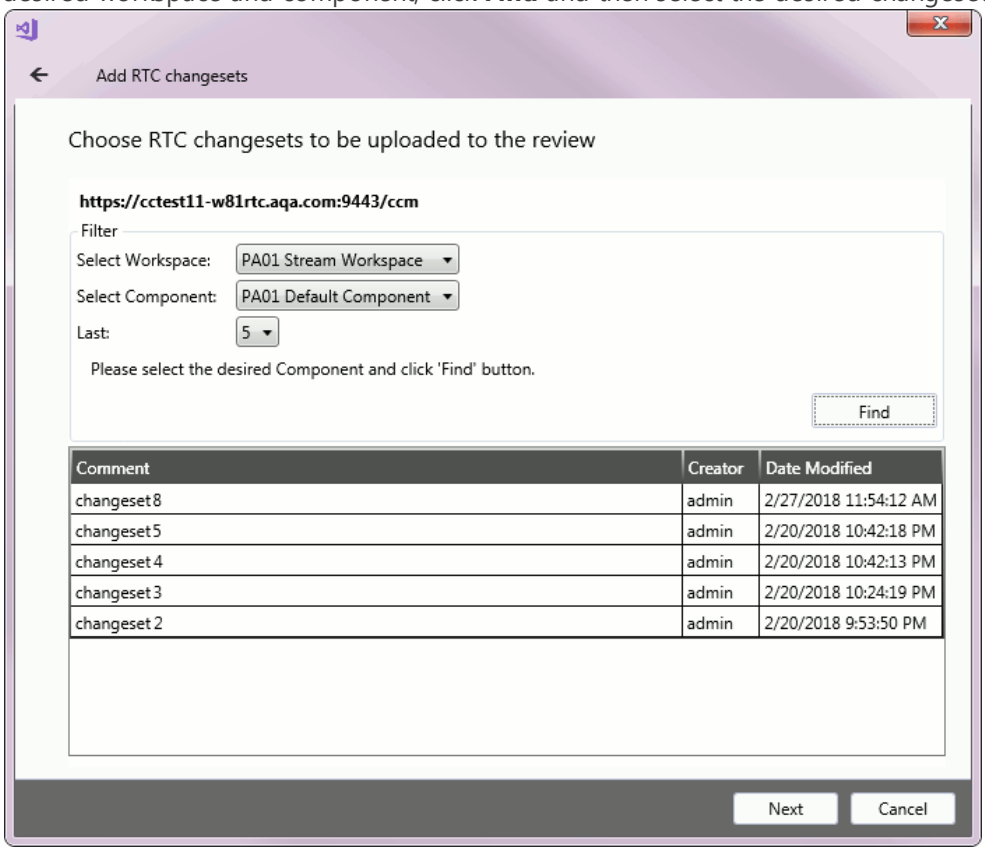
**Requirements:** to use this feature, you need to --

- Configure[713] the Collaborator server.
- Have Rational Team Concert plug-ins installed into your Eclipse IDE.
- Install[716] the Collaborator plug-in for Eclipse.

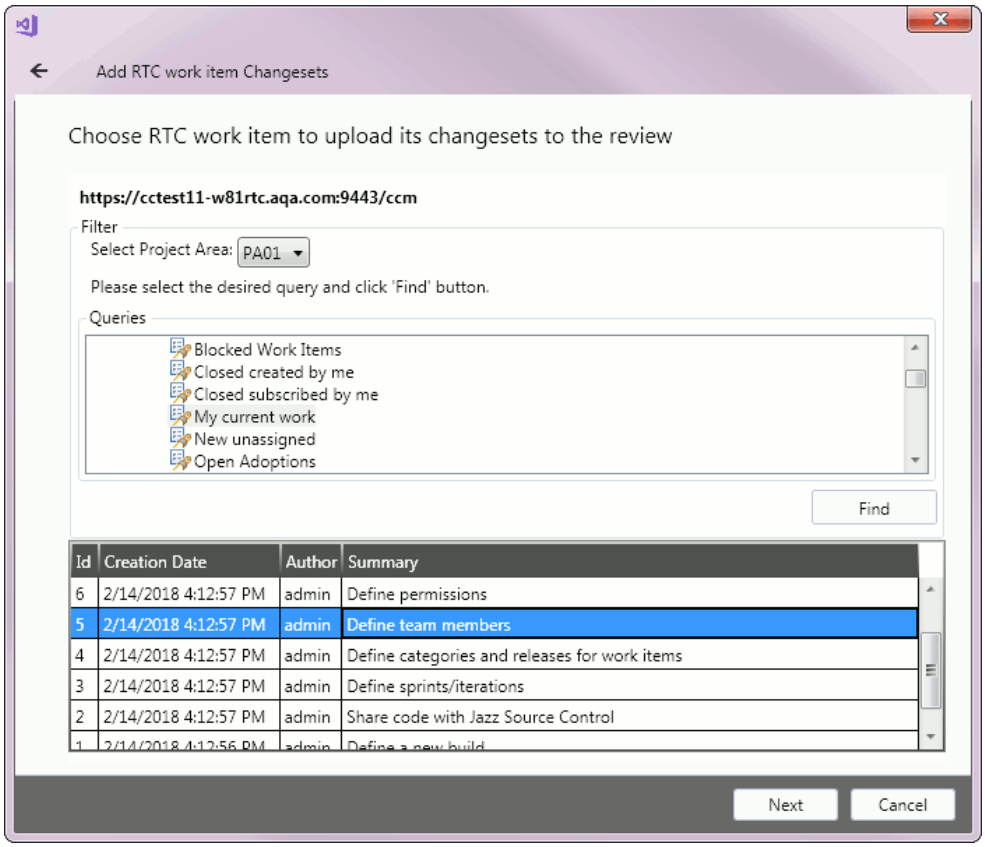
## Creating Reviews From Visual Studio Manually

You need to configure Collaborator's Visual Studio Extension[569] and select your RTC configuration as the default SCM configuration. After that you can call the extension's Add to Review Wizard and create reviews reviews based on RTC changesets and work items.

- To create reviews for a specific RTC changesets: Select "Add RTC changesets", choose the desired workspace and component, click **Find** and then select the desired changesets.



- To create reviews for changesets linked to a specific RTC work item: Select "Add RTC work item changesets", choose the project area and perform one of the listed queries, click **Find** and then select the desired work item.




**Requirements:** to use this feature, you need to --

- Configure 713 Collaborator server and client.
- Install 717 and configure 721 the Collaborator plug-in for Rational Team Concert.
- Install 568 and configure 569 the Collaborator Visual Studio Extension.

## Creating Reviews From Command-Line Client Manually

## Commands recommended for Rational Team Concert

`ccollab addchangelist` - Attaches an atomic changelist to a review

## Configuration

The Command-Line Client may need some configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Rational Team Concert configuration or you want to override the detected settings, you can manually specify Rational Team Concert settings using global options 519.

To manually configure the Command-Line Client to use Rational Team Concert, execute the following command:

```
ccollab set scm 517 rtc
```

## Rational Team Concert-specific Options

In order to use Rational Team Concert integration from command-line client, you need to configure both Collaborator server and clients and RTC server. See configuration instructions 713.

| Option | Description |
|---|---|
| `--rtc-repository-uri <value>` | The URI of the repository to work with. |
| `--rtc-username <value>` | The user name to use for the chosen RTC repository. |
| `--rtc-password <value>` | The password of the user. |

**Requirements:** to use this feature, you need to --

- Configure 713 Collaborator server and client.
- Install 717 and configure 721 the Collaborator plug-in for Rational Team Concert.

## Synchronizing the Approvals and Participants Lists

For reviews that were created automatically on the work item's state change, Collaborator synchronizes the contents of the work item's Approvals list and the review's Participant list:

- When you append a Reviewer entry (user) to the Approvals list, Collaborator includes the same user into the Participants list of the review as a *Reviewer*. This also works in the opposite way: if you add users to participants, they are also added to approvers.
- Synchronization also works for deletion: if you remove a user from participants in Collaborator, the plug-in will remove that user from the Approvals list of your work item. Note, however, that the opposite deletion operation is not available at the moment, that is, if you delete a user from the Approvals list of a work item, Collaborator will not remove that user from the Participants list.

**Notes:**

- In order for this feature to work, the user must have the same name in Collaborator and in Rational Team Concert.
- When tracking changes made to the Approvals list, Collaborator tracks the addition of Reviewer entries only. If you add an entry of the Approver type, the review's Participants list will not be updated.
- Collaborator updates the Approvals list after the review left the Planning phase and moved to the Annotating phase. Any participant that was added during the Planning phase will also be included into the work item once you move the review to the Annotating phase.
- Collaborator synchronizes only those users, whose role allows them to finish the review. Participants that do not need to finish the review (for example, Observers) are not added to the work item's Approvals list. Similarly, if you remove such a user from the Participants list, they will not be deleted from the Approvals list. If you assign a participant a role that does not allow them to finish review (for example, change the role from *Reviewer* to *Observer*), Collaborator will delete this participant from the Approvals list.
- Once again, synchronization works for those reviews that were created automatically. It does not work for the reviews that were created through the "Add To Review" menu command.

If you do not want automatic updates of the user list, disable the *Add reviewers* and *Remove reviewers* settings of the Collaborator server. See Configuring Collaborator Server.

**Requirements:** to use this feature, you need to --

- Configure the Collaborator server.
- Install and configure the Collaborator plug-in for Rational Team Concert.
- Set up follow-up actions in Rational Team Concert.

## Updating Approver State

If a review was created automatically on the work item's state change, Collaborator tracks the changes made to the review phase. If a user completes the review, or cancels or rejects it, Collaborator updates the State attribute of that user in the Approvals list.

- If a user completed the review, the state is changed to **Approved**.
- If a user cancels or rejects the review, the state is changed to **Rejected**.

**Requirements:** to use this feature, you need to --

- Configure the Collaborator server.
- Install and configure the Collaborator plug-in for Rational Team Concert.
- Set up follow-up actions in Rational Team Concert.

## Demonstration

Watch the following video to see Collaborator's integration with Rational Team Concert:

https://support.smartbear.com/screencasts/collaborator/rational-team-concert-demo/

## Troubleshooting

See Troubleshooting 727.

### 6.6.3 Configuring Servers, Clients and Plug-Ins

Topics of this section describes how to configure the Collaborator server, client and plug-ins to use the Rational Team Concert integration features.

## In This Section



#### 6.6.3.1 Configuring Collaborator Server and Clients

To use Collaborator with Rational Team Concert, you need to configure your Collaborator server and its clients. Follow these steps:

1. Install RTC service to your Collaborator server. This service is installed by default during Collaborator server installation. If have disabled this service earlier, launch server installer 70 in update mode and enable the "Install RTC service" advanced option 77 during installation.

2. Download the "Rational Team Concert Plain Java Client" libraries that match your Rational Team Concert version to your Collaborator server computer from the https://jazz.net/downloads/rational-team-concert/ web site.

3. Extract the files from the downloaded .zip to some temporary directory.

4. Delete the following .jar files from the temporary folder. This will prevent conflicts with libraries that the Collaborator server already uses.
   The .jar file names may include version numbers. Delete the files that matches the names below, regardless of the version number in their name:

   - `commons-io`
   - `org.apache.commons.codec`
   - `javax.servlet`

5. Stop the Collaborator server.

6. Copy the extracted .jar files from the temporary directory to **<Your_Collaborator_Server_Dir>**/tomcat/shared_lib. Create the shared_lib folder if it does not exist.

7. If you plan to use RTC integration from Command-Line Client or from Visual Studio Extension, then delete one more file from the temporary directory with the extracted .jar files.
   You need to delete the `org.apache.log4j` file (regardless of the version number in its name) and copy the remaining .jar files to ***<Your_Collaborator_Client_Dir>****/lib*.

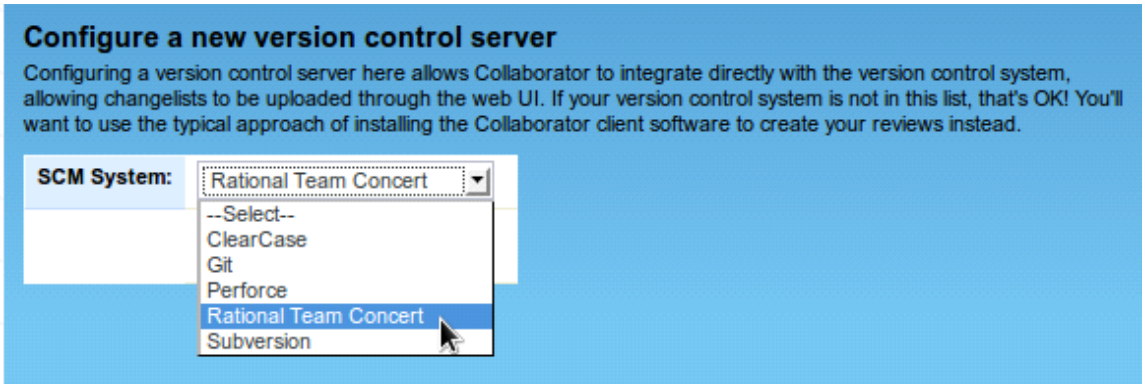
8. Start the Collaborator server.

9. Open the Collaborator Web Client. Select **Admin** from the main menu. This will open the Administration settings page.

10. Choose **Version Control** from the menu on the left. Scroll the page down to the **Configure a new version control server** section (it is at the end of the page).

11. Create a new configuration for Rational Team Concert:

    - Select **Rational Team Concert** from the **SCM System** drop-down list and click **Create**:



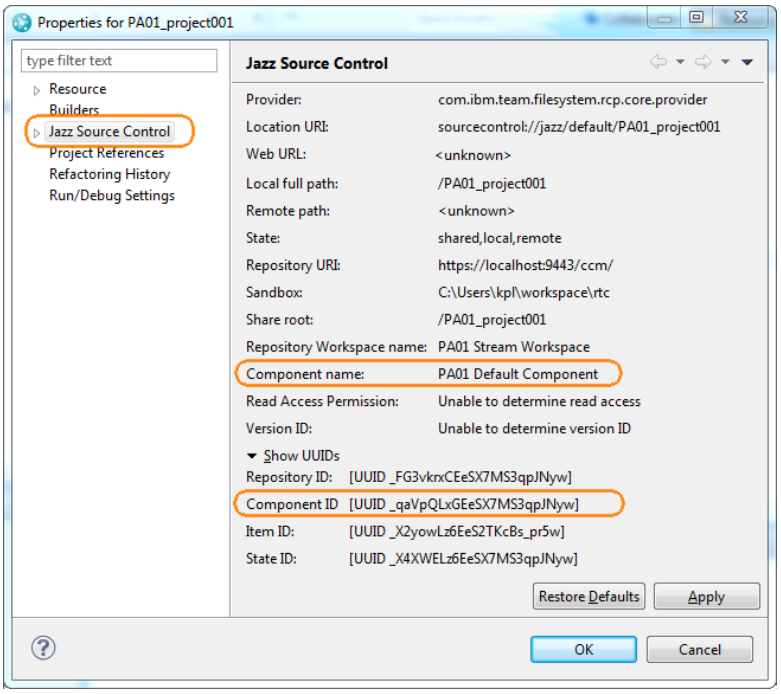

    - Specify the new configuration's settings:

| | |
|---|---|
| **Title** | The configuration name. It identifies the configuration in Collaborator. |
| **Server URI** | The address of your Rational Team Concert server. Typically, this is a Public URI of the server. If the Public URI is inaccessible to you for some reason, you can specify the server's IP address. |
| **Admin user name**, and **Admin password** | The user account and password that Collaborator will use to connect to the Rational Team Concert server.<br>**Note:** This account must have administrator permissions in Rational Team Concert. |
| **Add reviewers** | Specifies whether Collaborator will track addition of new entries (users) to a work item's Approvals list and adds the appropriate users to a review's Participants list. See How Integration Features Work 704. |
| **Remove reviewers** | Specifies whether Collaborator will track deletion of entries (users) in the Participants list and update the work item's Approvals list accordingly. See How Integration Features Work 704. |
| **Update assignments** | Specifies whether Collaborator updates the **State** attribute of a user in a work item's Approvals list when that user completes, cancels or rejects the review. See How Integration Features Work 704. |

- Click **Save** after you specified the settings.
- Click **Test Connection** to verify that the connection between Collaborator server and your Rational Team Concert server is established successfully and is trusted. For example, you might need to generate and/or accept trust certificates and so forth.
- The Client Configuration Mapping section defines correspondence between files uploaded by the various Collaborator clients and this server-side Rational Team Concert configuration.
  Specify Java-style regular expressions to map changelists uploaded from our client tools to this Rational Team Concert server:

| | |
|---|---|
| **Team repository** | The URL of your Team Concert repository. |
| **Component** and **Component UUID** | Parameters of your Team Concert project. To view them:<br>• In Eclipse, in **Project Explorer**, right-click the project and select **Properties** from the context menu.<br>• In the subsequent Properties dialog, select **Jazz Source Control** from the tree on the left. Find the parameter values on the right: |

**Team repository** The URL of your Team Concert repository.




- Press **Save** to apply mapping configuration and enable server-side integration with this Rational Team Concert repository.

## Other Configuration Actions

In addition to configuring the Collaborator server, you also need to install and configure special plug-ins for Rational Team Concert. For complete information on this, see topics of the Configuring Servers and Plug-Ins 713 section.

### 6.6.3.2 Installing Collaborator Plug-In for Eclipse

To create Collaborator reviews manually 704 from the Eclipse IDE, you need to install and configure the Collaborator plug-in for Eclipse. This topic provides detailed information on this.

**Note:** The Collaborator plug-in for Eclipse extends the functionality of Rational Team Concert plug-ins for Eclipse. If these Team Concert plug-ins are not installed, the Collaborator plug-in for Eclipse will not work.

To install and configure the Collaborator plug-in for Eclipse:

1. Download the Collaborator Eclipse Plug-In that matches Java version required by your Eclipse IDE:

| | |
|---|---|
| Eclipse 4.7 (Oxygen) | Java 8 or newer |
| Eclipse 4.6 (Neon) | Java 8 |
| Eclipse 4.5 (Mars) | Java 7 |
| Eclipse 4.4 (Luna) | Java 7 |
| Eclipse 4.3 (Kepler) | Java 6 (need to switch it to use Java 7 as described below) |

***Note:*** *For more information about different versions, see Version Compatibility Table in Overview 702.*

2. Follow the Eclipse plug-in's Install & Update instructions 526 to setup and configure the Eclipse plug-in.

3. If your version of Eclipse uses Java 6 or 5 (Eclipse 4.3 Kepler and earlier), then it is necessary to change the eclipse.ini file to use Java 7. Details about this change can be found at the following site:

   http://wiki.eclipse.org/FAQ_How_do_I_run_Eclipse%3F

Here is an example of the change:

```
-vm C:\Program Files\Java\jdk1.7\jre\bin\javaw.exe
```

## Other Configuration Actions

If you are going to create reviews manually, then all you need is to install and configure the Collaborator plug-in for Eclipse. To use other integration features 704, you need to install configure the Collaborator server and install and configure special plug-ins for Rational Team Concert. For complete information on this, see topics of the Configuring Servers and Plug-Ins 713 section.

### 6.6.3.3 Installing Collaborator Plug-In for Rational Team Concert Server

This topic explains how to install the Collaborator plug-in for Rational Team Concert.

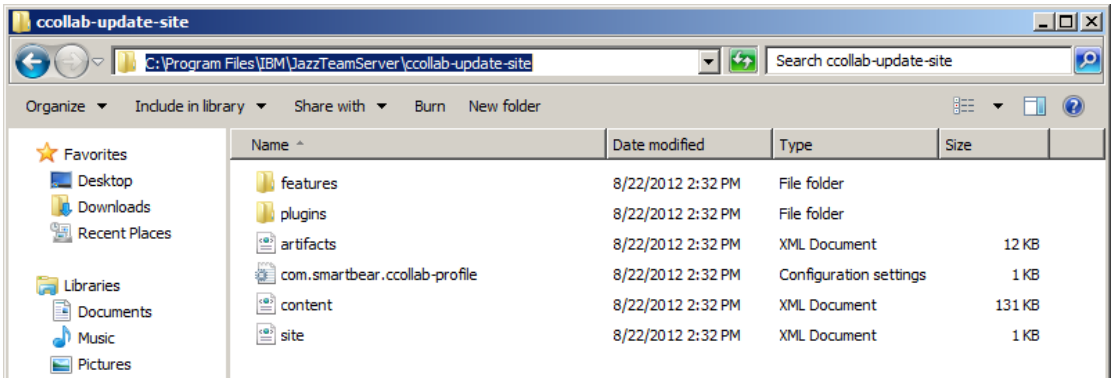
**Important:** Since version 8.5 Collaborator requires Java 7. Rational Team Concert 6.0.1 and later also require Java 7. Whereas, Rational Team Concert 4.x - 6.0 (prior to 6.0.1) require Java 6, and RTC 2.x - 3.x require Java 5. Therefore to integrate Collaborator and Rational Team Concert you will need to install a specific version of Rational Team Concert Plug-In.
To integrate with Rational Team Concert 2.x - 3.x you also need to downgrade your Collaborator (both Server and Client components) to CodeCollaborator/PeerReview Complete version 6.5.x (as only these versions

supported Java 5).

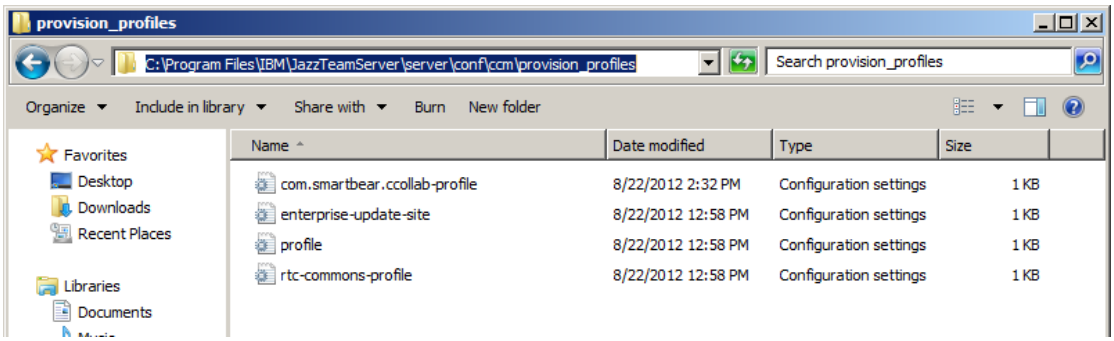
1. Download the update site .zip package with the plug-in from our web site:
   - For Rational Team Concert Server version 6.0.1 and higher download the java_7 version of Collaborator Eclipse Plug-In.
   - For Rational Team Concert Server versions 4.x - 6.0 (prior to 6.0.1) download Rational Team Concert Plug-In version 8.4.8406.001.

   ***Note:*** *For more information about different versions, see Version Compatibility Table in Overview* 702.

2. Extract files from the downloaded .zip archive to the <$JAZZ_ROOT>/ccollab-update-site folder on the Rational Team Concert server computer. For example, the folder name can be like C:\Program Files\IBM\JazzTeamServer\ccollab-update-site.




3. Copy the *com.smartbear.ccollab-profile.ini* file from this folder to the *<$JAZZ_ROOT>\server\conf\ccm\provision_profiles* folder.




4. Now you need to reset your Jazz Team Server:
   - Open the following web page in your web browser:

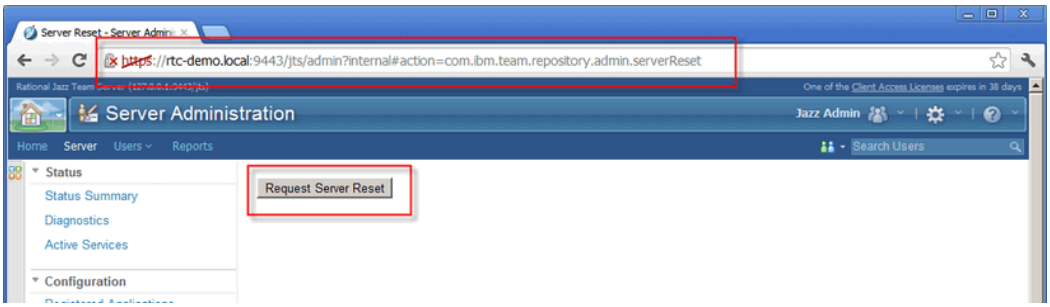
     If you use Rational Team Concert 4 or later:

     https://$Your_Jazz_server/jts/admin?internal#action=com.ibm.team.repository.admin.serverReset

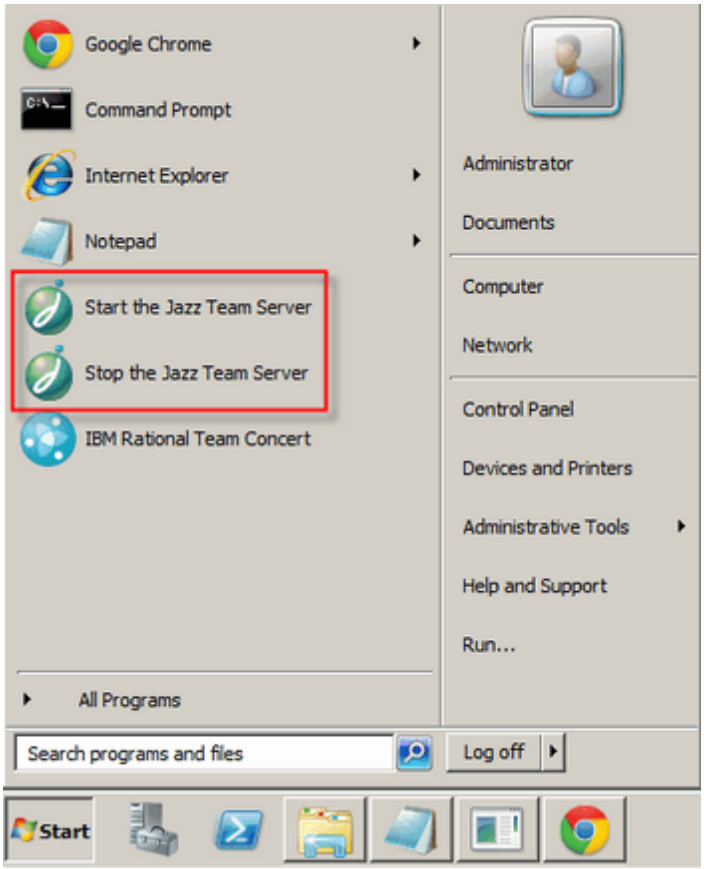
     If you use Rational Team Concert 3 or earlier:

https://$Your_Jazz_server/jazz/admin?internal=true

- On the page, click **Request Server Reset**. This will command Rational Team Concert to load the Collaborator plug-in next time the Rational Team Concert service restarts.

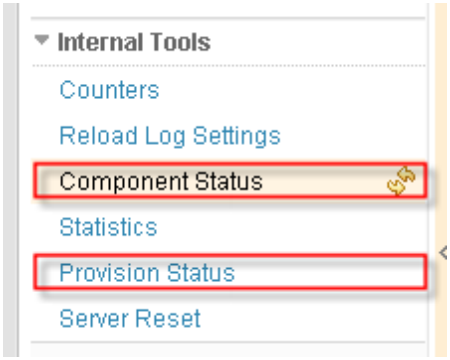



5. Restart the Jazz Server. To do this:
   - Select **Stop the Jazz Team Server** from Windows' Program Files menu, or run the *shutdown.bat* file in the Jazz Install directory.
   - Run the server by selecting **Start the Jazz Team Server** from the Program Files menu:



6. Check whether the plug-in is loaded:
   - Open the Component Status page:

   https://$Your_Jazz_server/ccm/admin?internal=true#action=com.ibm.team.repository.admin.componentStatus
   - Click the refresh icon next to the **Component Status** item in the left-side menu to update the page after it loads.
   - Check that the *com.smartbear.collaborator.rtc.process.ICollaboratorService* service is active.
   - Now click Provision Status in the menu menu entry:




   - Search for *smartbear* on the page and make sure that the plug-in has been installed and initialized.

Provision Status page: Checking whether the plug-in is installed

```
CRJAZ0300I This feature is being installed: "com.smartbear.collaborator.rtc.
process.feature_NN.N.NNNNN.NNN".
CRJAZ0299I Installing bundle from the URL "file:/<JAZZ_ROOT>/server/conf/../../
ccollab-update-site/plugins/com.smartbear.collaborator_NN.N.NNNNN.NNN.jar".
CRJAZ0299I Installing bundle from the URL "file:/<JAZZ_ROOT>/server/conf/../../
ccollab-update-site/plugins/com.smartbear.collaborator.rtc.common_NN.N.NNNNN.
NNN.jar".
CRJAZ0299I Installing bundle from the URL "file:/<JAZZ_ROOT>/server/conf/../../
ccollab-update-site/plugins/com.smartbear.collaborator.rtc.process_NN.N.NNNNN.
NNN.jar".
CRJAZ0299I Installing bundle from the URL "file:/<JAZZ_ROOT>/server/conf/../../
ccollab-update-site/plugins/com.smartbear.collaborator.rtc.component_NN.N.
NNNNN.NNN.jar".
```

where *NN.N.NNNNN.NNN* stands for the version number of Collaborator plug-in.

Provision Status page: Checking whether the plug-in is started

```
CRJAZ0307I Starting the bundle "com.smartbear.collaborator_NN.N.NNNNN.NNN".
```

```
CRJAZ0307I Starting the bundle "com.smartbear.collaborator.rtc.common_NN.N.
NNNNN.NNN".
CRJAZ0307I Starting the bundle "com.smartbear.collaborator.rtc.process_NN.N.
NNNNN.NNN".
CRJAZ0307I Starting the bundle "com.smartbear.collaborator.rtc.component_NN.N.
NNNNN.NNN".
```

where *NN.N.NNNNN.NNN* stands for the version number of Collaborator plug-in.

## Other Configuration Actions

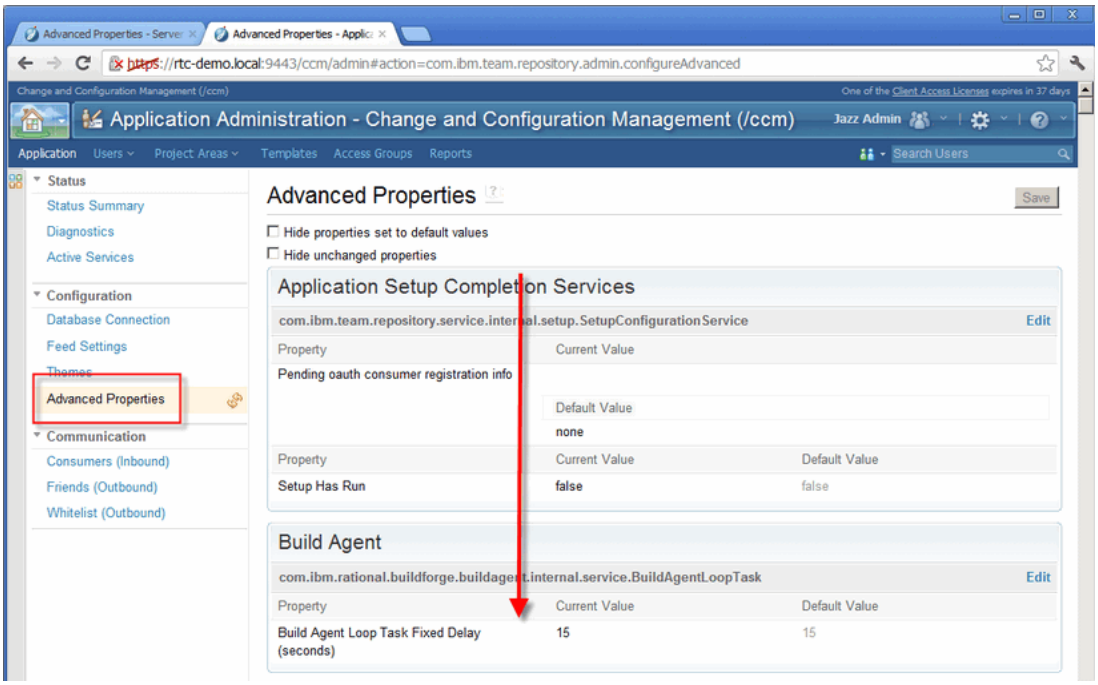
After you installed the plug-in, you need to configure it. See Configuring the Collaborator Plug-In for Rational Team Concert 721.

### 6.6.3.4 Configuring Collaborator Plug-In for Rational Team Concert Server

After installing the Collaborator plug-in for Rational Team Concert, you need to configure it. Follow these steps:

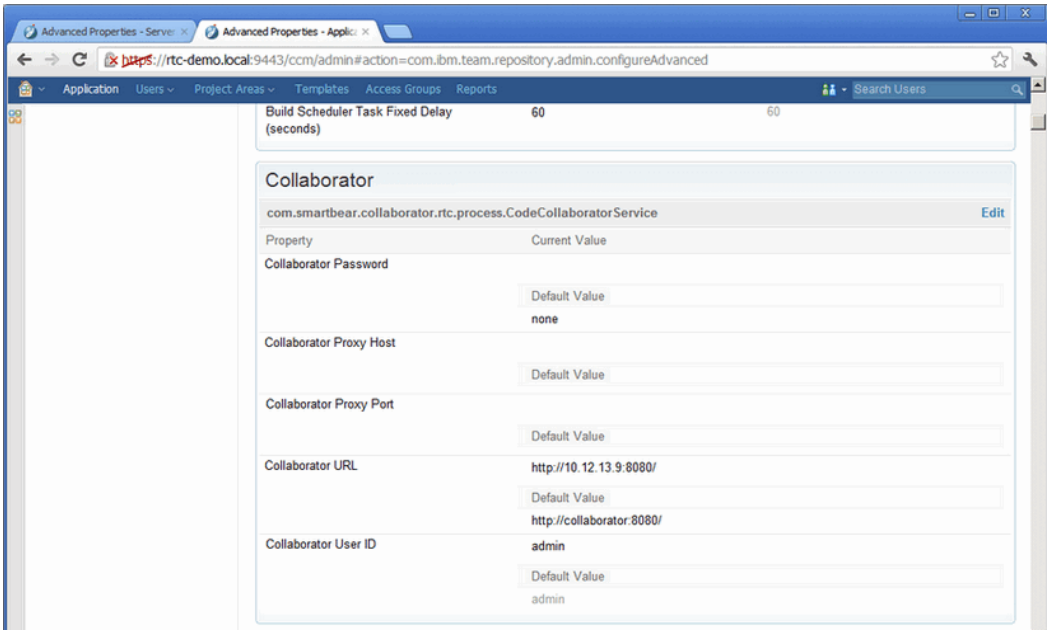
1. Login to the Change and Configuration Manager at https://***$Your_Jazz_server***/ccm/admin, for example, https://rtc-demo.local:9443/ccm/admin.

2. Select **Advanced Properties** from the menu on the left and then scroll down to the Collaborator section:



3. Configure the settings and save them (see the image below). The settings have self-descriptive names, and hardly require explanation.
   Two notes:

   - The Collaborator User ID setting specifies the user account that sends notifications on file uploads. Uploads from RTC to Collaborator will appear to have been done by this user. Collaborator is unable to display the name of the RTC user that actually made the changes, this is a limitation of the RTC/Collaborator integration. This account should have *administrator permissions* in Collaborator.
   - Specify host and port if you connect to Collaborator via proxy.




## Other Configuration Actions

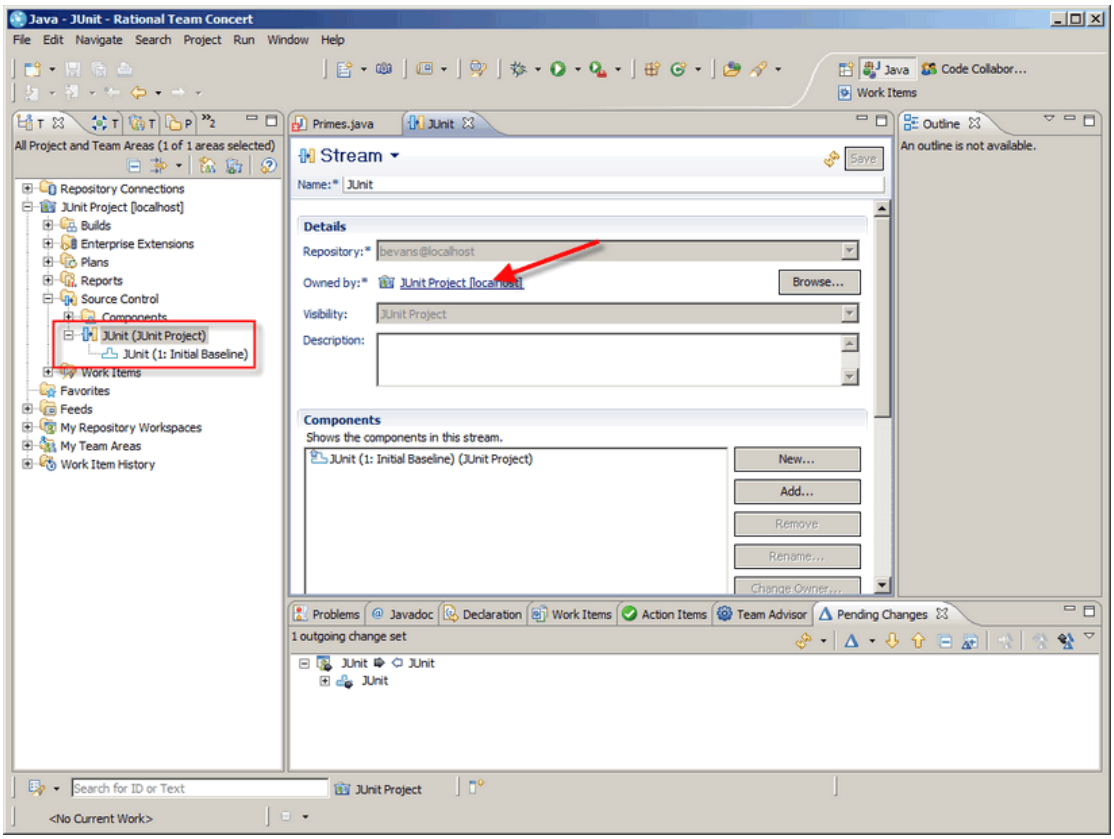
After configuring the plug-in, you can proceed with setting up the follow-up actions 722 in Rational Team Concert.

### 6.6.3.5 Setting and Configuring Follow-Up Actions

After you installed 717 the Collaborator plug-in for Rational Team Concert, you need to configure follow-up actions in Team Concert.

## Creating Follow-Up Actions

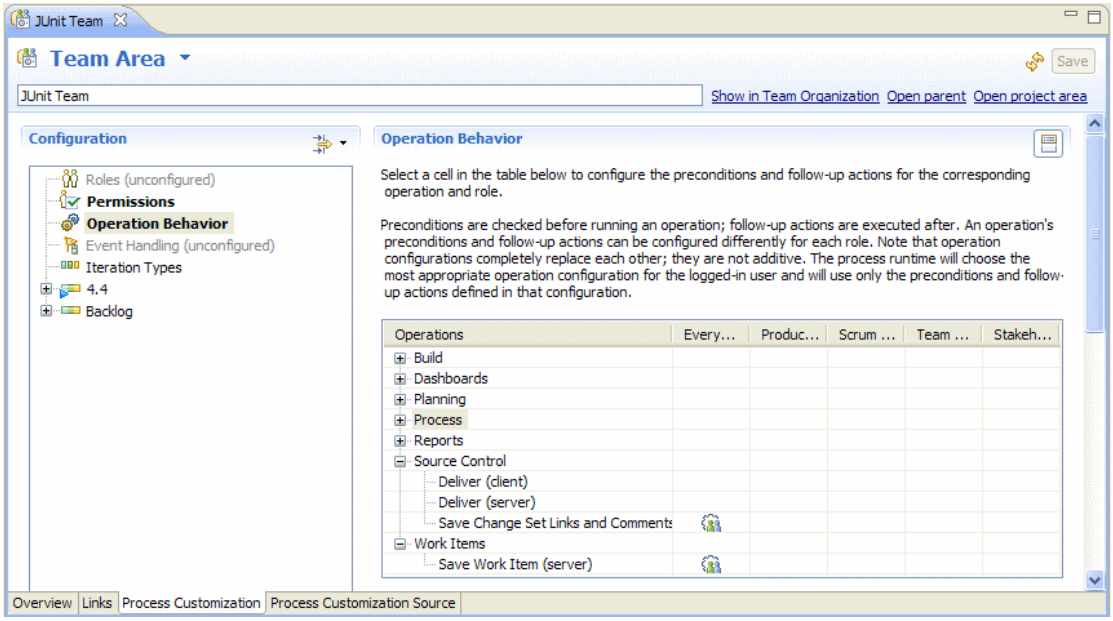
1. In the Rational Team Concert Client, open your project by clicking its name in the **Stream** section:




   This will open the Team Area view.

2. In the Team Area view, select **Operation Behavior** from the tree on the left and then go to **Source Control | Save Change Set Links and Comments** on the right and add a follow-up action **Update reviews for each linked work item** to it.

3. Select **Operation Behavior** from the tree on the left, then choose **Work Items | Save Work Item (server)** on the right, and add the follow-up **Upload changes from work item** to it:




# Action Parameters

## Update Reviews for Each Linked Work Item

This follow-up action creates a review associated to a work item when a new changeset is added to that work item.

The changeset must be new. If you add changes to some existing changeset associated with the work item, the review is not automatically updated. In this case, you need to change the work item's state and save the changes. If you have another follow-up action configured (see below) this will update the review.

The action's settings are stored in an XML file. You can change the action settings, and they will override the plug-in's settings 721. To do this, simply edit the XML file of the action. Below is a sample XML snippet with action settings:

```
<followup-action description="Collaborator - Update Reviews for each
linked Work Item" id="com.smartbear.collaborator.rtc.process.
ChangeSetModified" name="Collaborator - Update Reviews for each
linked Work Item">
 <CollaboratorURL value="http://localhost:8080"/>
 <CollaboratorProxyHost value="proxyhost"/>
 <CollaboratorProxyPort value="22"/>
 <CollaboratorUserID value="admin"/>
 <CollaboratorUserPassword value="123"/>
```

```
 <retries value="1"/>
 <retryDelayMs value="500"/>
</followup-action>
```

Here are some notes on the file contents:

- All child elements of the followup-action element are optional. If you do not specify some values, they are taken from the global plug-in settings.
- The `CollaboratorProxyPort` and `CollaboratorProxyPort` can have empty values to override the global plug-in settings.
- If you need to specify the user that will work for multiple Collaborator servers, create a user with the same name and password on all the needed Collaborator servers and then specify this user name and password in the global plug-in settings.
- The password is stored as plain text on the XML configuration.
- The action requests some information from the Team Concert repository. This information may be unavailable at the moment the action runs, so the action may need to wait for an answer.

  To prevent possible issues dues to these delays, use the `retries` and `retryDelayMs` values. The `retries` value specifies the number of times the follow-up action repeats its requests, and the `retryDelayMs` value indicates the number of milliseconds the action will wait for a response. Both settings accept only positive integer numbers as their values.

  Since the delay can depend on the Rational Team Concert server settings, there are no pre-defined value that will work on every server configuration. You can start using small numbers for the values (like 3 retries and 500 ms delay) and then increase or decrease these values until you find a configuration that works best for you.

### Upload Changes From Work Item

The Collaborator plug-in uses this follow-up action to track the changes of a work item state.

The action's settings are stored in an XML file. You can change the action settings, and they will override the plug-in's settings 721. To do this, simply edit the XML file of the action. Below is a sample XML snippet with action settings:

```
<followup-action id="com.smartbear.collaborator.rtc.process.
UploadToCollab">
 <CollaboratorURL value="http://localhost:8080"/>
 <CollaboratorProxyHost value="proxyhost"/>
 <CollaboratorProxyPort value="22"/>
 <CollaboratorUserID value="admin"/>
 <CollaboratorUserPassword value="123"/>
 <type value="Defect"/>
```

```
<type value="Enhancement"/>
<type value="Task"/>
<StateChangedTo value="New"/>
<OnStatus type="Story" state="Implemented"/>
<updateReviewers value="false"/>
<newReviewOnCancel value="false"/>
</followup-action>
```

Here are some notes on the file contents:

- All the elements are optional.
- The settings `CollaboratorURL`, `CollaboratorProxyHost`, `CollaboratorProxyPort`, `CollaboratorUserID`, and `CollaboratorUserPassword` override the plug-in settings' values 721.
- There are several elements that define the scope of work items, for which Collaborator will create reviews automatically:
  - `type` specifies the work item types for that Collaborator will create reviews. You can specify multiple `type` elements in the settings.
    You specify a type by its display name, not by its id. This name is case-sensitive.
  - `StateChangedTo` specifies the work item states that Collaborator will trace (Collaborator will create a review for a work item when this work item is switched to the specified state). You can use multiple `StateChangedTo` elements in the settings.
  - `OnStatus` combines functionality of both `type` and `StateChangedTo`. It specifies a couple of values (type and state) in one setting. These values can be types and states specified by the `type` and `StateChangeTo` elements, or any other types and states. That is, `OnStatus` elements is just one more way to define the conditions that trigger review creation. There can be multiple `OnStatus` elements in the settings.
- The Collaborator plug-in can trace changes in the Approvals list of a work item. When you add an entry of the Reviewer type to this list, the plug-in sends a command to the Collaborator server to update the Participants list of the review associated with that work item. See How Integration Features Work 704.

  The `updateReviewers` setting lets you enable or disable this functionality. If the value is `true` or not specified, the Collaborator will update the Participants list of a review. To disable this functionality, set the value to `false`.

- The `newReviewOnCancel` option specifies whether the follow-up action will create a new review for a work item, if its previous review was canceled or rejected. At that, all the changes from the work item changesets will be added to the new review. After the new review is created, the action replaces a link to the previous review with a new review link. The work item will have only one link to a review.

## Other Configuration Actions

In addition to configuring the Collaborator plug-in for Rational Team Concert and follow-up actions, you also need to configure the Collaborator server. For detailed information on this, see topics of the Configuring Servers and Plug-Ins 713 section.

# 6.6.4 Troubleshooting

This topic lists typical issues for the Collaborator integration with Team Concert and possible solutions for them.



## General Notes

In order for Collaborator and Rational Team Concert to integrate successfully, you need to configure the Rational Team Concert and Collaborator servers, as well as install and configure special plugins. If some feature is not working, then most likely some setting is invalid. Check the settings according to the information in the Configuring Servers and Plug-Ins 713 section.

## Issues

### Synchronizing the Review Participants List

For reviews that were created automatically, Collaborator synchronizes the work item's Approvals and the review's Participants lists (see How Integration Features Work 711). If the synchronization does not work, check the following:

- Look at the review phase. Collaborator does not synchronizes the lists if the review is in the Planning phase.

- The review must be created automatically. Synchronization does not work for the reviews that were created with the "Add to Review" command.
- Check if the user has the same names in Collaborator and Team Concert. Users with different names are not synchronized.
- Make sure the *Add reviewers* and *Remove reviewers* settings of the Collaborator server are enabled. See Configuring Collaborator Server 713.
- Check the connection between Collaborator and Team Concert. To do this, log in to Collaborator as administrator, go to **Admin > Version Control**, find your Rational Team Concert configuration there and click **Test Connection**.
- Check the user's roles in Collaborator and Team Concert.
  In Collaborator, the user should have a role that allows the user to finish reviews. By default, this is the *Reviewer* role, *Observers* are ignored.
  In Team Concert's, the user be added to the *Approvals* list as a Reviewer. *Approvers* are ignored.
- Check the **Client Configuration Mapping** settings of Collaborator and make sure the configuration matches the desired Team Concert server. See Configuring Collaborator Server 713.

## Automatic Review Creation and File Uploading

If Collaborator does not create reviews automatically when a work item's state changes, or if the files failed to upload to the review, check the following:

- Check the user names in Collaborator and Team Concert. They should be the same. Otherwise, errors will occur.
- Check if the Collaborator plugin for the Rational Team Concert server is installed and enabled. To do this, request the Provision Status page from the Team Concert server (`https://<your-team-concert-server>:port>/ccm/admin?internal=true#com.ibm.team.repository.admin.provisionStatus`) and search for *smartbear* on it. The page should say the Collaborator is installed and enabled. For information on installing and configuring the plugin, see Installing Collaborator Plug-In for Team Concert 717 and Configuring Collaborator Plug-in for Team Concert 721.
- Get the `ccm.log` file from the Team Concert server (see below). If the log says that follow-up actions of Collaborator cannot be found, set up and configure these actions. See Setting and Configuring Follow-Up Actions 722.
- If Collaborator does not create reviews, or creates them for unexpected item states, check the follow-up actions' settings.
- Check the work item categories in Team Concert. For example, if you are trying to create a review for work item that belongs to a team area, make sure this item does not use the project configuration.

- Get the log file from the Team Concert Client (see below). If the log reports about issues with class versions, make sure you installed the appropriate version of Collaborator's Eclipse plugin. If you use Rational Team Concert 3, you need to use the plugin from previous version of Collaborator. See a note in Installing Collaborator Plug-In for Rational Team Concert 717.

### Manual Review Creation

If you experience issues when creating reviews manually from the Team Concert client in Eclipse IDE, then most likely there are issues with the Collaborator's Eclipse plugin:

- Check the installation folder of your Team Concert client. If it is installed in the *<Program Files>* folder, this may cause problems for Eclipse. In this case, we recommend that your user account has administrator privileges and that you run the Client as administrator.
- Get the log file from the Team Concert Client (see below). If the log reports about issues with class versions, make sure you installed the appropriate version of Collaborator's Eclipse plugin. If you use Rational Team Concert 3, you need to use the plugin from previous version of Collaborator. See a note in Installing Collaborator Plug-In for Rational Team Concert 717.

## Getting Client and Server Logs

When some functionality does not work, you need to examine the Team Concert logs to understand what went wrong.

### To get the log of an Eclipse-based Team Concert client:

- In the Team Concert client, select **Help > About Rational Team Concert** from the main menu.
- In the subsequent About dialog box, click **Installation Details**.
- In the subsequent dialog, switch to the **Configuration** tab and click **View Error Log** there.

### To get the server log of Team Concert:

- On the server computer, go to the *<Jazz Team Server>*`/server/logs` folder.
- View the `ccm.log` file.

If the Team Concert server is installed on WebSphere, the log file will be in the `logs` directory of the Application server where Team Concert is installed.

## 6.7 Mercurial Integration

This section describes Collaborator integration with Mercurial:

### GUI Client 731

The GUI Client can upload arbitrary Mercurial diffs. The GUI Client can also upload local changes to files that are managed by Mercurial.

### Command-Line Client 734

The Command-Line Client can upload arbitrary Mercurial diffs and uncommitted changes to files that are managed by Mercurial.

### Supported Versions

Our integration uses your own installed Mercurial command line client executable to generate differences for review. We require hg v1.0 or later, as we use the -U flag to provide full context lines of differences so that the uploaded versions can contain the full file content of the previous and current file versions. The TortoiseHg client is not supported.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

### Technical Details and Limitations

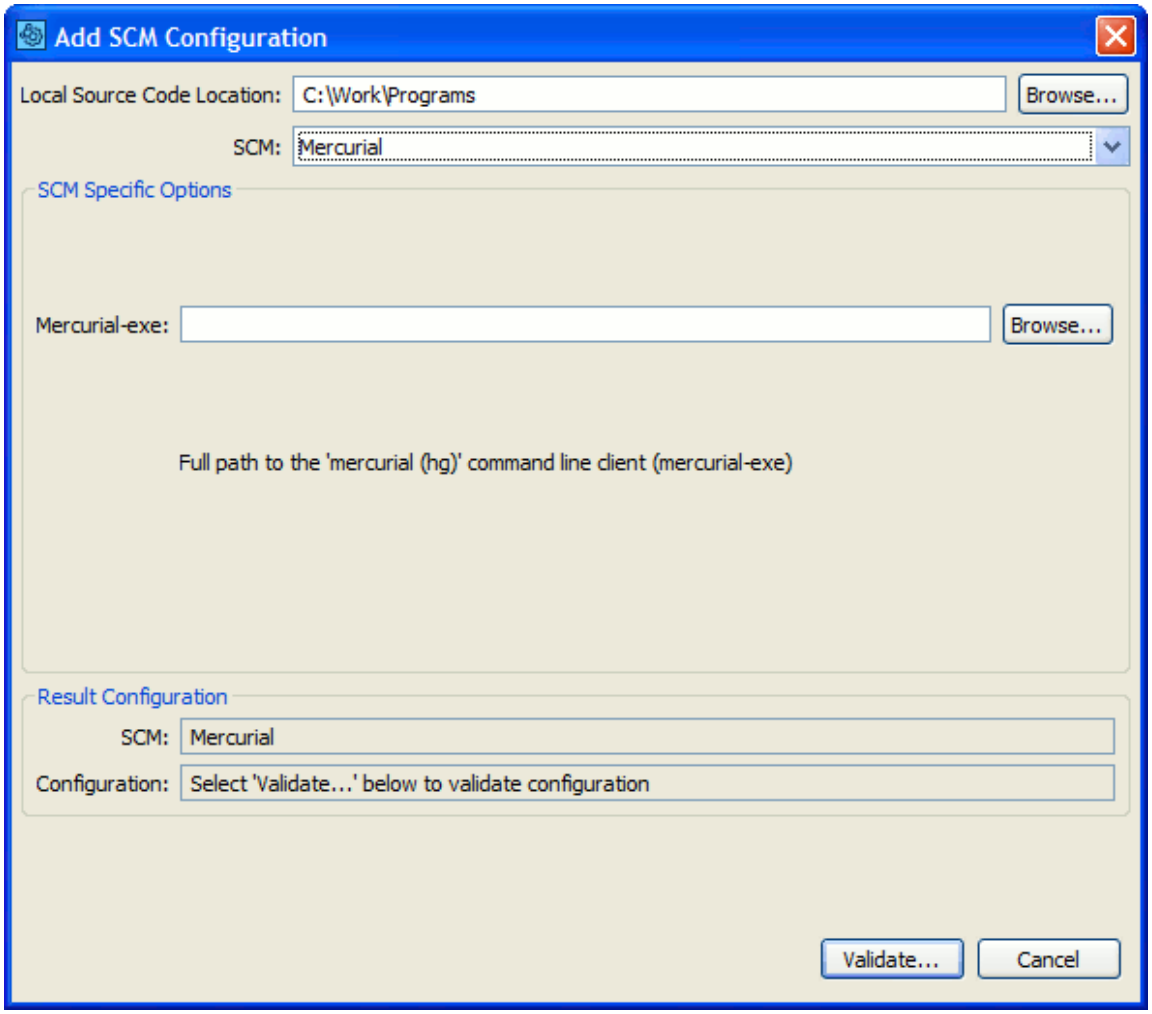
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you add several diffs (non atomic changelists) to the same review.

## 6.7.1 GUI Client

### Mercurial-specific Options

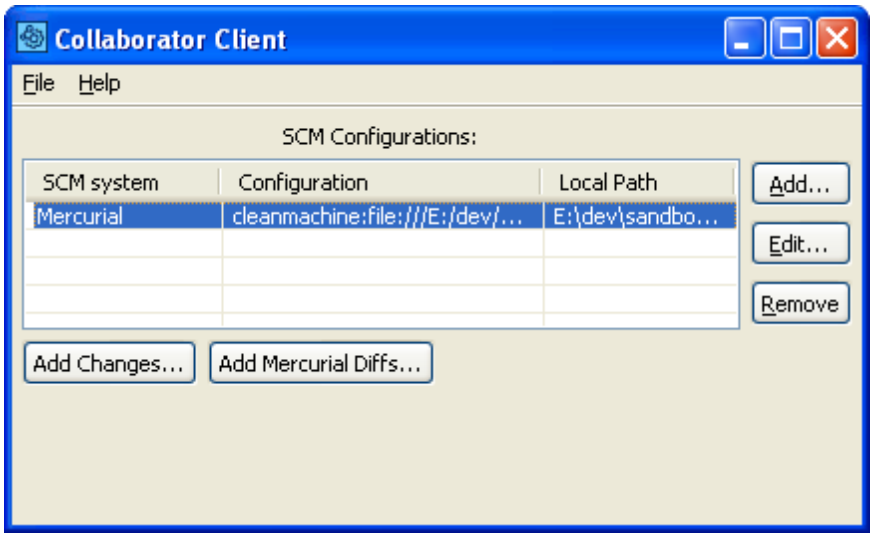
The SCM Configuration dialog[498] has one Mercurial-specific option. The full path to the Mercurial command line executable on your local file system should be specified if it is not already on your system path.




*Configure client to work with Mercurial*

### Uploading files to a Review

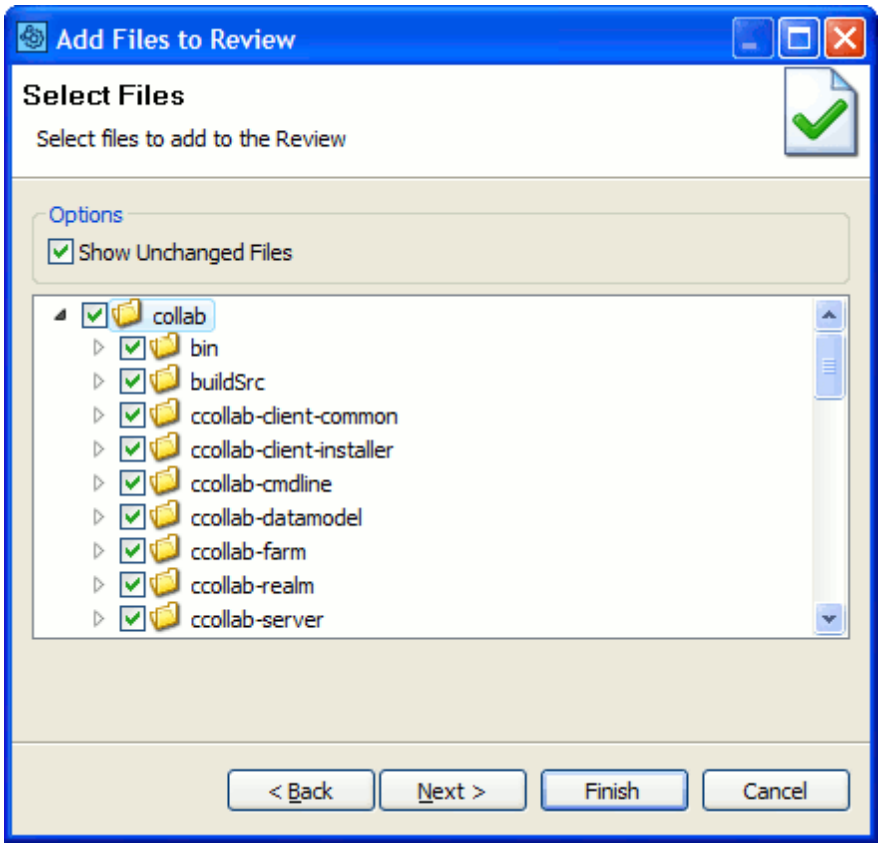
Selecting a Mercurial SCM Configuration in the GUI Client main screen[496] causes *Add to Review buttons* to appear.




*Uploading Mercurial files to a review*

## Add Changes

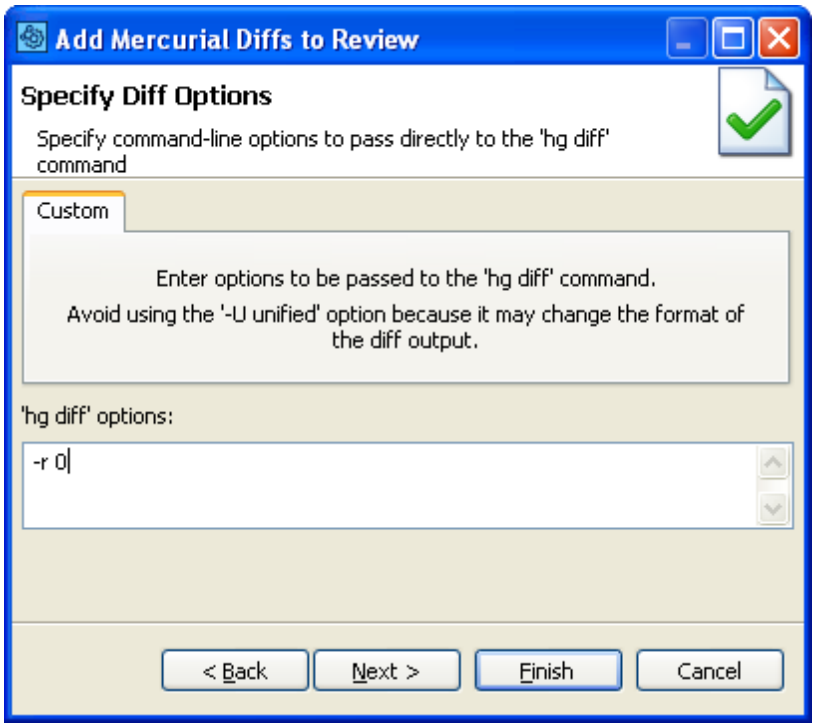
The Add Changes... button will be disabled if you have not specified a working copy in the Local Path field of the SCM Configuration dialog 498. Selecting Add Changes... allows you to upload your local modifications. These are the modifications that would be committed if you typed 'hg commit' from a command line.




*Add Changes*

## Add Mercurial Diffs

Press the *Add Mercurial Diffs...* button to upload arbitrary Mercurial diffs to the Collaborator Server for review. Note that if you did not specify a Local Source Code Location in the SCM Configuration dialog 498, then you will need to provide absolute paths in your diff options.




*Upload arbitrary Mercurial diffs*

### 6.7.2 Command-Line Client

## Commands recommended for Mercurial

`ccollab addchanges` 735 - Attaches locally-modified files to a review

`ccollab addhgdiffs` 736 - Uploads diffs generated by hg diff command

## Configuration

In most cases, the Command-Line Client can automatically detect your Mercurial configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Mercurial configuration or you want to override the detected settings, you can manually specify Mercurial settings using global options 517.

To manually configure the Command-Line Client to use Mercurial, execute the following command:

```
ccollab set scm 517 mercurial
```

## Mercurial-specific Options

| Option | Description |
|---|---|
| `--mercurial-exe <value>` | Full path to the 'mercurial (hg)' command line client |

#### 6.7.2.1 addchanges (for Mercurial)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by Mercurial to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review, |

| Option | Required? | Description |
|---|---|---|
| | | • `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br><br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

### 6.7.2.2 addhgdiffs

## Description

The `ccollab addhgdiffs` command uploads differences between arbitrary versions of files in Mercurial. The differences are generated using the native 'hg diff' command of Mercurial.

## Command Line Syntax:

```
ccollab [global-options] addhgdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files (defaults to command-line arguments) |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<user-diff-arg> [<user-diff-arg> ...]` | No | Command-line arguments to pass directly to the diff command |

## Remarks:

Do not use diff arguments that affect the diff output such as '-U unified'. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Mercurial treats Microsoft Office documents (.DOC, .DOCX, .XLS, .XLSX) and PDF files as binary files. Thus, the native 'hg diff' command is not able to generate meaningful difference for these types of files. Because of this the `ccollab addhgdiffs` command is also unable to upload differences between Microsoft Office documents (.DOC, .DOCX, .XLS, .XLSX) and PDF files.

### Examples:

To upload all changes between revisions 4 and 8:

```
ccollab addhgdiffs new -r 4 -r 8
```

To upload all changes in your local working directory:

```
ccollab addhgdiffs new .
```

## 6.8 Microsoft Team Foundation Server Integration

This section describes Collaborator integration with Team Foundation Server and its successor Azure DevOps Server.

### GUI Client 740

The GUI Client can upload local changes to files 742 in a Team Foundation Server Workspace. The GUI Client can also upload the files in a Shelveset 743 or Changeset 744.

### Command-Line Client 747

The Command-Line Client can upload local changes to files 748 controlled by Team Foundation Server before they are checked in. The Command-Line Client can also upload the files in a Shelveset 750 or Changeset 750.

### Supported Versions

Team Foundation Server and Azure DevOps Server versions 2010 through 2019 are supported by the Command-Line Client 747 and the GUI Client 740.

Our client integrations use Team Foundation Server SDK to communicate with Team Foundation Server. Because of this, we support all protocols, authentications, proxies, and other client configuration options that are supported by SDK.

### Remarks

Additionally to version control integration with Team Foundation Server, you can enable integration with TFS work items. This will allow synchronizing reviews and work items addressed in that reviews, appending the issues to the review's Remote System Links section and retrieving the item's current status. See Issue-Tracking Integrations: Overview 887 for more information.

The Team Foundation Server integration will not work with non-English installations of Visual Studio .NET. Regional settings for other locales are supported, but installing a non-English Visual Studio prevents correct parsing of the TF command line output.

When using Team Explorer Everywhere clients our client will need access to the server for changelist information without being prompted for a password. To do this you will need to save your login information to your credentials cache using the TF_AUTO_SAVE_CREDENTIALS environment variable, which is documented here. Once you have set this you will need to use a mutating command (such as checkout, checkin, and so on) and enter your credentials one more time for it to stick.

## Technical Details and Limitations

Review Screen, Diff Viewer, Eclipse Plug-in and Visual Studio Extension display atomic changelists (changesets in terms of TFS) in chronological order (from older to newer), regardless the order in which they have been uploaded to review.

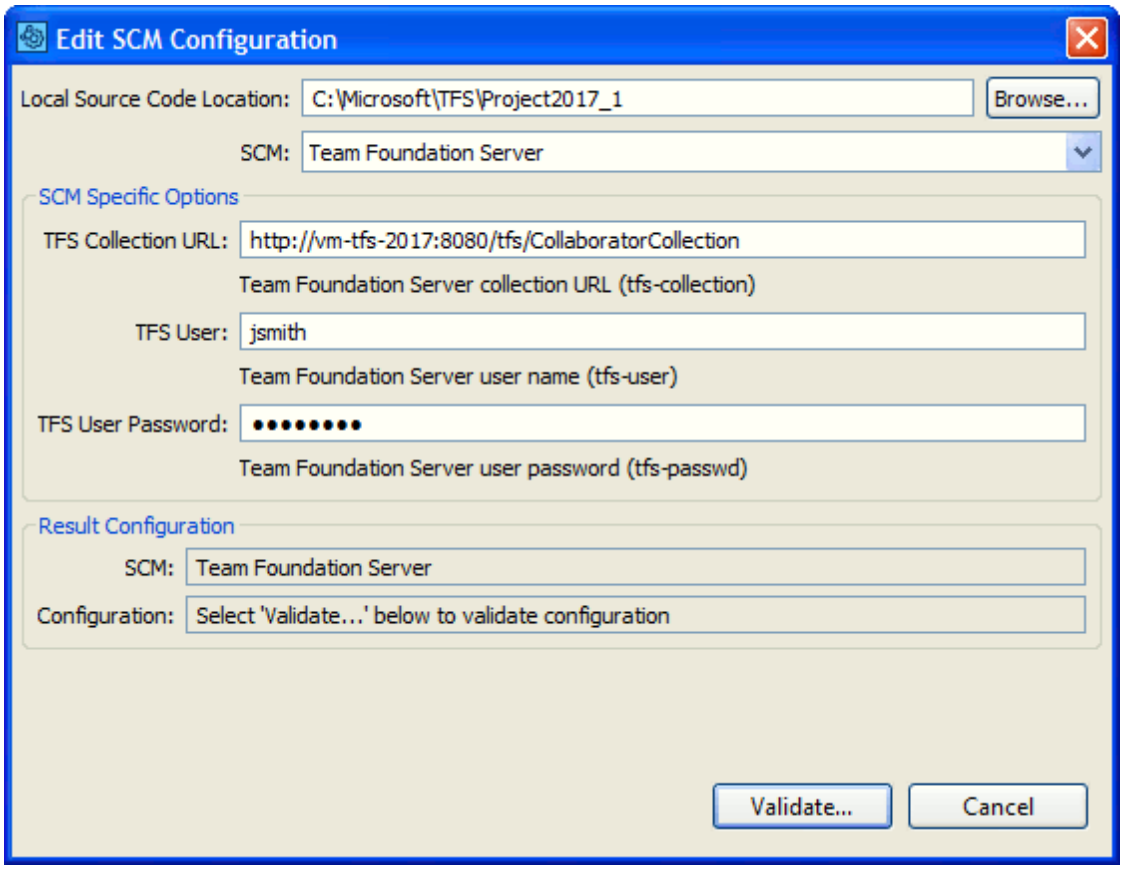
Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists (changesets in terms of TFS) to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- If you add pending changelists from different workspaces to the same review.
- If you add several diffs (non atomic changelists) to the same review.

### 6.8.1 GUI Client

## Team Foundation Specific Options

The SCM Configuration dialog[498] has several Team Foundation specific options.




*Team Foundation SCM Configuration*

Client configuration settings depend on whether you use a self-hosted version or a SaaS version of Team Foundation Server known as Visual Studio Team Services. If you use the latter, then you will need to set-up alternate authentication credentials at first and then use those credentials for Collaborator clients. See next section[741] for configuration instructions.

You will need to specify the following parameters:

**Local Source Code Location** — The local path to Team Foundation Workspace on your computer.

**TFS Collection URL** — For self-hosted version of Team Foundation Server, specify the URL of Team Foundation Project Collection to work with.

| | |
|---|---|
| | For instance: *http://tfs.acme.com/my-collection*<br><br>For SaaS version of Team Foundation Server, specify the URL of your Visual Studio Team Services account (without project or collection names).<br><br>For instance: *http://jsmith-ts.visualstudio.com* |
| **TFS User** | The name of Team Foundation user. For SaaS version of Team Foundation Server, specify alternate primary user name. |
| **TFS User Password** | The password of the user. For SaaS version of Team Foundation Server, specify alternate password. |

## Client Configuration for Visual Studio Team Services

If you use Visual Studio Team Services (a SaaS version of Team Foundation Server), then you will need to set-up alternate authentication credentials at first and then use these credentials for Collaborator clients.

### Enabling alternate authentication credentials

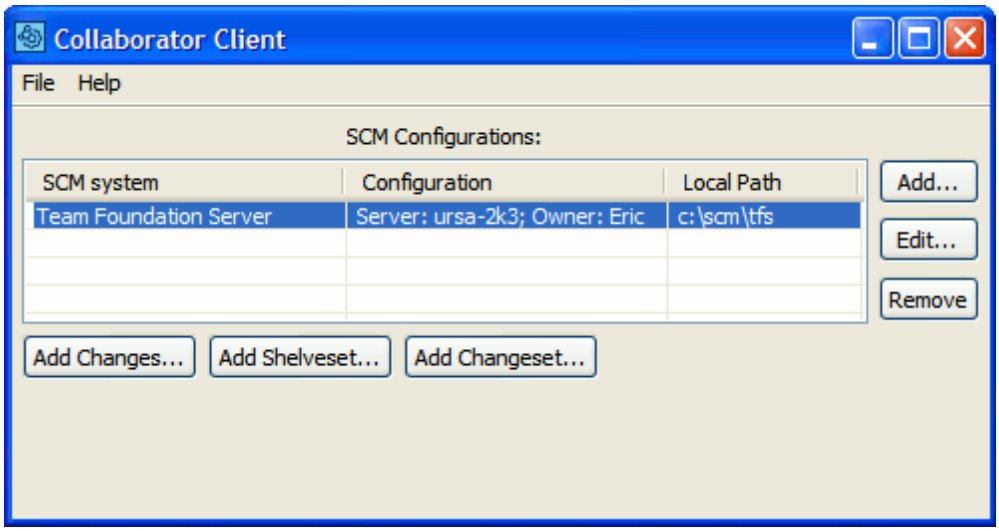
1. Login to your Visual Studio Team Services account (https://your-team-services-account-name.visualstudio.com),
2. Click on the Your Account icon in the top-right corner of the page,
3. Select Security,
4. Select Alternate authentication credentials,
5. Add new credentials. Collaborator clients use basic (user name and password) authorization.
6. Remember your primary user name and alternate credentials password.

### Configuring Collaborator client

1. Enter your Visual Studio Team Services account URL (https://your-team-services-account-name.visualstudio.com) into **TFS Collection URL** field. Do not include project name, collection name (or anything else) into the account URL.
2. Enter your alternate primary user name into **TFS User** field.
3. Enter your alternate credentials password into **TFS User Password** field.

## Uploading files to a Review

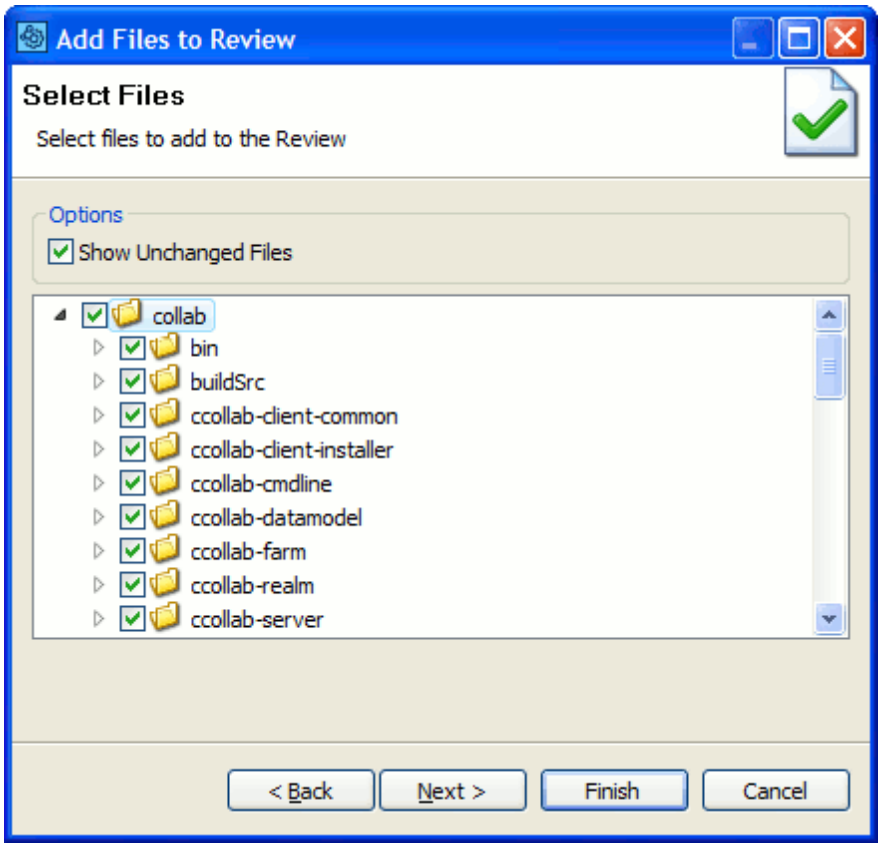
Selecting an Team Foundation SCM Configuration in the GUI Client main screen[496] causes three Add to Review buttons to appear. The *Add Changes...*[742] button uploads modified files in a Workspace. The *Add Shelvesets...*[743] button uploads the files in Shelvesets. The *Add Changesets...*[744] button uploads the files in committed Team Foundation Changesets.




*Uploading Team Foundation files to a Review*

## Add Changes

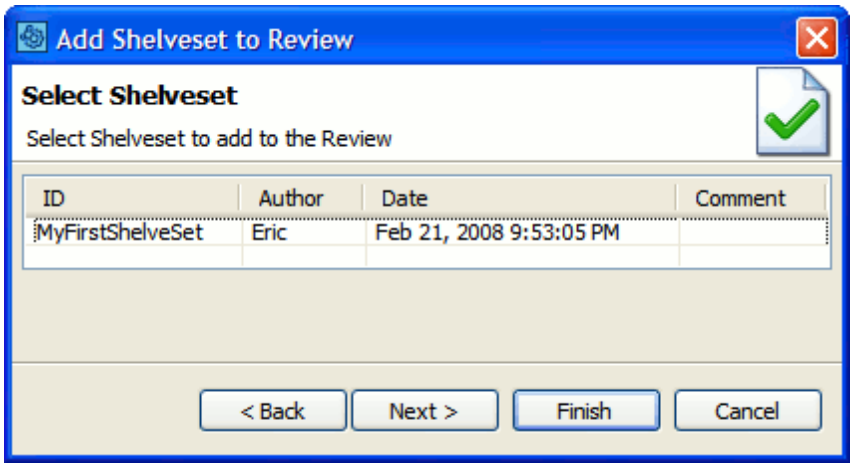
Press the *Add Changes...* button to upload modified files in a Workspace to the Collaborator Server for review.




*Add Changes*

## Add Shelvesets

Press the *Add Shelvesets...* button to upload the files in a Shelveset to the Collaborator Server for review. See the *Add Changesets...* button if you want to add a Shelveset to the review that's not in your list of shelvesets.

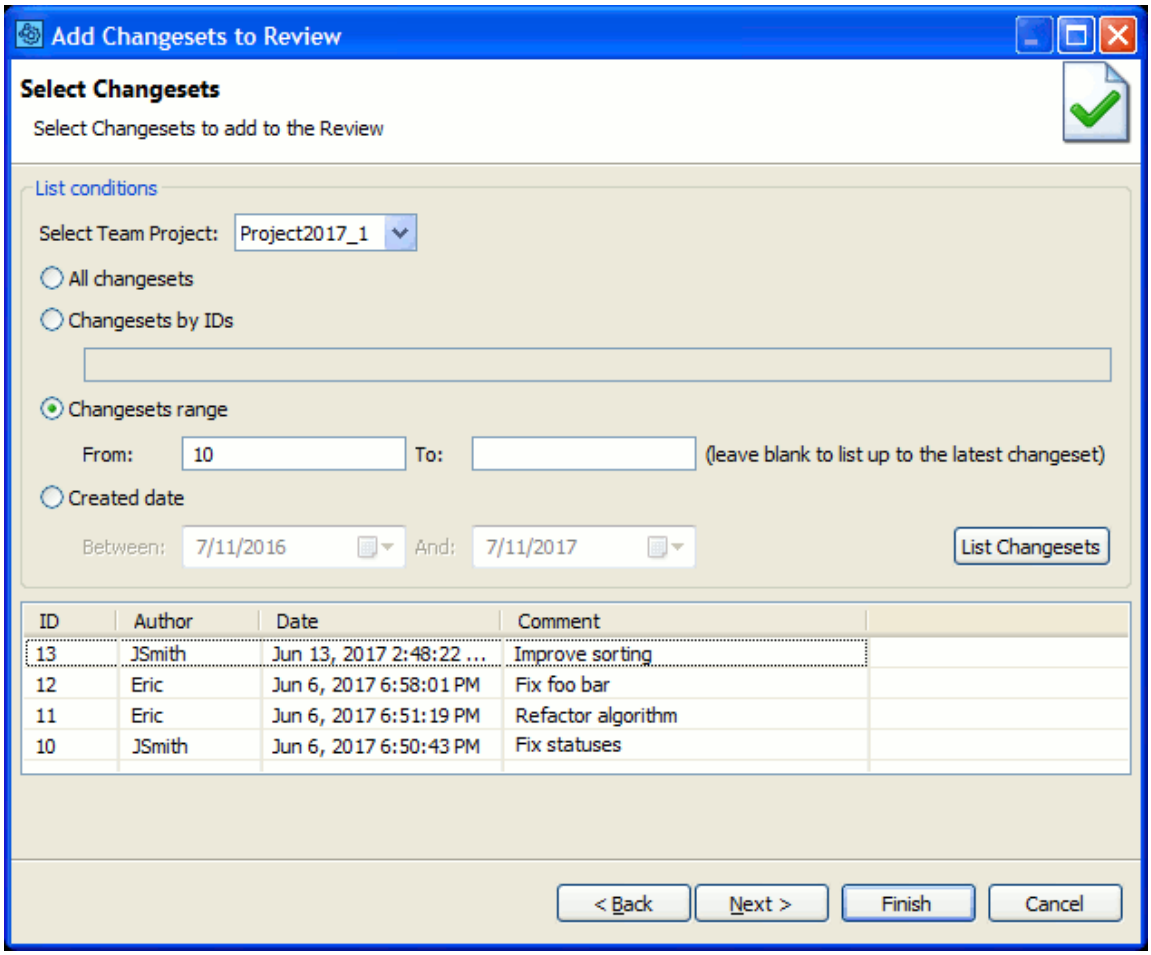



*Add Shelvesets*

## Add Changesets

Press the *Add Changesets...* button to upload the files in a Team Foundation Changeset to the Collaborator Server for review.

Select whether to display all available changesets, input ID's of particular changesets, list changelists within a specified range or within a specified date interval and press **List Changesets**. Then select the desired changesets in the table below and press **Next** to proceed.




*Add Changesets*

### 6.8.2 Gated Check-in Triggers

Collaborator provides special triggers that fire when someone submits files to a folder that is controlled by a gated check-in process in Team Foundation Server:

| Trigger | Description |
|---|---|
| `ensure-review-started` | Ensures that a review has been started (created in Collaborator) for the submitted changelist. |
| `ensure-completed` | Ensures that a review has been completed for the submitted changelist. |

These triggers ensure that files cannot be submitted unless a review for the respective changelist has been started (or completed). If the conditions are met, the commit will be allowed to continue normally.

The triggers work for Microsoft Team Foundation System 2018, 2017, 2015 and 2013. Currently, Update 1 and Update 1 Release 2 of Team Foundation System 2015 are not supported.

The triggers are implemented by dynamic-link libraries that you can download from the SmartBear web site as a .zip archive:

⇒ https://support.smartbear.com/downloads/collaborator/installers-for-collaborator-13/

## Installing Triggers

To install the triggers you will need to perform the following steps:

1) Download the .zip archive from the page mentioned above and unpack it to some folder on your computer.
The archive includes the *dlls* folder that have the dynamic-link libraries for the supported Visual Studio versions. Also, the archive includes a README file with instructions similar to those you can see below.

2) Open `<Team Foundation Server>\Application Tier\Web Services\web.config` file and append the following lines to the `<appSettings>` section:

```
<appSettings>
...
<!-- The following settings are for Collaborator triggers -->
<!-- Required setting -->
<add key="collab:serverUrl" value="http://yourcollabserver:port" />
<!-- Required setting -->
<add key="collab:userName" value="collabUser" />
<!-- Required setting. For an empty password, use value="" -->
```

```
<add key="collab:userPassword" value="collabPassword" />
<!-- Optional setting. Default is "started" -->
<add key="collab:triggerType" value="completed" />
...
</appSettings>
```

3) Copy the following DLLs from the unpacked *dlls\<Your-Visual-Studio-version>* folder to the `<Team Foundation Server>\Application Tier\Web Services\bin\Plugins` folder:

- `collab-ensure-review-started-or-completed.dll`
- `log4net.dll`

## Linking Reviews and Commits

To use the triggers, you must require developers to put the review ID somewhere in the TFS commit message. The review ID must be specified in the **Review:** *NNN* format, where *NNN* stands for review number. Here is an example:

```
Commit message example:
TFS: Add support for gated check-ins (Review: 12345)
```

Collaborator uses the `(@"Review:*\s?(\d+)")` regular expression to extract the review id from the commit information. This text can appear inline with other text or in a more formal "form-style" layout.

## Troubleshooting

If you run across any issues or problems, it will help SmartBear technical support if you send in a debugging log. To capture a debugging log:

1) Open `<Team Foundation Server>\Application Tier\Web Services\web.config` file and append the following lines to the `<configSections>` section:

```
<configSections>
...
<!-- The following setting is for debugging Collaborator triggers -->
<section name="log4net" type="log4net.Config.
Log4NetConfigurationSectionHandler, log4net"/>
...
</configSections>
```

2) Also append the the following section to the `web.config` file:

```
<log4net debug="true">
 <appender name="LogFileAppender" type="log4net.Appender.
FileAppender">
 <param name="File" value="C:/temp/collab_TFS_GatedCheckIn.log" />
 <param name="AppendToFile" value="true" />
 <layout type="log4net.Layout.PatternLayout">
 <param name="Header" value="[Header]\r\n" />
 <param name="Footer" value="[Footer]\r\n" />
 <param name="ConversionPattern" value="%d [%t] %-5p %c %m%n" />
 </layout>
 </appender>
 <root>
 <level value="DEBUG" />
 <appender-ref ref="LogFileAppender" />
 </root>
</log4net>
```

The above settings will generate the debugging log file at specified path.

3) Replace the DLLs in the Plugins folder with their fresh copy. This will force TFS to re-read the web.config file.

You may adjust the log output data for your needs by changing the above settings according to log4net convention of configuration files.

### 6.8.3 Command-Line Client

## Commands recommended for Team Foundation Server

`ccollab addchanges` 748 - Attaches locally-modified files to a review

`ccollab addchangelist` 750 - Attaches an atomic changelist to a review

`ccollab commit` 751 - Commit changes in the review

The addchanges[748] command will upload local changes to files controlled by Team Foundation before they are checked in.

The addchangelist[750] command will upload a the files in a Shelveset or Changeset.

## Configuration

In most cases, the Command-Line Client can automatically detect your Team Foundation configuration. Try testing your configuration[507] to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Team Foundation configuration or you want to override the detected settings, you can manually specify Team Foundation settings using global options[516].

To manually configure the Command-Line Client to use Team Foundation, execute the following command:

```
ccollab set scm[517] tfs
```

## Team Foundation Server-specific Options

Client configuration settings depend on whether you use a self-hosted version or a SaaS version of Team Foundation Server known as Visual Studio Team Services. If you use the latter, then you will need to set-up alternate authentication credentials at first and then use these credentials for Collaborator clients. See configuration instructions[741].

| Option | Description |
|---|---|
| `--tfs-collection <value>` | For self-hosted version of Team Foundation Server, specify the URL of Team Foundation Project Collection to work with.<br><br>For SaaS version of Team Foundation Server, specify the URL of your Visual Studio Team Services account (without project or collection names). |
| `--tfs-user <value>` | The name of Team Foundation user. For SaaS version of Team Foundation Server, specify alternate primary user name. |
| `--tfs-passwd <value>` | The password of the user. For SaaS version of Team Foundation Server, specify alternate password. |

### 6.8.3.1 addchanges (for Team Foundation Server)

## Description

The `ccollab addchanges` command uploads locally modified files controlled by Team Foundation Server to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `global-options` | No | A number of global or SCM-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-spec> [<file-spec> ...]` | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on. |

| Option | Required? | Description |
|---|---|---|
| | | After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

### 6.8.3.2 addchangelist (for Team Foundation Server)

## Description

The `ccollab addchangelist` command attaches all files from Team Foundation Server shelvesets or changesets to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or AccuRev-specific global options. See Command-line Global Options Reference 516. |

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ... ]` | Yes | Identifier(s) of the desired changeset(s) or shelveset(s) in your source control. |

The first argument is the review specifier, subsequent arguments are the IDs of the Shelvesets or Changesets to upload.

Note that the changesets are searched first - if a changeset is found, it will be added to the review. To avoid any naming conflicts, always use at least one non-numeric character in your shelveset names.

## Examples:

To upload Shelvesets MyShelveset and todays_work to a new review:

```
ccollab addchangelist new MyShelveset todays_work
```

To upload Changesets C3 and C12654 to review 111:

```
ccollab addchangelist 111 3 12654
```

#### 6.8.3.3 commit (for Team Foundation Server)

## Description

The `ccollab commit` command submits the changes from a pre-commit review to source control. Be sure to include a relevant comment.

### Command Line Syntax:

```
ccollab [global-options] commit [--comment <value>] [--dismiss-only]
[--force] <review>
```

### Command Options

| Option | Required? | Description |
|---|---|---|
| `--comment <value>` | No | Comment for reviewed changes |
| `--dismiss-only` | No | Just dismiss the Action Item |
| `--force` | No | Ignore potential problems |
| `<review>` | Yes | Identifier of the desired review (an integer number), or an `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

### Example:

```
ccollab commit 25 --comment "my code" --force
```

## 6.9 PTC Integrity Integration

Both GUI and command-line clients of Collaborator integrate with PTC Integrity:

- GUI Client 754

  The GUI Client can find and upload changes by pending or committed PTC Change Packages, by specific file version number, or by modified working files in local sandboxes.

- Command-Line Client 758

The Command-Line Client can find and upload changes by pending or committed PTC Change Packages, or by modified working files in local sandboxes.

## Supported PTC Integrity Versions

Collaborator uses Java client API that communicates with the PTC client (PTC Source) installed on your system. Collaborator supports the client versions starting from MKS Source 2007 and later (MKS Source and MKS Integrity are former names of PTC Source and PTC Integrity). MKS 2006 clients with the latest fix packs are also supported.

Since Collaborator works with a client applications that is installed on your computer, it supports all the protocols, authentication types, proxies, and other configuration options that you are using.

## Windows Installation

On Windows operating systems, Collaborator's integration modules require 32-bit Java Runtime Environment. 64-bit JRE is not supported at the moment.

## Linux Installation

To integrate with PTC Source on Linux, Collaborator's Java client API requires the following environment variables to be set:

```
export PATH=[integrity-client-install-dir]/bin:$PATH
export LD_LIBRARY_PATH=[integrity-client-install-dir]/lib/
linux:$LD_LIBRARY_PATH
```

## Notes

If your PTC projects use variant sub-projects, then you may need to set the `-Dcom.smartbear.ptc.forceprojectpath=true` VM option[1240] on Collaborator clients to make the integration use project path instead of configpath.

## Technical Details and Limitations

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you add several diffs (non atomic changelists) to the same review.

## 6.9.1 GUI Client

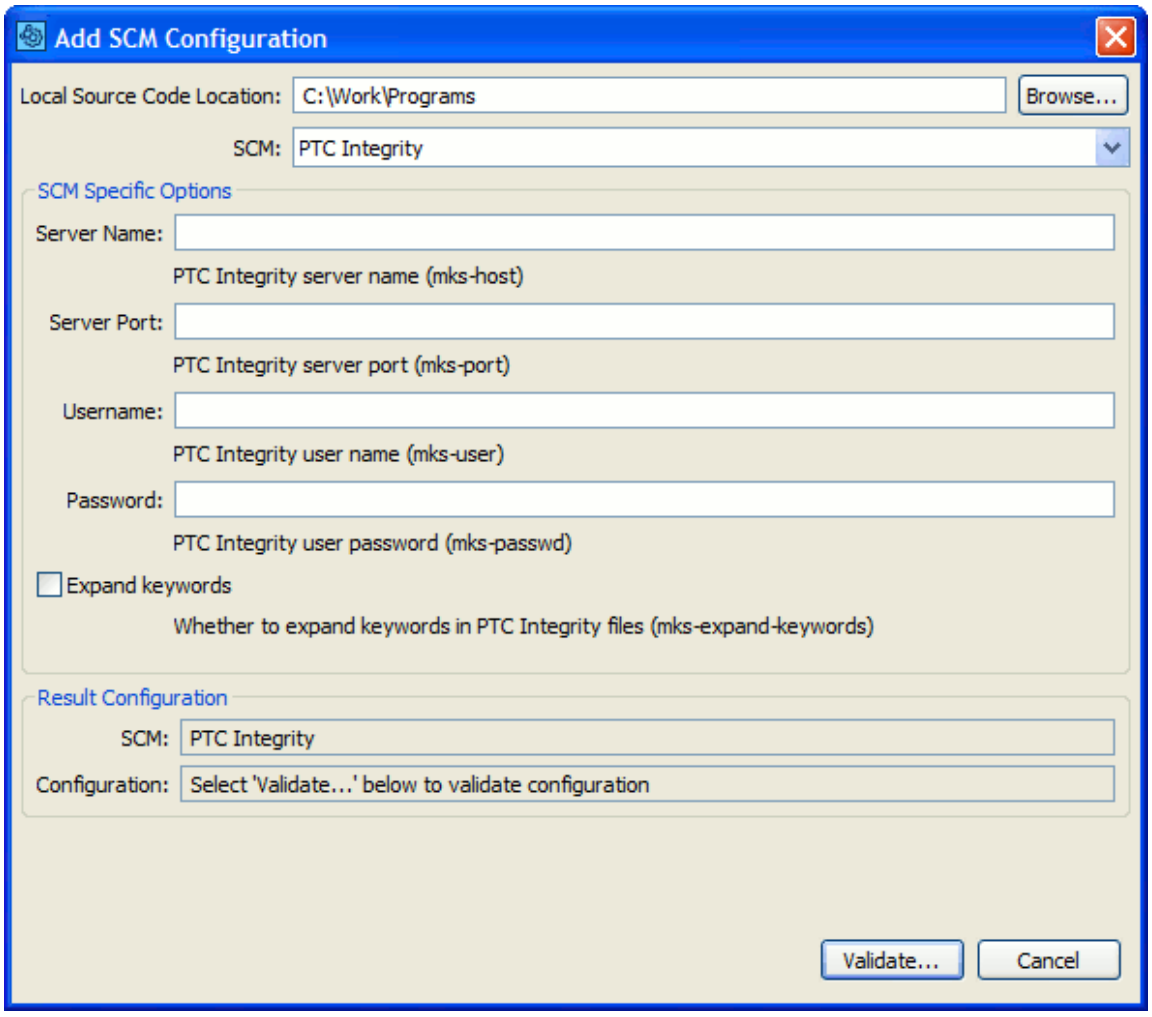
### Setting PTC Integration Options

In order for Collaborator to be able to work with PTC Integrity, you need to specify connection settings in Collaborator's GUI Client.

To do this:

1. Launch the GUI Client.

2. Click **Add** on the main screen.
   The Add SCM Configuration 498 dialog will appear:




*PTC Source Control Settings*

3. In the dialog:

   - Select **PTC Integrity** in the **SCM** drop-down list.

- Specify the desired PTC Integrity server (host), port, user name and password.

  We would like to remind that there is no need to specify some or all of these settings, if your PTC Integrity client automatically connects to the PTC Integrity server using default local settings.

  **Known Issue:** If your PTC Integrity server uses IPv6 connection, GUI Client cannot map the server host name. To workaround the issue, you will need to add the IPv6 address of your server to the hosts file on your machine. The hosts file resides at "/etc/environment/hosts" on Unix and at "%SystemRoot%\System32\drivers\etc\hosts" on Windows.

After you close the dialog, your PTC Integrity configuration will be added to the SCM Configurations list on the main window of the GUI Client.

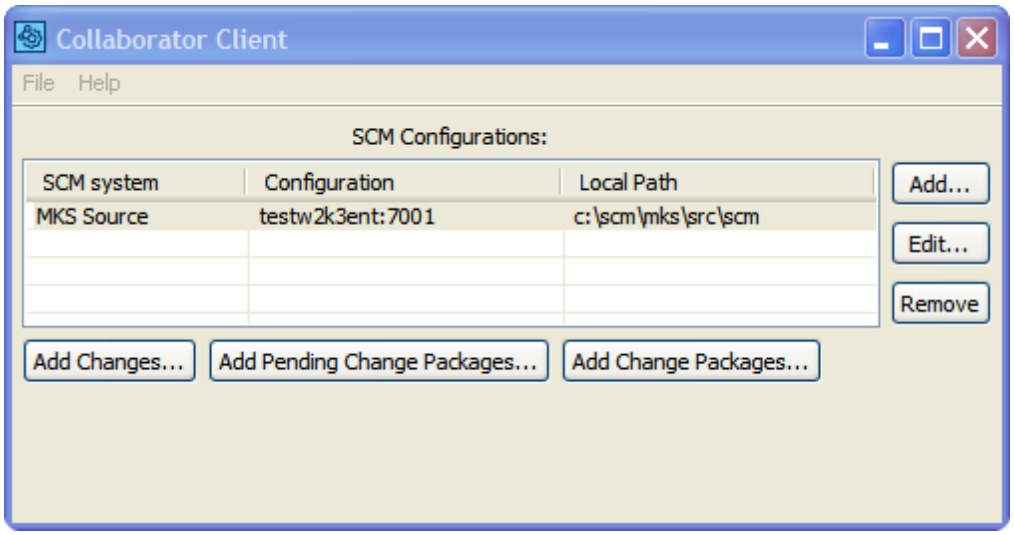
## Adding Files to Review

To add files to a review:

1. In the **SCM Configurations** list on the GUI Client's main screen, select your PTC Integrity configuration.

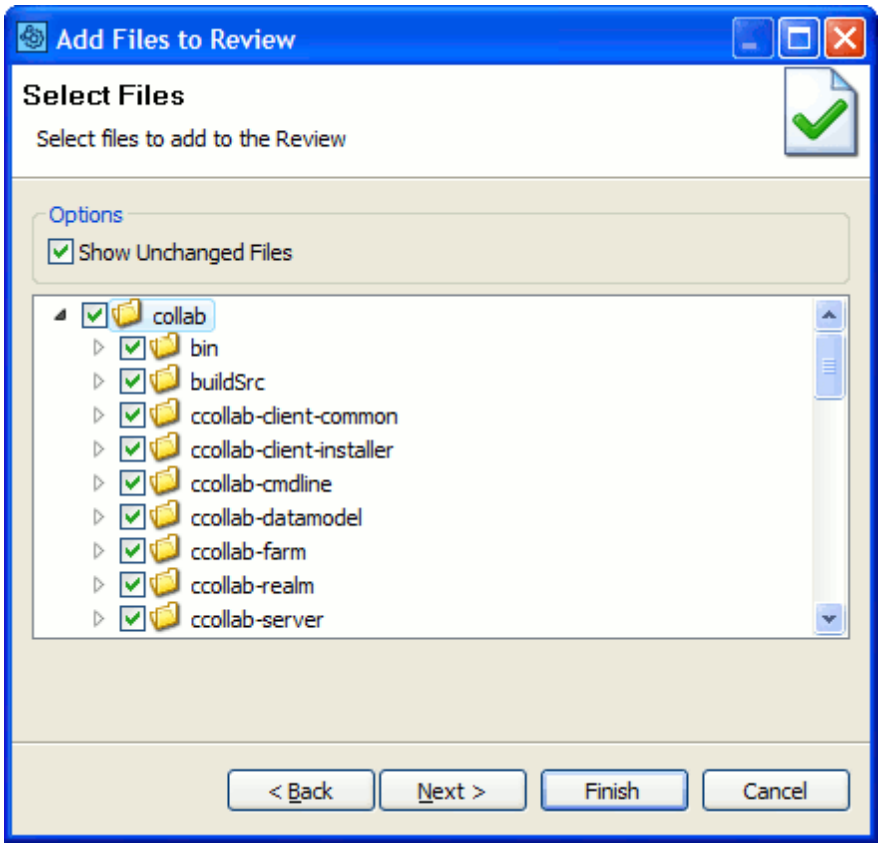
   This will enable several "Add" buttons below the list.

2. Click these buttons to append files to a review on the Collaborator server.

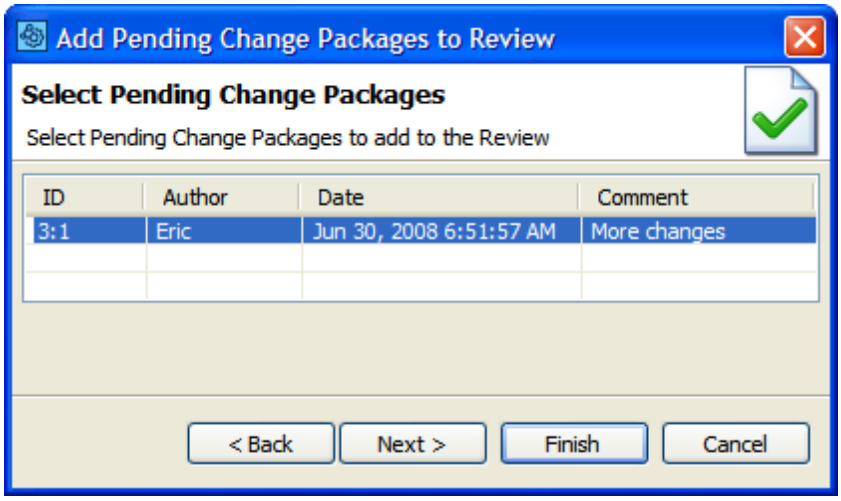


## What Buttons Do

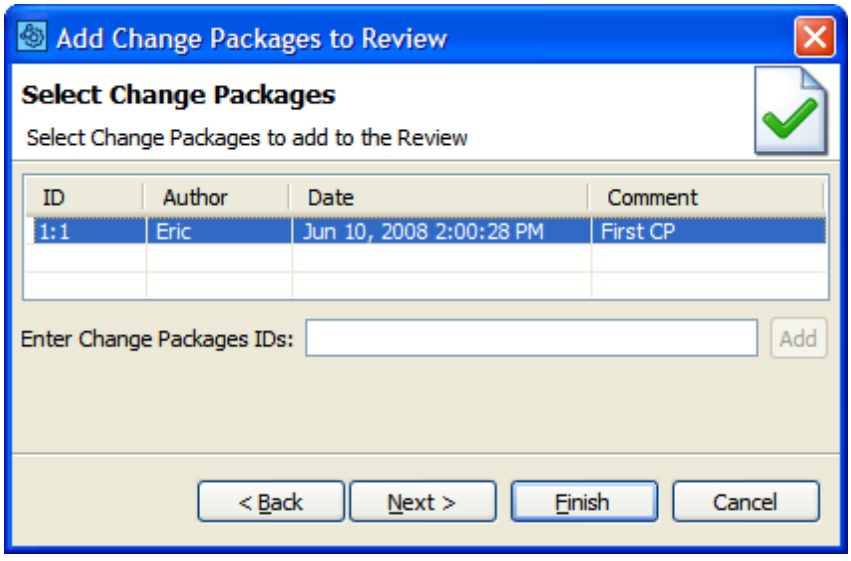
- **Add Changes** - Appends modified files from local PTC sandboxes. After you click the button, you can choose the files to be added:



- **Add Pending Change Packages** - Append all the files from a *pending* change package.
  Click the button and select the change package you need:




- **Add Change Packages** - Appends files from any change package from the user's sandbox to a review.
  Click the button and select the desired change package:




## Note on Adding Change Packages

Adding a pending or committed change package to a review generates file differences based on Multiple Version Changelists 970.

### 6.9.2 Command-Line Client

To work with PTC source clients, you can use the following Collaborator's:

- `ccollab addchanges`
- `ccollab addchangelist`
- `ccollab addversions`

## Configuration

In most cases, the Command-Line Client can automatically detect your PTC configuration and connection settings. Try testing your configuration[507] to check whether the settings are detected correctly.

If the Command-Line Client is unable to detect your PTC configuration, or if you want to override the detected settings, you can manually specify PTC settings using global PTC settings[516].

To manually configure the Command-Line Client to use PTC, run the following command-line:

```
ccollab set scm[517] mks
```

## Global PTC-Specific Command-Line Options

| Option | Description |
|---|---|
| `--mks-host <value>` | PTC Integrity server name. |
| `--mks-port <value>` | PTC Integrity server port. |
| `--mks-user <value>` | PTC Integrity user name. |
| `--mks-passwd <value>` | PTC Integrity user password. |
| `--mks-expand-keywords` | Specifies whether to expand keywords in source files. |

The arguments use the *--mks* prefix as MKS Integrity is a former name of PTC Integrity.

#### 6.9.2.1 addchanges (PTC Integrity Integration)

## Description

The `ccollab addchanges` command appends files that are controlled by PTC Integrity client (PTC Source) and that are located on your computer to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| global-options | No | A number of global or PTC-specific global options. See Command-line Global Options Reference 516. |
| --upload-comment <value> | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| <review> | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <file-spec> [<file-spec> ...] | Yes | Files to be added and/or folders to scan for modified files. |

| Option | Required? | Description |
|---|---|---|
| | | Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br><br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

The following command line creates a new review and add all changes in the current folder and its subfolders, plus the *foo.txt* file, to the new review:

```
ccollab addchanges new . foo.txt
```

The following command line uploads modified files from the current working folder and its subfolders to review with the identifier 157:

```
ccollab addchanges 157 .
```

The following command uploads the *foo.txt* file and modified files from the *c:\dev\project* folder to the review, to which you uploaded files last time:

```
ccollab addchanges last foo.txt c:\dev\project
```

#### 6.9.2.2 addchangelist (PTC Integrity Integrtion)

## Description

The `ccollab addchangelist` command attaches all files from a change package controlled by PTC Integrity client (PTC Source) to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or PTC-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ... ]` | Yes | Identifier(s) of the desired changeset(s) in your source control. |

## Examples:

The following command uploads the change packages 1:2 and 3:2 to a new review:

```
ccollab addchangelist new 1:2 3:2
```

The following command adds the change packages 1:1 and 3:1 to the review with the identifier 182:

```
ccollab addchangelist 182 1:1 3:1
```

#### 6.9.2.3 addversions (PTC Integrity Integration)

## Description

The `ccollab addversions` command appends the specified versions (revisions) of a file controlled by the PTC Integrity client (PTC Source) on your computer to a review.

## Command Line Syntax:

```
ccollab [global-options] addversions [--upload-comment <value>] [--
version-spec <value> [<value> ...]] <review> [<file-path>]
[<version>] [<predecessor-version>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or PTC-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `--version-spec <value> [<value> ...]` | No | The version to be added to a review. A version-spec value consist of three components:<br><br>*path version* [*previous-version*],<br><br>where *path* is the file name or server path of the file, *version* is the file version to be reviewed, and *previous-version* is an optional version, against which *version* should be compared.<br><br>If any of these arguments contains spaces, enclose it in quotes. |

| Option | Required? | Description |
|---|---|---|
| | | Typically a version-spec is not used in the command line. We recommend specifying the file and version using the <file-path>, <version> and the <predecessor-version> arguments (see below). |
| <review> | Yes | Identifier of the desired review (an integer number), or a new, ask, or last keyword. Where keywords define the following behaviour:<br>• new - the command will create a new review,<br>• ask - the command will pause execution and prompt for the identifier of the desired review,<br>• last - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <file-path> | No | The path of the file whose versions are to be added to the review. If filename is omitted, entire directory will be added.<br>**Important:** If you use this option, you should also specify <version> (see below). |
| <version> | No | Required, if <file-path> is specified.<br>The version (revision) of the file to be added to the review. You can specify the keyword *local* to tell the command to use the local version of the file. |
| <predecessor-version> | No | Preceding file version to be added to the review. If you skip this argument, Collaborator will attempt to determine the preceding version based on the information from the PTC Integrity source control. |

## Examples:

The following command line adds versions (revisions) 1.95 and 1.88 of the hello.c file to the review 861:

```
ccollab addversions 861 ./hello.c 1.95 1.88
```

The following command adds these revisions to a new review:

```
ccollab addversions new ./hello.c 1.95 1.88
```

### Remarks

- If you skip the predecessor version, Collaborator will generate diffs using the predecessor version reported by your source control system.
- By default, the command lets you add versions of one file only. To add versions of multiple files, create a text file and specify this file in the command line as the standard input stream (stdin):

  ```
  ccollab addversions last < versionlist.txt
  ```

  Each line in the file must consist of the following components: *path version* [*predecessor-version* ].

  For information on them, see description of the `version-spec` arguments.
- If you skip the file name and versions in the command line, the command will expect to read them from the standard input stream (stdin). Below are some examples for reading versions from the standard input:

  ```
  ccollab addversions 86753
  ccollab addversions last < versionlist.txt
  cat versionlist.txt | ccollab addversions new
  ```
- When specifying the version in the command line or in an input file, you can use the keyword *local* to denote the version corresponding to the local version of the file. The *local* keyword can only be used for the first version argument, not for the predecessor version.

## 6.10 Perforce Integration

This section describes Collaborator integration with Perforce:

### Perforce Server Integration 767

The Collaborator server can pull submitted changelists directly from your Perforce server for review, without users needing to install any client programs. It can also enforce file content access permissions 768 (protections) configured in Perforce.

## GUI Client 769

The GUI Client can upload Changelists into Collaborator. You can upload Pending 771 or Submitted 772 Changelists, but you cannot upload the Default Changelist. The GUI Client can also upload arbitrary Perforce diffs 772, files in a Branch 773, or the difference between two Labels 774 or dates 775.

## Command-Line Client 776

The Command-Line Client can upload Changelists into Command-Line Client. You can upload Changelists Pending or Submitted Changelists, but you cannot upload the Default Changelist. The Command-Line Client can also upload arbitrary Perforce diffs.

## Eclipse Plug-in

The Eclipse Plug-in 525 can upload 545 Perforce changelists. Just right-click on any changelist entry in Eclipse, either before or after it is submitted.

## P4V and P4Win

Collaborator comes with plug-ins to P4V and P4Win 786 that let you right-click on any changelist (pending or submitted) and add to a new or existing review.

## Perforce Server Triggers

Perforce server-side triggers can ensure code is reviewed 788. There is also a trigger to automatically update a changelist 789 with information about the review of that changelist.

## Perforce Changelist Renumbering

Perforce nearly always renumbers changelists upon submission, but when you are doing pre-commit reviews Collaborator always has the pre-submit number. We have included a script you can customize that will ask Perforce what a changelist's number was before it was submitted, and update changelists in Collaborator to have the new number. This requires a Collaborator client and server **5.0.5005** or better, and a Perforce client and server **2007.3** or later. This script is intended to be run periodically via cron or a similar task scheduler. Perforce Changelist Renumbering Script

## Supported Versions

Our integration uses your own Perforce command-line client (p4) to communicate with the server. We support all client and server versions later than **2002.1**. Our Eclipse Plug-in supports Perforce's eclipse plugin version **2009.2.234487** or later.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using. This includes configuration from environment variables, `$P4CONFIG` files, `p4 set` registry values, and so forth.

## Support for Branch / Integrate

Collaborator fully supports Perforce's file branching and integration semantics.

If files in a changelist are marked for branching, they are not considered "changed". The file content itself is not changed; only the file paths are being changed.

If files in a changelist are being integrated, this works just like a regular change. Many customers choose to review integrations especially carefully since the changes might interact in unexpected ways.

## Support for Copy / Move

Collaborator fully supports Perforce's file copy/move semantics. Thus, if a file is copied or moved rather than added from scratch, it will show up that way in the various user interfaces.

## Support for Directory-level New/Delete/Copy/Move

Collaborator partially supports Perforce's concept of directories (not just files) being altered.

All files underneath the directories in question will be scanned, uploaded and represented properly in the GUI. The directories themselves will *not* be shown in any GUI.

## Support for Shelvesets

Collaborator supports adding Perforce Shelvesets to reviews.

## Additional Information About Mark for Delete

If you want to use "Mark for Delete" on a file and add this file to the newly created Collaborator review, synchronize this file with the depot. Otherwise, this file can be absent from your Perforce workspace and Collaborator will return an exception.

## Technical Details and Limitations

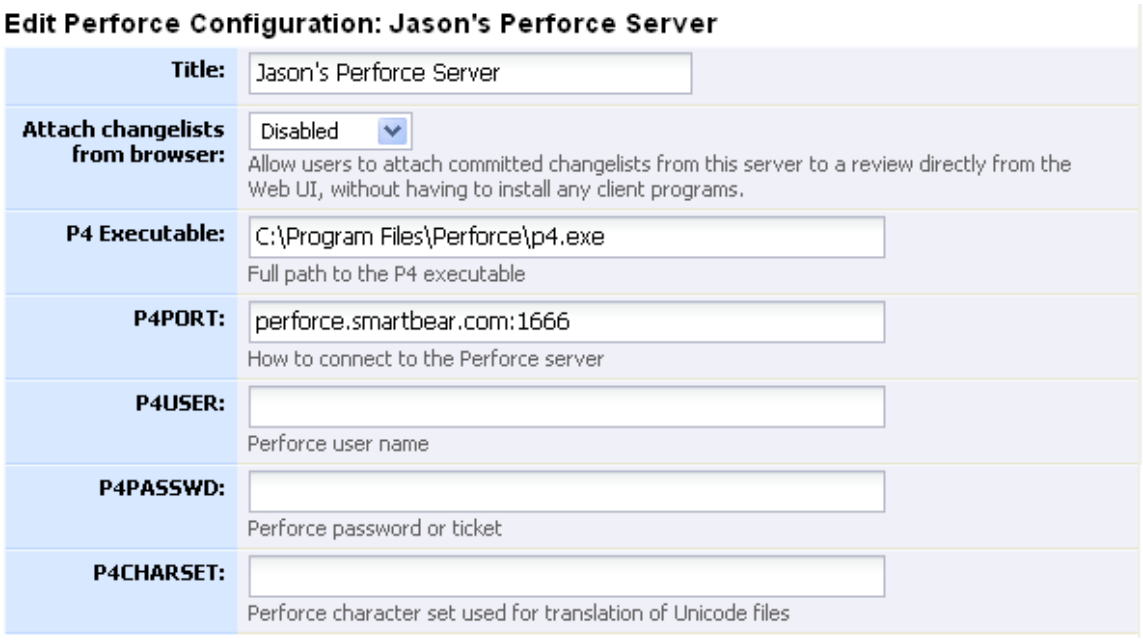
Review Screen, Diff Viewer, Eclipse Plug-in and Visual Studio Extension display atomic changelists in chronological order (from older to newer), regardless the order in which they have been uploaded to review.

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- If you add pending changelists from different workspaces to the same review.
- If you add several diffs (non atomic changelists) to the same review.

### 6.10.1 Perforce Server Integration

The Collaborator server can be configured to communicate directly with your Perforce server. This allows users to review submitted changelists completely from the browser, without having to install any client programs. To enable this feature, first install and configure a Perforce client on the Collaborator server and then create an entry for your Perforce server in the Version Control 204 tab of the administration interface. Version control server entries are also created automatically if one of the client programs uploads files from a server that does not match any of the currently configured servers.




**Title** — The title is displayed to users, so it should be something that everyone will understand, even if they are going through proxies, VPNs, or other such things. When a version control server entry is created automatically, this is filled in with the server address (p4port) of the server.

| | |
|---|---|
| **Attach changelists from browser** | If enabled, this feature lets users select submitted changelists to review directly from the web browser, without having to install any client programs. |
| **P4 Executable** | Full path to the P4 executable |
| **P4PORT** | Address to use to connect to the server. When a version control server entry is created automatically, this is filled in with the server address obtained from the client. |
| **P4USER** | Perforce user name |
| **P4PASSWD** | Perforce password or ticket |
| **P4CHARSET** | Perforce character set used for translation of Unicode files |

## Perforce Protections

The Collaborator server can be configured to check Perforce protections with the Perforce server whenever a user tries to access a file.


When enabled, Collaborator will check access permission with your Perforce server whenever a user tries to access the content of a file managed by this server. Note that this assumes that the user's Perforce username is the same as their Collaborator login.

To check Perforce protections the configured P4USER[768] must have Perforce "super" permission. If you do not want to provide an account with that level of permission, you can instead configure a script for Collaborator to run instead of calling `'p4 -s protects'` directly.

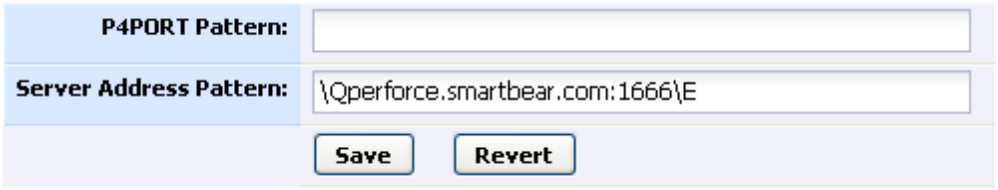
The script will be passed arguments `"-p <p4port> -u <user> -h <host> <depotPath>"`. For example, this script could be implemented as a Windows batch file:

```
@p4 -u admin -s -p %2 protects -u %4 -h %6 %7
```

The script must produce *exactly* the same stdin, stdout, and process error code as if Collaborator ran "`p4 -s protects`" directly. Also make sure the script runs as fast as possible, because it will be invoked every time any user tries to view the contents a file from this server.

### Client Configuration Mapping

You can supply Java-style regular expressions to map changelists uploaded from our client tools to this Perforce server. It is important to set up these regular expressions so that files uploaded by the various Collaborator client tools are correctly associated with this server-side Perforce configuration.



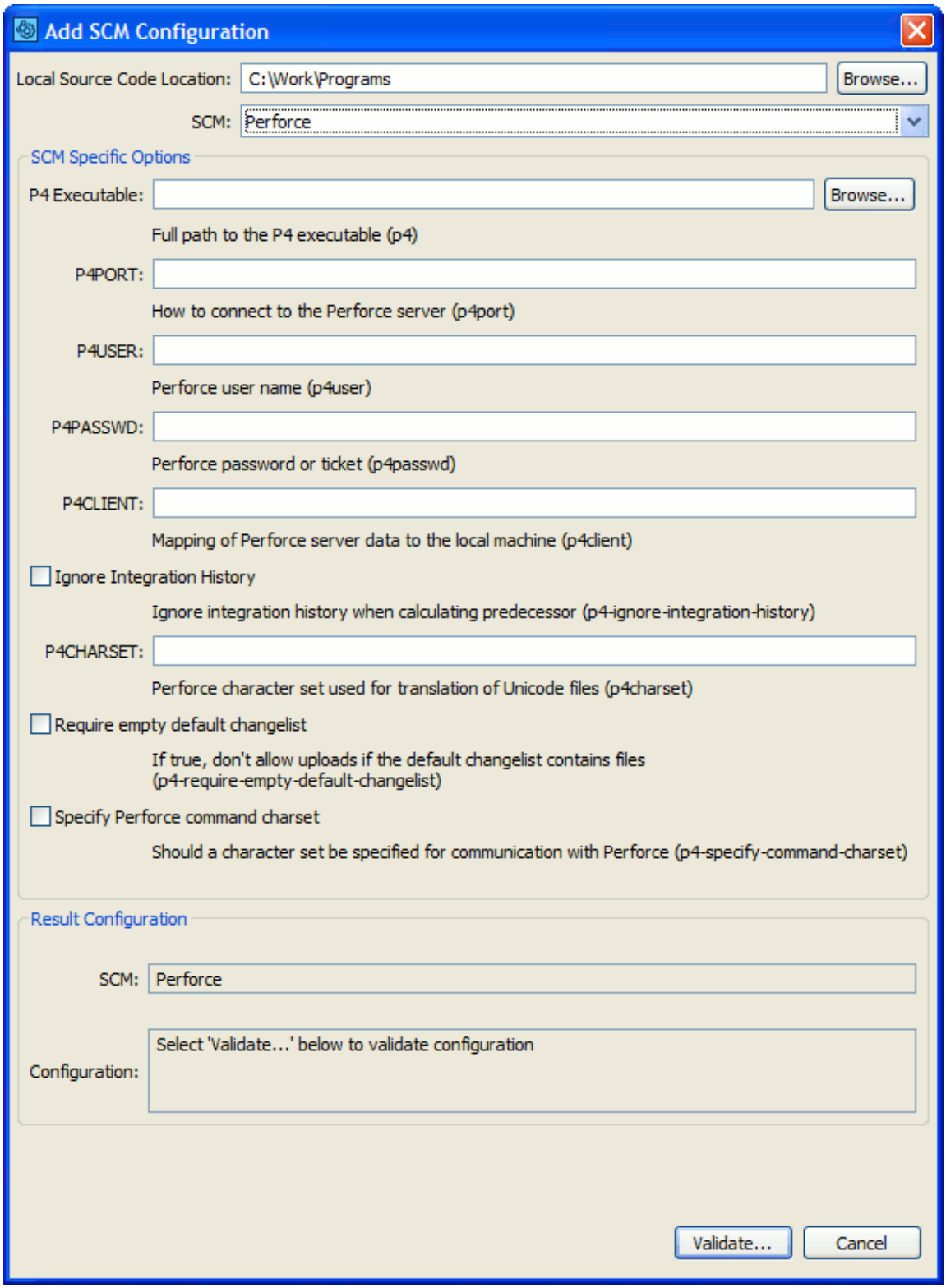

**P4PORT Pattern** — Match on the client's configured P4PORT. Using the Server Address Pattern 769 is better, but clients before 5.0.5009 did not upload "Server Address", so you can configure this as a fallback.

**Server Address Pattern** — Match on the "Server Address" returned from running "`p4 -p <p4port> info`" on the client. This is generally more reliable than matching on the P4PORT, and works through Perforce proxy servers and other complications. When a version control server entry is created automatically, this is filled in with the server address obtained from the client.

## 6.10.2 GUI Client

### Perforce-specific Options

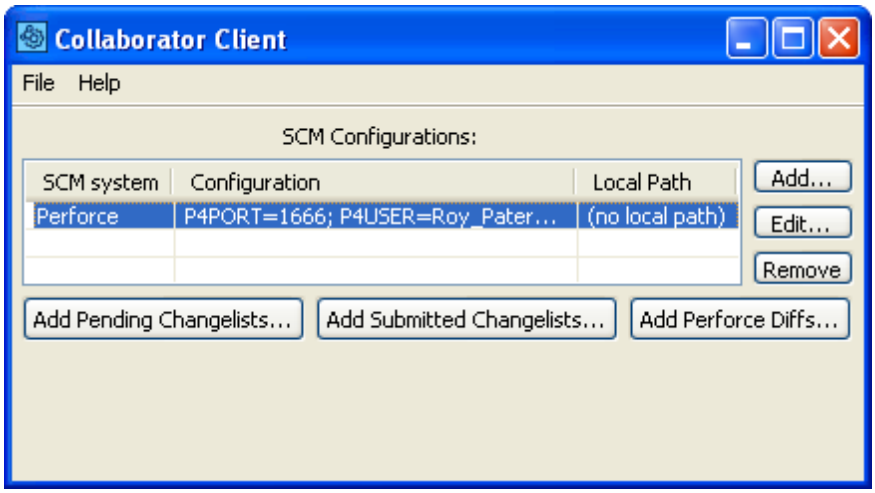
The SCM Configuration dialog 498 has several Perforce-specific options. These can be set as necessary to override Perforce options derived from the environment.




*Perforce SCM Configuration*

## Uploading files to a Review

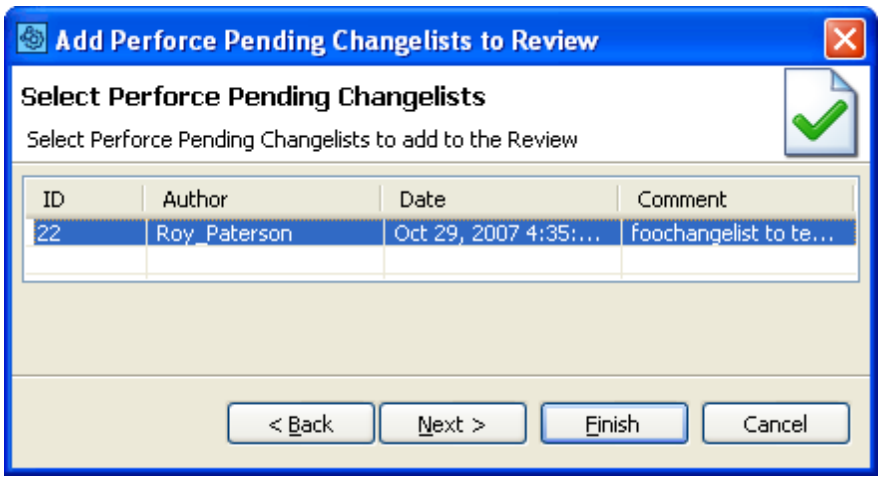
Selecting a Perforce SCM Configuration in the GUI Client main screen[496] causes several Add to Review buttons to appear. The *Add Pending Changelists...*[771] button uploads numbered, pending changelists. The *Add Submitted Changelists...*[772] button uploads submitted changelists. The *Add Perforce Diffs...*[772] button uploads arbitrary diffs, compares files in a Branch[773], or compares the difference between two Labels[774] or dates[775].




*Uploading Perforce files to a Review*

## Add Perforce Pending Changelists

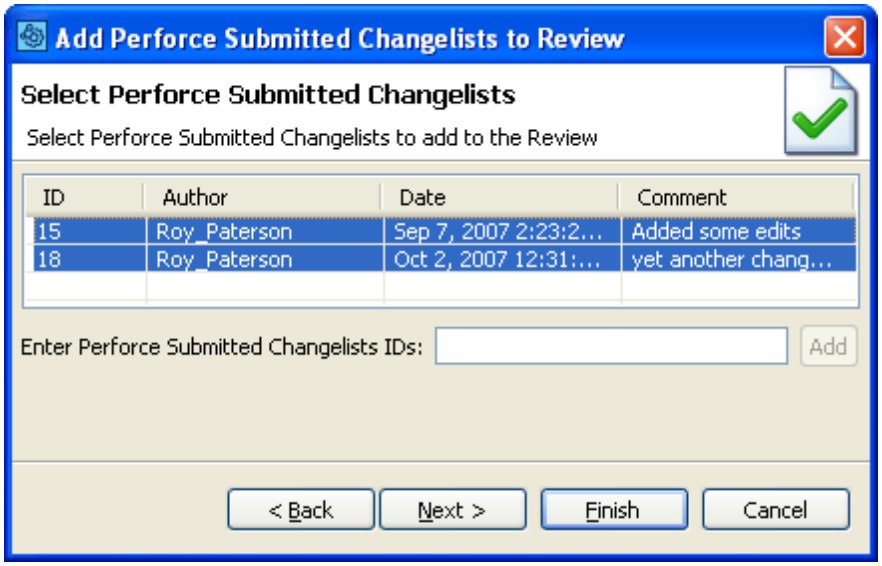
Press the *Add Perforce Pending Changelists...* button to upload the files in a Perforce pending changelist to the Collaborator Server for review. You cannot upload the default changelist.




*Add Perforce Pending Changelists*

## Add Perforce Submitted Changelists

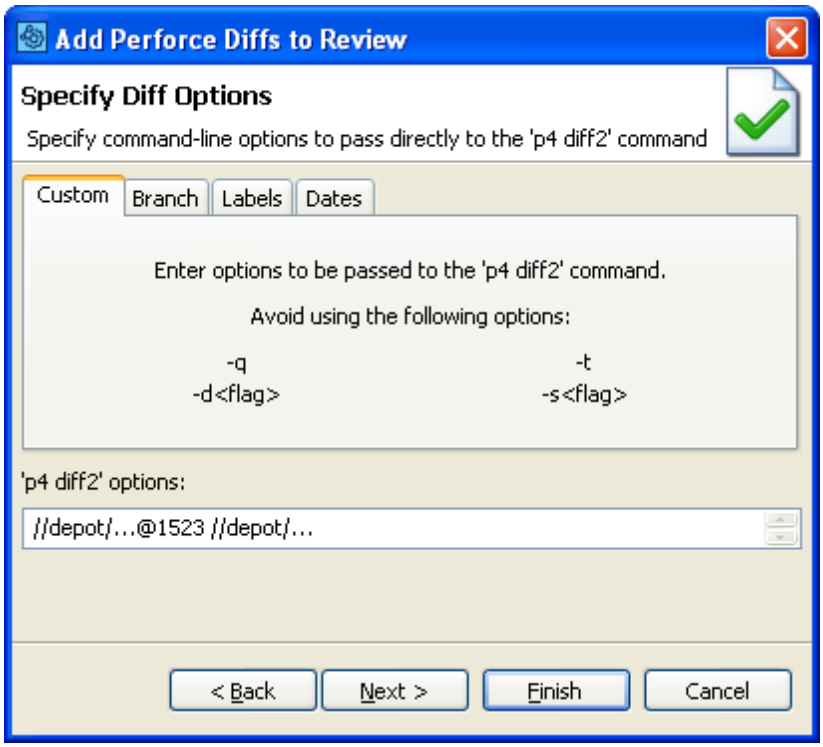
Press the *Add Perforce Submitted Changelists...* button to upload the files in a Perforce submitted changelist to the Collaborator Server for review.




*Add Perforce Submitted Changelists*

## Add Perforce Diffs

Press the *Add Perforce Diffs...* button to upload arbitrary Perforce diffs to the Collaborator Server for review.

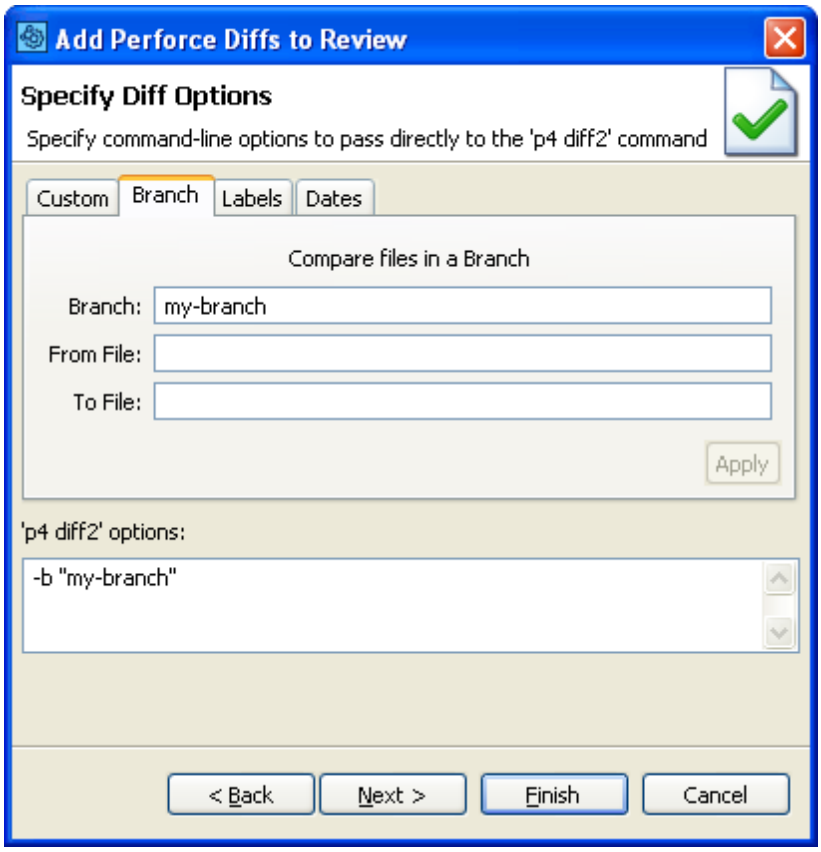



*Add Perforce Diffs*

You can enter arbitrary Perforce diff options, compare files in a Branch[773], or compare the difference between two Labels[774] or dates[775].

### 6.10.2.1 Comparing files in a Branch

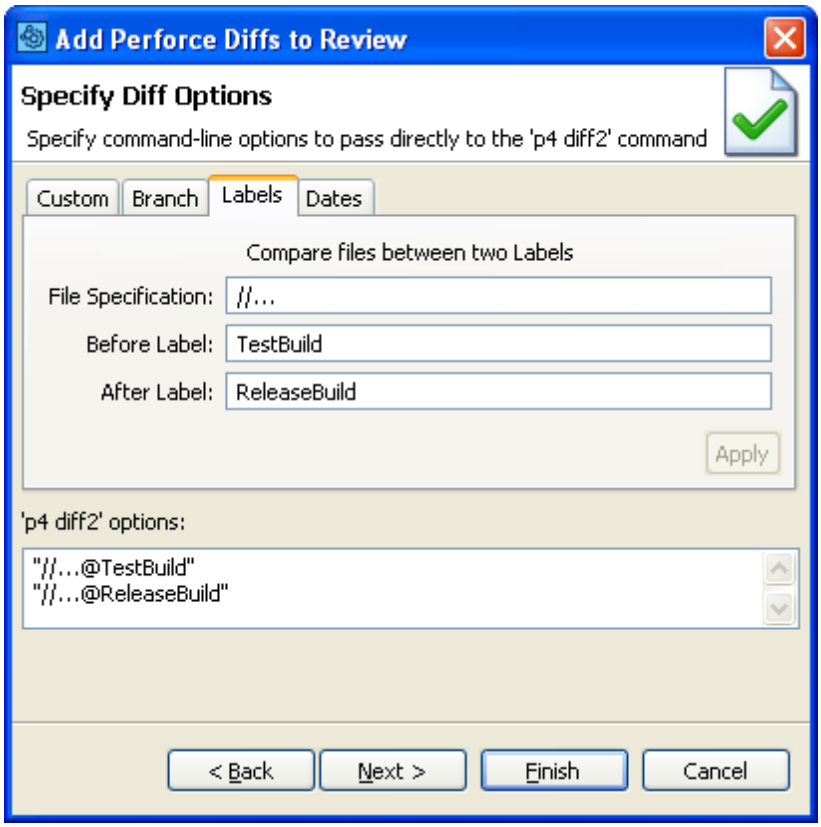
Press the *Add Perforce Diffs...*[772] button on the main screen[496] and then select the *Branch* tab to upload the files in a Branch.




*Upload the files in a Branch*

### 6.10.2.2 Comparing two Labels

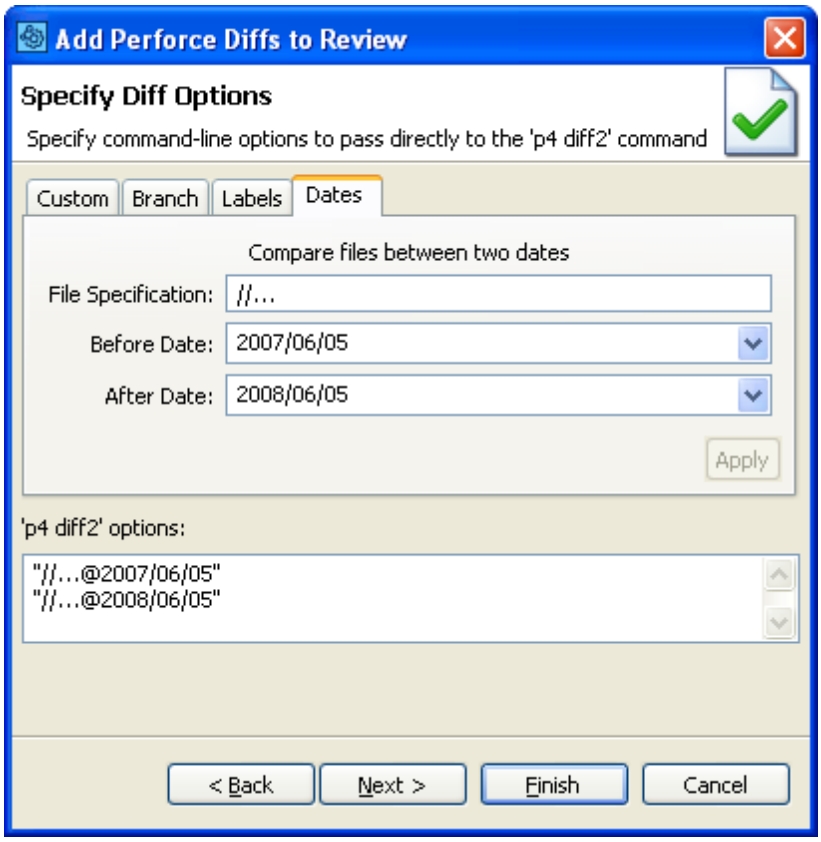
Press the *Add Perforce Diffs...*[772] button on the main screen[496] and then select the *Labels* tab to upload the difference between two Labels.




*Upload the difference between two Labels*

### 6.10.2.3 Comparing two dates

Press the Add Perforce Diffs... 772 button on the main screen 496 and then select the Dates tab to upload the difference between two dates.




*Upload the difference between two dates*

### 6.10.3 Command-Line Client

## Commands recommended for Perforce

`ccollab addchangelist` 778 - Attaches an atomic changelist to a review

`ccollab addp4diffs` 779 - Uploads diffs generated from p4 diff2 command

`ccollab addversions` 781 - Attaches any 2 given versions to a review

`ccollab commit` 783 - Commit changes in the review

`ccollab addp4job` 785 - Adds all numbered local changes that fix a job to the review

`ccollab admin syncusers` 785 - Synchronizes the Perforce user list with the Collaborator server

`ccollab admin changelist update-id` 786 - Updates the changelist ID so Collaborator reflects the renumbered Perforce changelist ID

The addchangelist 778 command will upload Perforce changelists into Collaborator. You can upload changelists either before or after they are submitted, but you cannot upload the default changelist.

## Configuration

In most cases, the Command-Line Client can automatically detect your Perforce configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Perforce configuration or you want to override the detected settings, you can manually specify Perforce settings using global options 519 .

To manually configure the Command-Line Client to use Perforce, execute the following command:

```
ccollab set scm 517 perforce
```

## Perforce-specific Options

| Option | Description |
|---|---|
| `--p4 <value>` | Full path to the P4 executable |
| `--p4port <value>` | How to connect to the Perforce server |
| `--p4user <value>` | Perforce user name |
| `--p4passwd <value>` | Perforce password or ticket |
| `--p4client <value>` | Mapping of Perforce server data to the local machine |
| `--p4-ignore-integration-history <value>` | Ignore integration history when calculating predecessor |
| `--p4charset <value>` | Perforce character set used for translation of Unicode files |
| `--p4-require-empty-default-changelist` | If true, do not allow uploads if the default changelist contains files |

| Option | Description |
|---|---|
| `--p4-specify-command-charset <value>` | Should a character set be specified for communication with Perforce |

If your Perforce server requires ticket-based authentication (server security level 3) then the configured p4passwd will be used to automatically issue 'p4 login' and acquire a new ticket as necessary.

If you want to ignore the integration history of files when determining the previous version of the file and look only at the path you should set p4-ignore-integration-history to yes like so:

```
ccollab set p4-ignore-integration-history yes
```

**Note for Windows users**: Sometimes Perforce server uses Windows machine host name as a default name, ignoring the real Perforce client name. To avoid that issue, specify the needed Perforce client name directly in the Collaborator with the --`p4client <name>` command.

#### 6.10.3.1 addchangelist (for Perforce)

## Description

The `ccollab addchangelist` command attaches all files from a pending or a submitted Perforce changelist to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or Perforce-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour: |

| Option | Required? | Description |
|---|---|---|
| | | • new - the command will create a new review,<br>• ask - the command will pause execution and prompt for the identifier of the desired review,<br>• last - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| <changelist> [<changelist> ...] | Yes | Identifier(s) of the desired changeset(s) in your source control. |

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Changelists or Submitted Changelists to upload.

You cannot specify the default changelist.

## Examples:

To upload Pending Changelists @4321 and @7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

To upload Submitted Changelists @5432 and @12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

### 6.10.3.2 addp4diffs

## Description

The `ccollab addp4diffs` command uploads differences between arbitrary versions of files in Perforce. The differences are generated using the native 'p4 diff2' command of Perforce.

## Command Line Syntax:

```
ccollab [global-options] addp4diffs [--upload-comment <value>]
<review> [<p4-diff-arg> [<p4-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files (defaults to command-line arguments) |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<p4-diff-arg> [<p4-diff-arg> ...]` | No | Options which should be passed to the p4 diff2 command |

## Remarks:

Do not use diff arguments that affect the diff output such as `'-q', '-t', '-d<flag>'`, or `'-s<flag>'`. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.

## Examples:

```
ccollab addp4diffs 698 //depot/file1 //depot/file2
ccollab addp4diffs new //depot/...@1523 //depot/...
ccollab addp4diffs 698 -b my-branch
```

#### 6.10.3.3 addversions (for Perforce)

## Description

The `ccollab addversions` command appends the specified versions (revisions) of a file controlled by Perforce on your computer to a review.

## Command Line Syntax:

```
ccollab [global-options] addversions [--upload-comment <value>] [--
version-spec <value> [<value> ...]] <review> [<file-path>]
[<version>] [<predecessor-version>]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or Perforce-specific global options. See Command-line Global Options Reference 516. |
| `--upload-comment <value>` | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| `--version-spec <value> [<value> ...]` | No | The version to be added to a review. A version-spec value consist of three components:<br><br>*path version* [*previous-version*],<br><br>where *path* is the file name or server path of the file, *version* is the file version to be reviewed, and *previous-version* is an optional version, against which *version* should be compared.<br><br>If any of these arguments contains spaces, enclose it in quotes.<br><br>Typically a version-spec is not used in the command line. We recommend specifying the file and version using the <file-path>, <version> and the <predecessor-version> arguments (see below). |

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<file-path>` | No | The path of the file whose versions are to be added to the review. If filename is omitted, entire directory will be added.<br>**Important:** If you use this option, you should also specify <version> (see below). |
| `<version>` | No | Required, if <file-path> is specified.<br>The version (revision) of the file to be added to the review. You can specify the keyword *local* to tell the command to use the local version of the file. |
| `<predecessor-version>` | No | Preceding file version to be added to the review. If you skip this argument, Collaborator will attempt to determine the preceding version based on the information from the source control. |

## Examples:

The following command line adds versions (revisions) 1.95 and 1.88 of the hello.c file to the review 861:

```
ccollab addversions 861 ./hello.c 1.95 1.88
```

The following command adds these revisions to a new review:

```
ccollab addversions new ./hello.c 1.95 1.88
```

## Remarks

- If you skip the predecessor version, Collaborator will generate diffs using the predecessor version reported by your source control system.
- By default, the command lets you add versions of one file only. To add versions of multiple files, create a text file and specify this file in the command line as the standard input stream (stdin):

  ```
  ccollab addversions last < versionlist.txt
  ```

  Each line in the file must consist of the following components: *path version* [*predecessor-version* ].

  For information on them, see description of the `version-spec` arguments.
- If you skip the file name and versions in the command line, the command will expect to read them from the standard input stream (stdin). Below are some examples for reading versions from the standard input:

  ```
  ccollab addversions 86753
  ccollab addversions last < versionlist.txt
  cat versionlist.txt | ccollab addversions new
  ```
- When specifying the version in the command line or in an input file, you can use the keyword *local* to denote the version corresponding to the local version of the file. The *local* keyword can only be used for the first version argument, not for the predecessor version.

#### 6.10.3.4 commit (for Perforce)

## Description

The `ccollab commit` command submits the changes from a pre-commit review to source control. Be sure to include a relevant comment.

## Command Line Syntax:

```
ccollab [global-options] commit [--comment <value>] [--dismiss-only]
[--force] <review>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--comment <value>` | No | Comment for reviewed changes |
| `--dismiss-only` | No | Just dismiss the Action Item |
| `--force` | No | Ignore potential problems. Must be authenticated as Perforce user with administrator or super access rights. See Remarks. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or an `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

## Remarks

To use this command (without the `--force` option) your Perforce user name (`--p4user`) should coincide with the name of a user who has submitted the desired changelist. If you use the --force option, then you must specify a Perforce user having administrator or super access rights.

## Example:

```
ccollab commit 25 --comment "my code" --force
```

#### 6.10.3.5 addp4job

## Description

The `ccollab commit` command adds all numbered local changes that fix a job to the review.

## Command Line Syntax:

```
ccollab [global-options] addp4job <review> <job-name>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<job-name>` | Yes | Perforce changes that fix this job will be added to the review |

## Example:

```
ccollab addp4job new new_job
```

#### 6.10.3.6 syncusers

## Description

The `ccollab admin syncusers` command synchronizes the Perforce user list with the Collaborator server. All users from Perforce will be be mirrored into Collaborator.

## Command Line Syntax:

```
ccollab [global-options] admin syncusers
```

## Remarks:

- The algorithm is smart enough not to overwrite existing users, only adding new users, so this can be run periodically by a script to keep the lists in sync.
- This is not necessary -- or desirable -- if you are using LDAP authentication.
- You must be logged in as an administrator to execute this command.

#### 6.10.3.7 update-id

## Description

The `ccollab admin changelist update-id` command updates all references to the old changelist ID so Collaborator reflects the renumbered Perforce changelist ID.

## Command Line Syntax:

```
ccollab [global-options] admin changelist update-id <old-id> <new-
id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<old-id>` | Yes | Old ID of Perforce changelist. |
| `<new-id>` | Yes | New ID of Perforce changelist. |

## Example:

```
ccollab --scm p4 --p4port p4server:1666 admin changelist update-id
123 147
```

### 6.10.4 P4V / P4Win Integration

The Collaborator client installer includes an integration point with **P4V** and **P4Win**. When you right-click on one or more Changelists in the GUI a new menu item appears at the bottom of the Tools menu allowing you to add those Changelists to a Review.

# Installation

If you installed the client yourself and opted to "Configure Addons To Perforce Visual Tools[491]", then you should not have to manually configure the Perforce visual tools integrations. However, if the Collaborator client was installed by an administrator on your computer, or the configuration failed at install time, you can manually configure them as follows:

## P4V Installation

Go to the "`Tools`" menu in P4V and choose "`Manage Custom Tools...`". In the dialog box click "`Import Tools...`". In the ensuing open-file dialog, navigate to the Collaborator Client installation directory and select the `P4V-Tools-Import.txt` or `P4V-Tools-Import.xml` file.

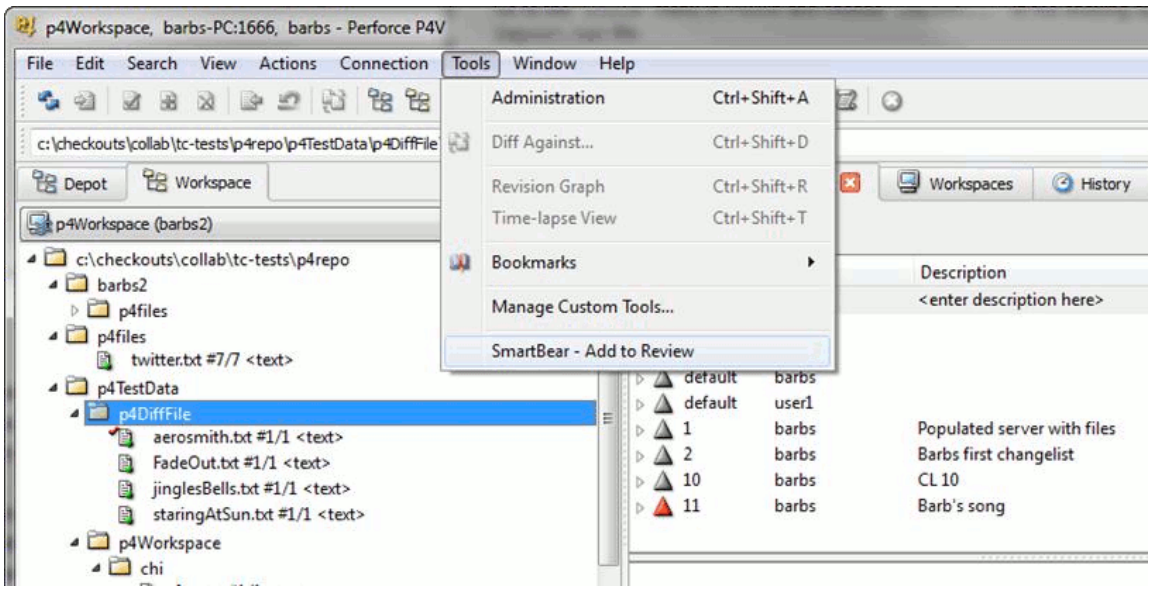
## P4Win Installation

Go to the "Tools" menu in P4Win and choose "Import..".. In the ensuing open-file dialog, navigate to the Collaborator Client installation directory and select the P4Win-Tools-Import.txt file.

There should now be a menu item under "`Tools`" with the text "SmartBear - Add to Review". It will be dimmed unless a Changelist is selected.

> **Warning:** The plug-ins will work only if the Perforce command-line tool (p4) is in your PATH environment variable.

It is also available in the pop-up menu when you right-click a Changelist:



## Working with multiple Perforce workspaces and servers

The P4V/P4Win plug-ins work perfectly and automatically with multiple Perforce servers and servers.

All the Perforce connectivity parameters are taken from the UI, so whatever workspace, server, and authentication you are currently using when you right-click the item will be used automatically be the plug-in.

None of this affects the behavior or configuration of the command-line client[507], so you can freely use the plug-ins without disturbing that configuration.

## Troubleshooting

If you are experiencing problems with the P4V/P4Win plug-ins, it will help SmartBear technical support if you send in a debugging log. To do this, edit the custom tool configuration (Tools > Manage Custom Tools > Edit), and add a --debug argument to the beginning of the arguments list. This will create a log file at the following location: `~/.smartbear/log/ccollab.log` on Unix platforms and `%USERPROFILE%/.smartbear/log/ccollab.log` on Windows platforms.

Please contact our technical support[32] and send the log file along with a full description of what you were trying to do when the error occurred.

### 6.10.5 Perforce Server Triggers

Topics of this section provide information on the Collaborator-specific triggers added to Perforce by Collaborator's Perforce Server Integration package.

## Triggers

Follow the links below to learn more about a trigger:

| Trigger | Description |
|---|---|
| ensure-review-started[789] | Ensures that a review has been started (created in Collaborator) for the submitted changelist. |
| ensure-reviewed[791] | Ensures that a review has been completed for the submitted changelist. |

| Trigger | Description |
|---|---|
| ensure-content-reviewed 792 | Ensures that a review has been completed for the submitted changelist, and that the changelist was not changed after the review was over (that is, that the file contents and the file list are the same). |
| ensure-diffs-reviewed 794 | Ensures that the difference between the current content and proposed content of each file is the same as the difference between the base and proposed content. |
| update-changelist 796 | Appends or updates review-specific information in a changelist's description. |

## Related Materials

For general information on Perforce triggers, see Perforce System Administrator's Guide.

#### 6.10.5.1 ensure-review-started (for Perforce)

## Description

Use the `ensure-review-started` trigger to ensure that a review for the specified changelist has been started. The trigger blocks the submit operation if there is no review for the changelist in Collaborator, and displays an error message telling that the changes need to be reviewed before submitting them to the Perforce repository.

For Perforce this trigger has "change-content" type and is invoked after the changelist creation and file transfer, but before file commit.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-review-started [--
ignore-integrate] [--review-id-regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| <changelist-id> | Yes | The changelist identifier. |

| Option | Required? | Description |
|---|---|---|
| `--ignore-integrate` | No | If specified, Collaborator ignores the merge changes to the source control (that is, the trigger is not fired for these changes. See below). |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install the trigger, add a line like this to your Perforce Triggers list:

```
ccollab change-content //depot/... "/usr/bin/ccollab --url
<collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4client %client% --p4user <p4user> --
p4passwd <p4passwd> admin trigger ensure-review-started %changelist%"
```

Some notes:

- Specify your installation directory for `ccollab` and replace *<values>* with appropriate data. Also, type all parameters in a single line even though the text above occupies several lines.
- The Perforce user that you specify must have at least read-only access to the repository.
- On Unix systems, Perforce incorrectly recognizes arguments that include spaces. It ignores quotes and other standard ways of indicating that the data with spaces is actually one argument. The workaround is to use a separate bash script to call our trigger.

## Remarks

- The trigger ignores changelists that correspond to new branch creation. Creating a new branch does not involve changes in code, so it does not require a review.
- Perforce treats branch merge changelists like ordinary changelists as they bring changes to source code and can introduce bugs. So, by default, the trigger handles this type of changes. However, since the merge changelists often include a large number of files and since you often need to merge several branches, this approach could require cumbersome reviews to be performed many times.
  To avoid possible issues, use the `--ignore-integrate` command-line option. If it is specified, the trigger will ignore changelists that include "merge" changes only. Make sure to review changes in one branch before merging them with other branches.

- For information on Perforce triggers, see Chapter 6 of the Perforce System Administrator's Guide.

#### 6.10.5.2 ensure-reviewed (for Perforce)

## Description

Use the `ensure-reviewed` trigger to ensure that the review that was created for the specified changelist has been completed by the time you submit the changelist to the Perforce repository. If the review has not been completed, the trigger blocks the submit operation and displays an error message informing the user about the problem.

For Perforce this trigger has "change-content" type and is invoked after the changelist creation and file transfer, but before file commit.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-reviewed [--ignore-integrate] [--review-id-regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<changelist-id>` | Yes | The changelist identifier. |
| `--ignore-integrate` | No | If specified, Collaborator ignores the merge changes to the source control (that is, the trigger is not fired for these changes. See below). |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install the trigger, add a line like this to your Perforce Triggers list:

```
ccollab change-content //depot/... "/usr/bin/ccollab --url
<collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4client %client% --p4user <p4user> --
p4passwd <p4passwd> admin trigger ensure-reviewed %changelist%"
```

Some notes:

- Specify your installation directory for `ccollab` and replace <*values*> with appropriate data. Also, type all parameters in a single line even though the text above occupies several lines.
- The Perforce user that you specify must have at least read-only access to the repository.
- On Unix systems, Perforce incorrectly recognizes arguments that include spaces. It ignores quotes and other standard ways of indicating that the data with spaces is actually one argument. The workaround is to use a separate bash script to call our trigger.

## Remarks

- The trigger ignores changelists that correspond to new branch creation. Creating a new branch does not involve changes in code, so it does not require a review.
- Perforce treats branch merge changelists like ordinary changelists as they bring changes to source code and can introduce bugs. So, by default, the trigger handles this type of changes. However, since the merge changelists often include a large number of files and since you often need to merge several branches, this approach could require cumbersome reviews to be performed many times.
  To avoid possible issues, use the `--ignore-integrate` command-line option. If it is specified, the trigger will ignore changelists that include "merge" changes only. Make sure to review changes in one branch before merging them with other branches.
- For information on Perforce triggers, see Chapter 6 of the Perforce System Administrator's Guide.

### 6.10.5.3 ensure-content-reviewed

## Description

Use the `ensure-content-reviewed` trigger to ensure that the review created for the specified changelist has been completed and that the file list and file contents were not changed by the time you submit the changelist. If the review has not been completed or if the review contains changed files or updated file list, the trigger blocks the submit operation and displays an error message informing the user about the problem.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-content-reviewed [--
ignore-integrate] [--no-keywords --review-id-regex <value>]
<changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<changelist-id>` | Yes | The changelist identifier. |
| `--ignore-integrate` | No | (Perforce only.) If specified, Collaborator ignores the merge changes to the source control (that is, the trigger is not fired for these changes. See below). |
| `--no-keywords` | No | If specified, this option disallows keyword expansion in source files. Expanding the keywords may cause Perforce to consider the files as changed. |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install the trigger, add a line like this to your Perforce Triggers list:

```
ccollab change-content //depot/... "/usr/bin/ccollab --url
<collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4client %client% --p4user <p4user> --
p4passwd <p4passwd> admin trigger ensure-content-reviewed %changelist
%"
```

Some notes:

- Specify your installation directory for `ccollab` and replace < *values*> with appropriate data. Also, type all parameters in a single line even though the text above occupies several lines.
- The Perforce user that you specify must have at least read-only access to the repository.

- On Unix systems, Perforce incorrectly recognizes arguments that include spaces. It ignores quotes and other standard ways of indicating that the data with spaces is actually one argument. The workaround is to use a separate bash script to call our trigger.

## Remarks

- The trigger ignores changelists that correspond to new branch creation. Creating a new branch does not involve changes in code, so it does not require a review.
- If your source files use Perforce RCS keywords, we would recommend using the `--no-keywords` command-line option to prevent keyword expansion. Expanded keywords (like `$Date$`, `$Change$` and some others) may cause changes to the source code.
- Perforce treats branch merge changelists like ordinary changelists as they bring changes to source code and can introduce bugs. So, by default, the trigger handles this type of changes. However, since the merge changelists often include a large number of files and since you often need to merge several branches, this approach could require cumbersome reviews to be performed many times.
  To avoid possible issues, use the `--ignore-integrate` command-line option. If it is specified, the trigger will ignore changelists that include "merge" changes only. Make sure to review changes in one branch before merging them with other branches.
- If any of Collaborator clients have the `smartbear.ccollab.upload.truncate.size` 1240 VM-option set up, consider using the `ensure reviewed trigger` instead of this trigger. This VM-option truncates the uploaded files up to the specified size. As a result, the `ensure-diffs-reviewed` and `ensure-content-reviewed` triggers will compare only the remaining parts of the truncated files.
- For information on Perforce triggers, see Chapter 6 of the Perforce System Administrator's Guide.

### 6.10.5.4 ensure-diffs-reviewed

## Description

Use the `ensure-diffs-reviewed` trigger to ensure that the difference between the current content and proposed content of each file is the same as the difference between the base and proposed content in the review by comparing the line-wise changes.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-diffs-reviewed [--
ignore-integrate] [--no-keywords --review-id-regex <value>]
<changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--ignore-integrate` | No | Allow integration changes to proceed without review |
| `--no-keywords` | No | Disallow keyword expansion |
| `--review-id-regex <value>` | No | Regular Expression that identifies Review ID in commit comment |
| `<changelist-id>` | Yes | Changelist ID |

## Installation

To install the trigger, add a line like this to your Perforce Triggers list:

```
ccollab change-content //depot/... "/usr/bin/ccollab --url
<collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4client %client% --p4user <p4user> --
p4passwd <p4passwd> admin trigger ensure-diffs-reviewed %changelist%"
```

Some notes:

- Specify your installation directory for `ccollab` and replace *<values>* with appropriate data. Also, type all parameters in a single line even though the text above occupies several lines.
- The Perforce user that you specify must have at least read-only access to the repository.
- On Unix systems, Perforce incorrectly recognizes arguments that include spaces. It ignores quotes and other standard ways of indicating that the data with spaces is actually one argument. The workaround is to use a separate bash script to call our trigger.

## Remarks

**Important:** If any of Collaborator clients have the `smartbear.ccollab.upload.truncate.size` 1240 VM-option set up, consider using the `ensure reviewed trigger` instead of this trigger. This VM-option truncates the uploaded files up to the specified size. As a result, the `ensure-diffs-reviewed` and `ensure-content-reviewed` triggers will compare only the remaining parts of the truncated files.

### 6.10.5.5 update-changelist

## Description

It is convenient to have information like review id and participants' names in the description of a changelist, for which you created a review in Collaborator. Use this trigger to append review-specific information to or update it in the description automatically.

If you use this trigger as a "submit" trigger, it will add or update review-specific values in the description of a pending changelist when you are submitting this changelist to the repository. If you use this trigger as a "form-in change" trigger, then users will see the updated description whenever they edit the changelist.

All review-specific values appear in a single line in the description. You specify the values to be added in the command line (see below).

## Command Line Syntax:

```
ccollab [global-options] admin trigger update-changelist [--
description-prefix <value>] [--description-template <value>]
<changelist-spec>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--description-prefix <value>` | Yes | Specifies the prefix text for the review line in changelist description. The trigger uses this text to find the review line in the description (see the Remarks section below 797). |
| `--description-template <value>` | Yes | A single-line string that contains an arbitrary text and review-specific data to be posted to the changelist description. To insert review-specific values, use Collaborator variables 161 (`${...}`). See examples in the Installation section below 797. |
| `<changelist-spec>` | Yes | `%changelist%` for a trigger of the *submit* or *commit* type, or `%formfile%` for a trigger of the *form-in* type. |

## Installation

To install the trigger, add a line like one of those below to your Perforce Triggers list:

- Example 1: Adding the trigger as a submit trigger:

```
ccollabupdatechangelist change-commit //depot/... "/usr/bin/ccollab
--url <collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> --
p4client <p4client> admin trigger update-changelist --description-
prefix %quote%Review: %quote% --description-template %quote%ID
${review.id} by ${review.participants.rolename}%quote% %changelist%"
```

- Example 2: Adding the trigger as a form-in trigger:

```
ccollabupdatechangelist form-in change "/usr/bin/ccollab --url
<collabUrl> --user <collabUser> --password <collabPasswd> --scm
perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd>
admin trigger update-changelist --description-prefix %quote%Review: %
quote% --description-template %quote%ID ${review.id} by ${review.
participants.rolename}%quote% %formfile%"
```

Some notes:

- Specify your installation directory for `ccollab` and replace <*values*> with appropriate data. Also, type all parameters in a single line even though the text above occupies several lines.
- The Perforce user that you specify must have read-write access to the changelist.
- On Unix systems, Perforce incorrectly recognizes arguments that include spaces. It ignores quotes and other standard ways of indicating that the data with spaces is actually one argument. The workaround is to use a separate bash script to call our trigger.

## Remarks

- The *`--description-prefix`* option serves as an identifier of the review line in the changelist description. The trigger uses it to find the review-specific string within the description. If the string is found, the trigger updates information in-place. If the string is not found, the trigger appends a new line with review-specific data to the end of the description.

- If there is a review in Collaborator that matches the specified changelist, the changelist description is always updated (either in-place, or appended as it was described above).
  If there is no review for the specified changelist, the description is updated only if the prefix is already present in the description. If the prefix is not found, the trigger does not append review-specific data to the description.

- If the specified changelist is used in several reviews, then the `${review.id}` variable will refer to the most recent of these reviews.

- For information on variables that you use to insert review-specific data, see Variable Substitution 161. The examples in the Installation section insert the review identifier and a list of participants, plus the appropriate English words.

- For information on Perforce triggers, see Chapter 6 of the Perforce System Administrator's Guide.

# 6.11 Subversion Integration

Topics of this section describes Collaborator integration with Subversion.

## Overview

- **Server Integration**
  The Collaborator server can upload files from committed revisions directly from your Subversion server to a review, without users needing to install any client programs. See Subversion Server Integration 801.
  Subversion server-side hooks can ensure code is reviewed 819 or automatically create reviews 819.
- **GUI Client Integration**

  Collaborator's GUI Client can upload local changes to files 805 in a Working Copy, or upload files in Revisions 806. The GUI Client can also upload arbitrary Subversion diffs 807, or the difference between two Revisions 808, branches/tags 809, or dates 810. See Integrating Through Collaborator 804.
- **Command-Line Client Integration**

  Collaborator's Command-Line Client can upload local changes to files in a Working Copy, or upload the files in committed Revisions. The Command-Line Client can also upload arbitrary Subversion diffs. See Integrating Through Command-Line Client 811.
- **Eclipse Plug-in Integration**

  Collaborator's Eclipse Plug-in 525 integrates with the Subclipse and Subversive plug-ins, so you can upload locally modified files 545 or the files in Revisions 564.

## Supported Subversion Versions

Collaborator's Command-Line and GUI clients use the Subversion command-line client (`svn`) to communicate with the Subversion server.

Collaborator's Command-Line and GUI clients support Subversion 1.4.x - 1.9.4. For versions 1.3.x and earlier only the `ccollab addsvndiffs` 816 command is supported.

The TortoiseSVN client is not supported, unless it is installed along with the "command line client tools" option.

Since Collaborator clients work through the Subversion command-line client that is already present on your computer, Collaborator supports all protocols, authentications, proxies, and other client configuration options that you are currently using.

Collaborator's Eclipse Plug-in integrates with the Subclipse and Subversive Eclipse plug-ins and supports the following versions of these plug-ins:

- Subclipse 1.6.x
- Subclipse 1.4.x

- Subversive 4.x (SVN Connector 6.x)

## Support for Directory-Level Changes

Collaborator partially supports Subversion's concept of directories (not just files) being altered.

All the files in directories in question will be scanned, uploaded and represented properly in the Collaborator's GUI. The directories themselves will *not* be shown, or even in the file list confirmation screen presented by the command-line client.

This works correctly even in cases, when you move a parent directory *and* change, add or delete a file in that directory. In the review, you will get the correct content for the file, but the directory itself will not be listed.

Note this limitation affects "commit" functionality. If a directory and a file in it were included in a review and then modified, the Collaborator client will not be able to commit them automatically. The workaround is to commit the changes manually and then "dismiss" the commit Action Item.

## Support for `svn+ssh`

The Collaborator clients support running Subversion over SSH (the `svn+ssh`:// protocol). On some systems, especially those under Unix, Mac OS X or Cygwin/Windows, this will probably work correctly out-of-the-box.

If your SSH connection displays a banner, this will interfere with the client being able to parse the Subversion output correctly. We would recommend suppressing banner output by adding the "-q" option to your SSH executable --

```
svn_ssh = ssh -q
```

-- or by creating a local SSH configuration file and adding the following line to it:

```
LogLevel QUIET
```

On some Windows computers, you might have to configure Subversion to default to your SSH client. To do this:

- Go to the <`Documents and Settings`>*`username`*\`Application Data\Subversion` folder, and open the `config` file for editing.
- In the `[tunnels]` section of this file, find a line that starts with `ssh=`.
- In the line, specify your SSH client. For example:

```
ssh = "C:\\Program Files\\PuTTY\\Plink.exe" -l <your username>
```

**Note:** It is important to use double slashes.

## Support for Symlinks

Collaborator partially supports Subversion's ability to version symlinks. If a committed revision contains a symlink, and you attached this revision to a review, Collaborator will be able to display the symlink contents correctly. However, if a symlink was added to a review before you commit the changes to Subversion, Collaborator displays an incorrect content for the symlink (the symlink will be traversed).

## Changes to Properties

Collaborator does not support reviews of the changes made to Subversion properties.

## Technical Details and Limitations

Review Screen, Diff Viewer, Eclipse Plug-in and Visual Studio Extension display atomic changelists (revisions in terms of Subversion) in chronological order (from older to newer), regardless the order in which they have been uploaded to review.

Collaborator does not guarantee that Diff Viewer will display correct comparison results for the following cases:

- If you have "gaps" while adding subsequent atomic changelists (revisions in terms of Subversion) to the same review. For example, add changelists 1, 2, and 4, but forget to add changelist 3.
- If you add several diffs (non atomic changelists) to the same review.

### 6.11.1 Subversion Server Integration

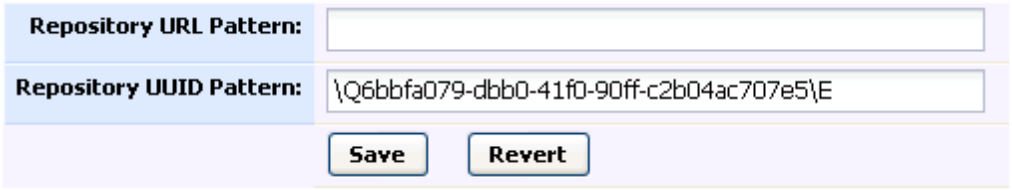
The Collaborator server can be configured to communicate directly with your Subversion server. This allows users to review committed revisions completely from the browser, without having to install any client programs. To enable this feature, first install and configure a Subversion client on the Collaborator server and then create an entry for your Subversion server in the Version Control 204 tab of the administration interface. Version control server entries are also created automatically if one of the client programs uploads files from a server that does not match any of the currently configured servers.


| | |
|---|---|
| **Title** | The title is displayed to users, so it should be something that everyone will understand, even if they are going through proxies, VPNs, or other such things. When a version control server entry is created automatically, this is filled in with the URL the client used to connect to the Subversion server. |
| **Attach changelists from browser** | If enabled, this feature lets users select committed revisions to review directly from the web browser, without having to install any client programs. |
| **svn Executable** | Full path to the 'svn' executable |
| **Repository URL** | Subversion repository URL. When a version control server entry is created automatically, this is filled in with the URL the client used to connect to the Subversion server. |
| **Username** | Subversion user name |
| **Password** | Subversion password |

Click the ***Test Connection*** to make sure Collaborator can contact your Subversion server. If you get an "untrusted certificate authority" error, this is probably because your Subversion server is using a self-signed certificate. You can click ***Accept Certificate*** to tell Collaborator to permanently accept the certificate.

## Client Configuration Mapping

You can supply Java-style regular expressions to map changelists uploaded from our client tools to this Subversion server. It is important to set up these regular expressions so that files uploaded by the various Collaborator client tools are correctly associated with this server-side Subversion configuration.



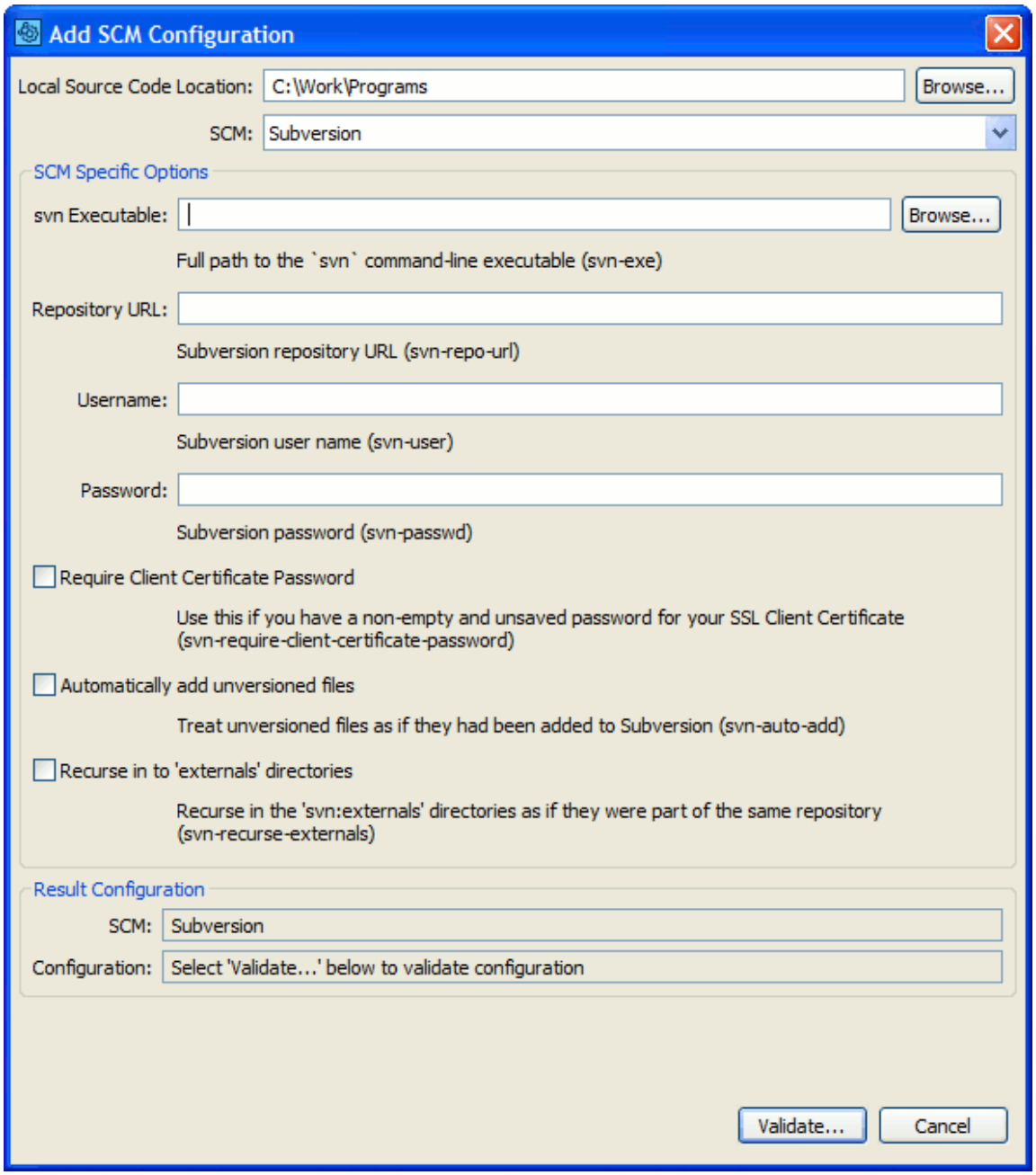

**Repository URL Pattern** — Match on the client's configured repository URL. This is not very reliable because clients may have various network configurations that make the URL look different. It is usually far better to use the Repository UUID Pattern 803.

**Repository UUID Pattern** — Match on the "Repository UUID" returned from running "`svn info`". This is a unique ID generated by every Subversion repository, and usually works perfectly for identifying uploads to this Subversion server. When a version control server entry is created automatically, this is filled in with the "Repository UUID" obtained from the client.

### 6.11.2 GUI Client

## Subversion-specific Options

The SCM Configuration dialog[498] has several Subversion-specific options.

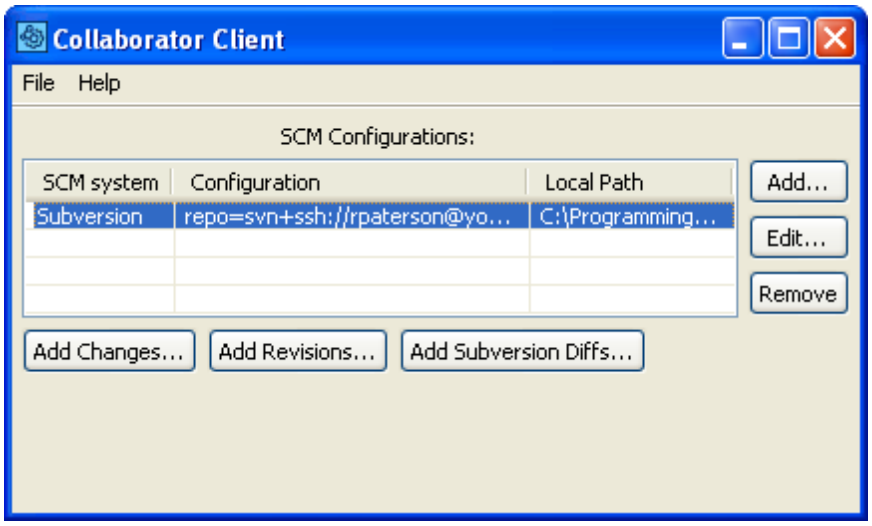



*Subversion SCM Configuration*

## Uploading files to a Review

Selecting a Subversion SCM Configuration in the GUI Client main screen[496] causes several Add to Review buttons to appear. The *Add Changes...* button will be disabled if you have not specified a working copy in the *Local Path* field of the SCM Configuration dialog[804].

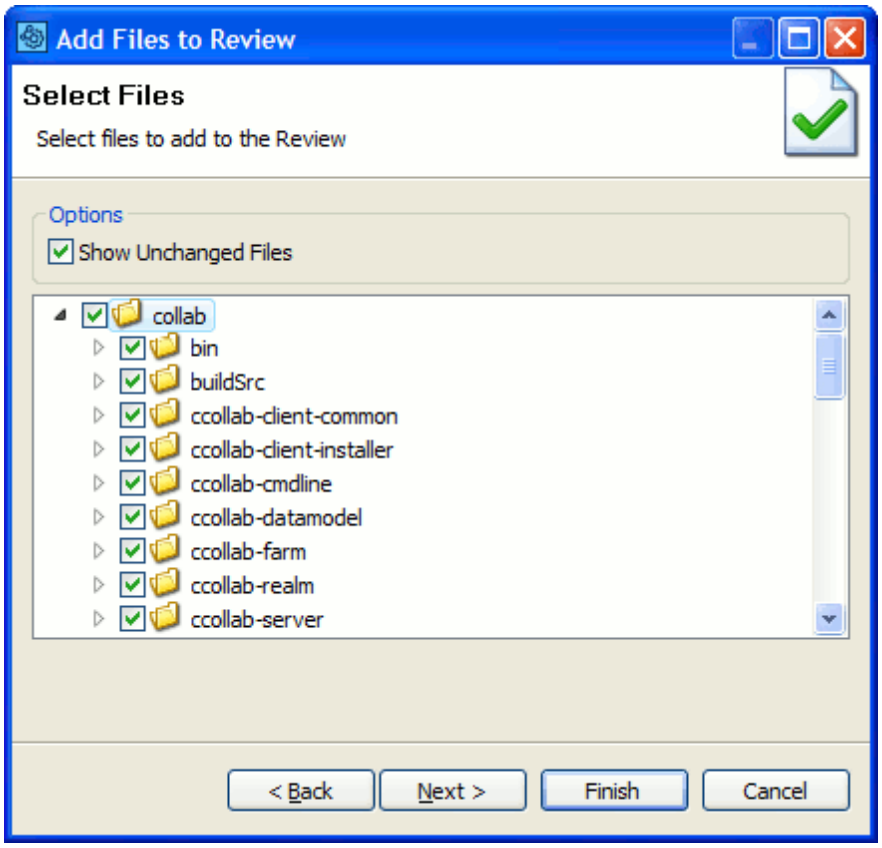
The *Add Changes...*[805] button uploads modified files from a Working Copy. The *Add Revisions...*[806] button uploads files in committed Revisions. The *Add Subversion Diffs...*[807] button uploads arbitrary diffs, or compares the difference between two Revisions[808], branches / tags[809], or dates[810].




*Uploading Subersion files to a Review*

## Add Changes

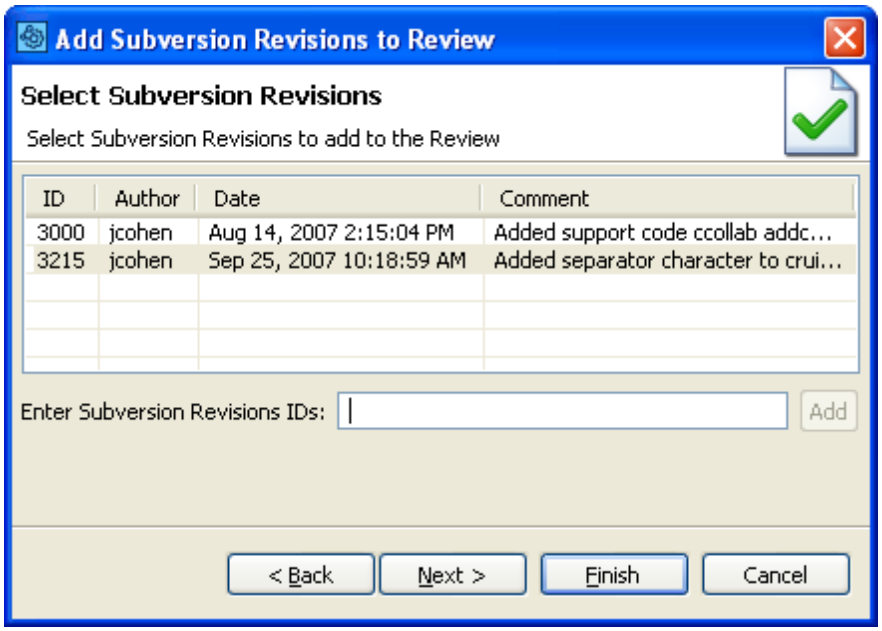
Press the *Add Changes...* button to upload modified files in a Subversion working copy to the Collaborator Server for review. The *Add Changes...* button will be disabled if you have not specified a working copy in the *Local Path* field of the SCM Configuration dialog[804].




*Add Changes*

## Add Revisions

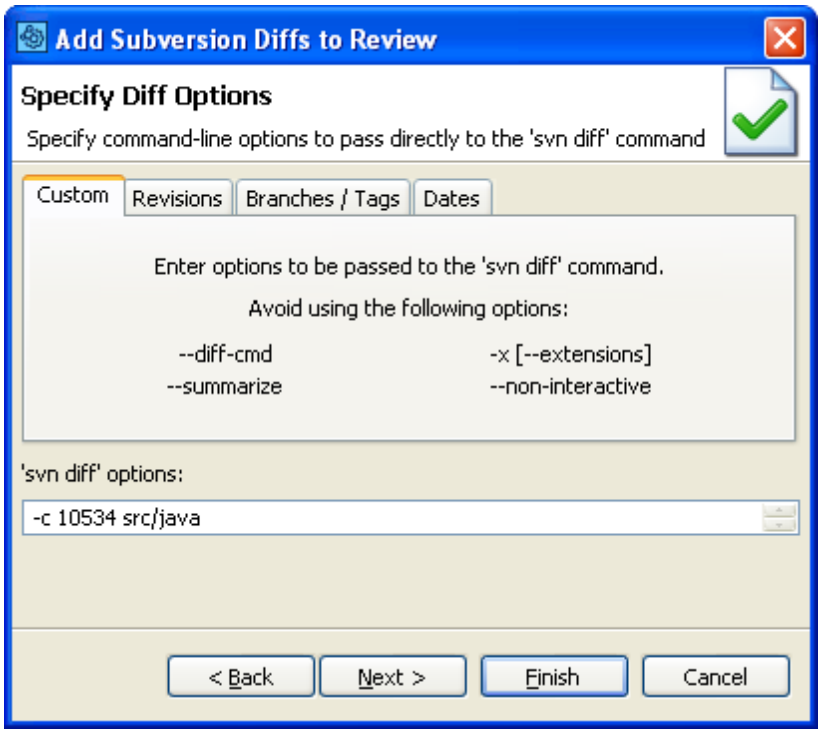
Press the *Add Revisions...* button to upload committed Revisions to the Collaborator Server for review. All of the files modified in the selected Revisions will be uploaded.




*Add Subversion Revisions*

## Add Subversion Diffs

Press the *Add Subversion Diffs...* button to upload arbitrary Subversion diffs to the Collaborator Server for review.

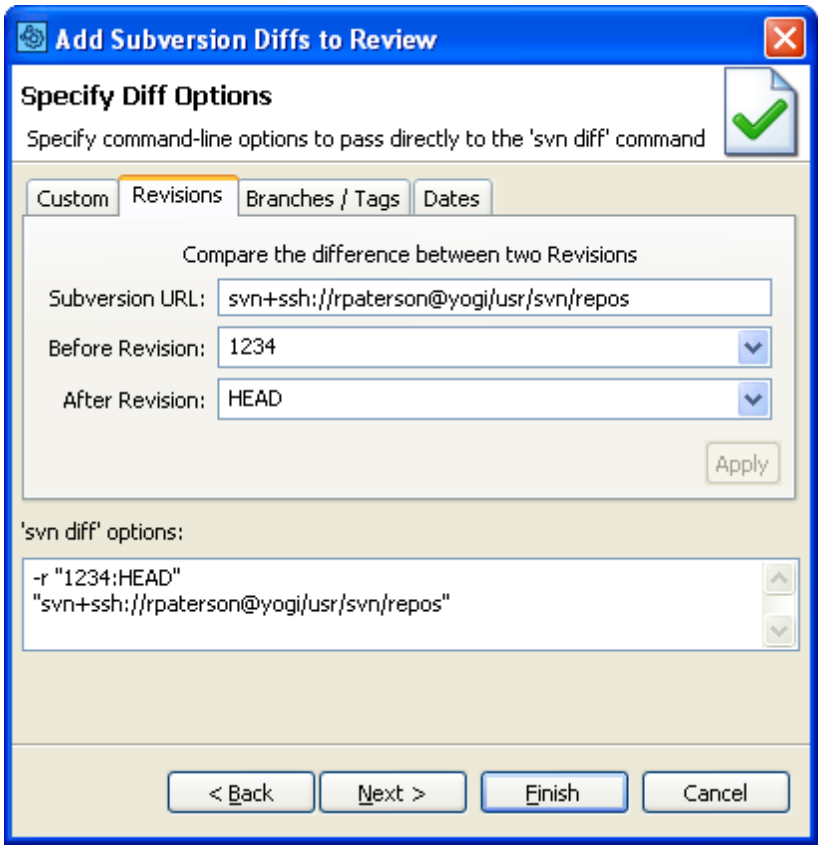



*Add Subversion Diffs*

You can enter arbitrary Subversion diff options, or compare the difference between two Revisions 808, branches / tags 809, or dates 810.

### 6.11.2.1 Comparing two Revisions

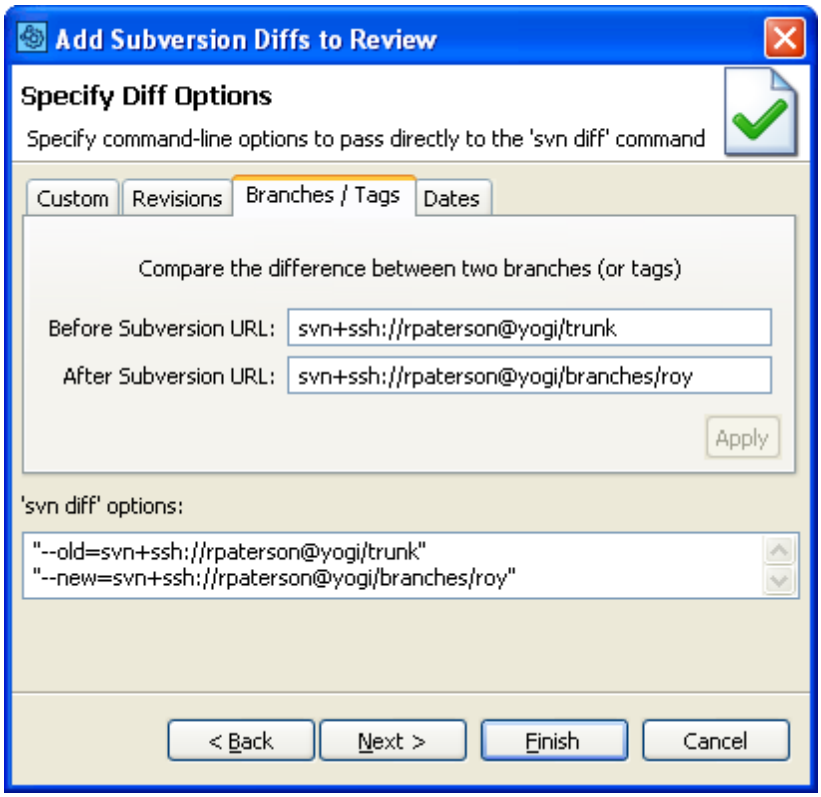
Press the *Add Subversion Diffs...* 807 button on the main screen 496 and then select the *Revisions* tab to upload the difference between two Revisions.




*Upload the difference between two Revisions*

### 6.11.2.2 Comparing two branches / tags

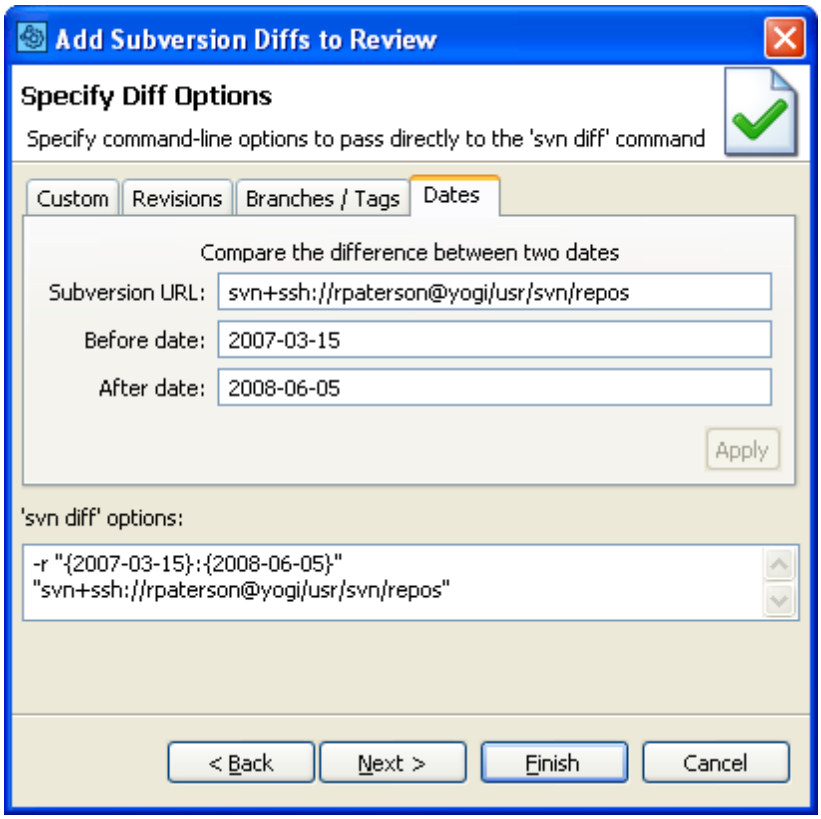
Press the *Add Subversion Diffs...*[807] button on the main screen[496] and then select the *Branches* tab to upload the difference between two branches / tags.




*Upload the difference between two branches / tags*

### 6.11.2.3 Comparing two dates

Press the *Add Subversion Diffs...* 807 button on the main screen 496 and then select the *Dates* tab to upload the difference between two dates.




*Upload the difference between two dates*

### 6.11.3 Command-Line Client

## Commands recommended for Subversion

ccollab addchanges 813 - Attaches locally-modified files to a review

ccollab addchangelist 814 - Attaches an atomic changelist to a review

ccollab addsvndiffs 816 - Uploads diffs generated from the svn diff command

ccollab commit 818 - Commit changes in the review

The addchanges 813 command will upload local changes to files controlled by Subversion before they are checked into version control.

The addchangelist 814 command will upload committed Subversion revisions. The changelist 815 id is the Subversion Revision number. All the files involved in the Revision are uploaded.

## Configuration

In most cases, the Command-Line Client can automatically detect your Subversion configuration. Try testing your configuration 507 to verify the configuration is detected correctly.

If the Command-Line Client is unable to detect your Subversion configuration or you want to override the detected settings, you can manually specify Subversion settings using global options 521.

To manually configure the Command-Line Client to use Subversion, execute the following command:

```
ccollab set scm 517 subversion
```

## Subversion-specific Options

| Option | Description |
|---|---|
| `--svn-exe <value>` | Full path to the `svn` command-line executable |
| `--svn-look-exe <value>` | Full path to the `svnlook` command-line executable (used by Subversion triggers) |
| `--svn-repo-url <value>` | Subversion repository URL |
| `--svn-user <value>` | Subversion user name |
| `--svn-passwd <value>` | Subversion password |
| `--svn-require-client-certificate-password <value>` | Use this if you have a non-empty and unsaved password for your SSL Client Certificate |
| `--svn-auto-add` | Treat unversioned files as if they had been added to Subversion |
| `--svn-recurse-externals` | Recurse in the 'svn:externals' directories as if they were part of the same repository |
| `--svn-repo-path <value>` | Full path to the repository (used by Subversion Triggers) |

If you want to review committed Subversion revisions but you do not have a working copy checked out, you must configure your Subversion URL using `svn-repo-url`.

#### 6.11.3.1 addchanges (for Subversion)

## Description

The `ccollab addchanges` command locally modified files controlled by Subversion to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchanges [--upload-comment <value>]
<review> <file-spec> [<file-spec> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| global-options | No | A number of global or Subversion-specific global options. See Command-line Global Options Reference 516. |
| --upload-comment <value> | No | A comment to be used for the uploaded files. Default is *Local changes*. |
| <review> | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

| Option | Required? | Description |
|---|---|---|
| <file-spec> [<file-spec> ...] | Yes | Files to be added and/or folders to scan for modified files.<br><br>Separate multiple file and folder names with spaces. If a file or folder name contains spaces, enclose this name is quotes.<br><br>`ccollab` scans folders recursively. The resulting list includes the name of modified files. "Modifications" include include edits, additions, deletions, branches, integrations, moves, copies, and so on.<br><br>After the scan is complete, a file list is presented in a graphical editor so you can review the files to be uploaded and make correct the list, if needed. |

## Examples:

To create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

#### 6.11.3.2 addchangelist (for Subversion)

## Description

The `ccollab addchangelist` command attaches all files from a submitted Subversion changelist (revision) to a review on the Collaborator server.

## Command Line Syntax:

```
ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `[global-options]` | No | A number of global or PTC-specific global options. See Command-line Global Options Reference 516. |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<changelist> [<changelist> ... ]` | Yes | Identifier(s) of the desired changeset(s) in your source control. |

## Examples:

To upload revisions r4321 and r7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

To upload revisions r5432 and r12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

#### 6.11.3.3 addsvndiffs

## Description

The `ccollab addsvndiffs` command uploads differences between arbitrary versions of files in Subversion. The differences are generated using the native '`svn diff`' command of Subversion.

## Command Line Syntax:

```
ccollab [global-options] addsvndiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--upload-comment <value>` | No | Comment used to upload files (defaults to command-line arguments) |
| `<review>` | Yes | Identifier of the desired review (an integer number), or a `new`, `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `new` - the command will create a new review,<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |
| `<user-diff-arg> [<user-diff-arg> ...]` | No | Command-line arguments to pass directly to the diff command |

## Remarks:

Do not use diff arguments that affect the diff output such as '`--diff-cmd`', '`-x [--extensions]`', '`--summarize`', or '`--non-interactive`'. The Collaborator command-line client will automatically select an output format that ensures you will get all the data you need on the server.

## Examples:

```
ccollab addsvndiffs 698 -r 2:16
ccollab addsvndiffs new -r PREV http://my.svn.server/svn/repo
ccollab addsvndiffs 698 -c 10534 src/java
ccollab addsvndiffs new OLDURL[@OLDREV] NEWURL[@NEWREV]
```

#### 6.11.3.4 commit (for Subversion)

## Description

The `ccollab commit` command submits the changes from a pre-commit review to source control. Be sure to include a relevant comment.

## Command Line Syntax:

```
ccollab [global-options] commit [--comment <value>] [--dismiss-only]
[--force] <review>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--comment <value>` | No | Comment for reviewed changes |
| `--dismiss-only` | No | Just dismiss the Action Item |
| `--force` | No | Ignore potential problems |
| `<review>` | Yes | Identifier of the desired review (an integer number), or an `ask`, or `last` keyword. Where keywords define the following behaviour:<br>• `ask` - the command will pause execution and prompt for the identifier of the desired review,<br>• `last` - the command will use the last review that was created on the current machine via Command-Line Client (that is, it does not know about reviews created elsewhere). |

## Example:

```
ccollab commit 25 --comment "my code" --force
```

## 6.11.4 Subversion Server Hooks

### Triggers recommended for Subversion

`ccollab admin trigger ensure-review-started` 820 - Changelist cannot be submitted until review of this changelist exists

`ccollab admin trigger ensure-reviewed` 822 - Changelist cannot be submitted until review of this changelist has been completed

`ccollab admin trigger create-review` 823 - Creates a new Review for a changelist if no Review already exists

The ensure-review-started 820 and ensure-reviewed 822 hooks ensure that files cannot be committed unless certain conditions are met. If a user attempts to commit files that do not meet those conditions, an error message describing the unfulfilled conditions will be displayed and the files will not be committed. If the conditions are met, the commit will be allowed to continue normally. The ensure-review-started hook requires that the review exist; ensure-reviewed requires that the review be completed.

The create-review 823 hook automatically creates a review in Collaborator after the revision is committed to the Subversion server. Because you can supply the regular expression for identifying reviews, you can provide users with the ability to review before check-in without having an additional review automatically created after the check-in. This way some groups (or just some check-ins arbitrarily) can use pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

### Linking reviews with commits

To use the ensure-review-started 820 and ensure-reviewed 822 hooks, you must first require developers to put the review ID somewhere in the Subversion commit message (also optionally for the create-review 823 hook). The format of this text is completely up to you; you will need to supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text. The regular expression is specified using the --review-id-regex hook command option.

Here are some common ways of specifying the review ID and the corresponding regular expressions. Note that regular expressions are *case-insensitive* and you must *identify the review ID portion with parenthesis*:

| Text | --review-id-regex |
|---|---|

| | |
|---|---|
| Review: 4233 | review:\s*(\d+) |
| rID4233 | rid(\d+) |
| (review 4233) | \(review (\d+)\) |

This text can appear in-line with other text or in a more formal "form-style" layout. Because you control the regular expression, you can control exactly what this looks like.

For more information about Subversion hooks in general, see the Subversion documentation.

#### 6.11.4.1 ensure-review-started (for Subversion)

## Description

Use the `ensure-review-started` trigger to ensure that a review for the specified changelist has been started. The trigger blocks the submit operation if there is no review for the changelist in Collaborator, and displays an error message telling that the changes need to be reviewed before submitting them to the Subversion repository.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-review-started [--
review-id-regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| <changelist-id> | Yes | The changelist identifier. |
| --review-id-regex <value> | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

### Example Windows batch file:

```
"C:\Program Files\Collaborator Client\ccollab.exe" --url <collabUrl>
--user <collabUser> --password <collabPasswd> --scm subversion --svn-
user <svnUser> --svn-passwd <svnPasswd> --svn-repo-path %1 --svn-
look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger
ensure-review-started --review-id-regex "review:\s+(\d+)" %2 || exit
1
```

### Example Linux/OSX shell script:

```
/collab/install/ccollab --url <collabUrl> --user <collabUser> --
password <collabPasswd> --scm subversion --svn-user <svnUser> --svn-
passwd <svnPasswd> --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook
admin trigger ensure-review-started --review-id-regex "review:
\s+(\d+)" $2 || exit 1
```

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

## Remarks

- You must specify the --svn-repo-path and --svn-look-exe global options.
- You need to specify the --svn-user and --svn-passwd global options. At that, in order for this hook to work smoothly you will need your Subversion usernames and Collaborator logins to match (differs at most in capitalization).
- If you have an empty password for Collaborator account (which is not recommended), you should pass an empty string as the password global option: `--password ""`
- You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

#### 6.11.4.2 ensure-reviewed (for Subversion)

## Description

Use the `ensure-reviewed` trigger to ensure that the review that was created for the specified changelist has been completed by the time you submit the changelist to the Subversion repository. If the review has not been completed, the trigger blocks the submit operation and displays an error message informing the user about the problem.

## Command Line Syntax:

```
ccollab [global-options] admin trigger ensure-reviewed [--review-id-regex <value>] <changelist-id>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `<changelist-id>` | Yes | The changelist identifier. |
| `--review-id-regex <value>` | No | A regular expression that identifies the review ID in the commit is comment. |

## Installation

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

### Example Windows batch file:

```
"C:\Program Files\Collaborator Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-user <svnUser> --svn-passwd <svnPasswd> --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" %2 || exit 1
```

### Example Linux/OSX shell script:

```
/collab/install/ccollab --url <collabUrl> --user <collabUser> --
password <collabPasswd> --scm subversion --svn-user <svnUser> --svn-
passwd <svnPasswd> --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook
admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" $2
|| exit 1
```

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

## Remarks:

- You must specify the --svn-repo-path and --svn-look-exe global options.
- You need to specify the --svn-user and --svn-passwd global options. At that, in order for this hook to work smoothly you will need your Subversion usernames and Collaborator logins to match (differs at most in capitalization).
- If you have an empty password for Collaborator account (which is not recommended), you should pass an empty string as the password global option: `--password ""`
- You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

#### 6.11.4.3 create-review

## Description

The `admin trigger create-review` trigger automatically creates a review in Collaborator after the change is committed to the Subversion server. It is smart enough to not create reviews if they have already been created for this code.

## Command Line Syntax:

```
ccollab [global-options] admin trigger create-review [--add-on-match
<value>] [--review-id-regex <value>] <changelist>
```

## Command Options

| Option | Required? | Description |
|---|---|---|
| `--add-on-match <value>` | No | If a review ID regex matches, add this changelist to the review |
| `--review-id-regex <value>` | No | Regular Expression that identifies Review ID in commit comment |
| `<changelist>` | Yes | SCM-specific ID of an atomic set of changes |

## Installation

For Subversion, to install this trigger you will need to create a post-commit hook. If you already have a post-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

### Example Windows batch file:

```
"C:\Program Files\Collaborator Client\ccollab.exe" --url <collabUrl>
--user <collabUser> --password <collabPasswd> --scm subversion --svn-
user <svnUser> --svn-passwd <svnPasswd> --svn-repo-url svn://url/to/
repo --svn-exe "C:\Program Files\Subversion\bin\svn.exe" admin
trigger create-review --review-id-regex "review:\s+(\d+)" %2 || exit
1
```

### Example Linux/OSX shell script:

```
/collab/install/ccollab --url <collabUrl> --user <collabUser> --
password <collabPasswd> --scm subversion --svn-user <svnUser> --svn-
passwd <svnPasswd> --svn-repo-url svn://url/to/repo --svn-exe /usr/
bin/svn admin trigger create-review --review-id-regex "review:
\s+(\d+)" $2 || exit 1
```

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

## Remarks:

- You need to specify the *--svn-user* and `--svn-passwd` global options. At that, in order for this hook to work smoothly you will need your Subversion usernames and Collaborator logins to match (differs at most in capitalization). This ensures that when a developer checks in code the review is created under his Collaborator account.
  If you do not do this the hook will still work, but reviews will be created under the system administrator's account. Someone will have to log into the system with that account and assign the review to someone else.

- If you have an empty password for Collaborator account (which is not recommended), you should pass an empty string as the password global option: `--password ""`

- You may optionally specify a regular expression that identifies a Collaborator review ID inside a Subversion commit message. If you supply the regular expression, the create-review hook will check for this text in commit messages and automatically not create a new review if a non-canceled review with that ID already exists. This allows some users to do pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

### 6.11.5 Subversion-Specific Tips

- Error messages about "inconsistent line ending style" indicate that the file(s) being operated on have mixed line endings (typically Unix and DOS-style) and that one or more of them has the subversion property svn:eol-style set to native. This is a fatal error as subversion stops processing when the condition is found. The fix is to either force the line endings to a consistent state (that is, typically either all unix or all DOS-style) or propdel the svn property if there is a need for mixed line endings. A variety of methods are available to automate conversion of line ending styles; the free utility unix2dos/dos2unix exists for many platforms for example.

# 7 Repository Hosting Service Integrations

Collaborator provides support for most popular repository hosting services: GitHub, Bitbucket, GitLab. This gives you the possibility to review the changes pushed to web-based repository or pull/merge requests in that repository. The integrations save the time and efforts needed to synchronize commits in web-based repositories and Collaborator reviews.

## In This Section

- Repository Hosting Service Integrations: Overview 826
  Provides general information on how Collaborator integrates with repository hosting services.

- Configure Repository Hosting Service Integrations 840
  Describes how to setup integrations with remote repository hosting services supported by Collaborator.

## Related Topics of Interest

- Version Control Integrations 620
  Describes integrations with version control systems.

# 7.1 Repository Hosting Service Integrations: Overview

This topic provides an overview of integrations with repository hosting services.



## Supported Repository Hosting Services

- GitHub
- GitHub Enterprise
- Bitbucket (both Git and Mercurial repositories)
- Bitbucket Server (both Git and Mercurial repositories)

Collaborator's Bitbucket integration will no longer be able to support Mercurial starting on June 1, 2020, when Atlassian will officially remove Mercurial features and repositories from Bitbucket and its API.

- GitLab
- GitLab Community Edition
- GitLab Enterprise Edition
- Azure DevOps Services (cloud hosting at dev.azure.com),
- Azure DevOps Services (cloud hosting at visualstudio.com) formerly named Visual Studio Team Services (VSTS),

- Azure DevOps Server (on-premises hosting) formerly named Team Foundation Server (TFS).

Team Foundation Version Control (TFVC) repositories of Azure DevOps are not supported.

## Requirements

1. **Setup, configure and enable integration** - Your Collaborator administrator should create a new Remote System Integration configuration. The configuration defines what repository hosting services to use, what repository and what branches to monitor, repository access credentials so on. Each single configuration instance tracks changes in a single web-based repository. To track multiple repositories, your administrators should create a separate configuration for each repository. For detailed instructions on how to setup each integration, see Configuring Repository Hosting Service Integrations 840.

2. **Create webhook** - Webhooks notify Collaborator about the activity of the remote repository. Collaborator tries to create webhooks automatically. Your administrators would need to make your Collaborator server be accessible through Internet (configure a firewall, enable tunneled connections and so on).

3. **Setup user mapping** - In order to assign reviews to the appropriate person, we need to link the owner of web-based repository with some of Collaborator user accounts. If Collaborator fails to match the web-based repository owner with any of Collaborator users, it will make the Collaborator Administrator the creator and author of the new review. See Link User Accounts 885 for detailed instructions on how to setup user mapping.

## How Web-Repository Integration Works

**When a review is created**

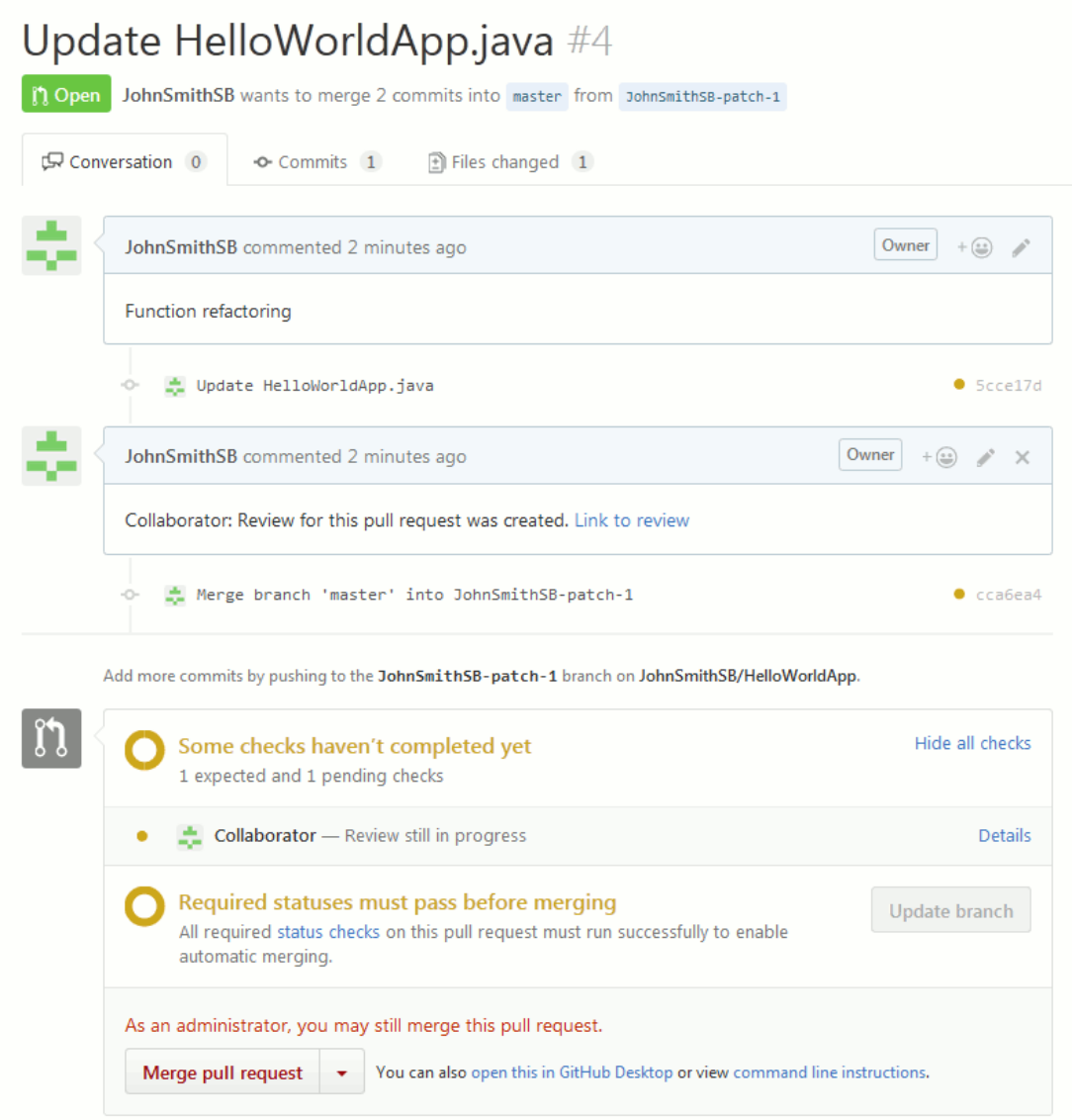
Once configured Collaborator tracks changes at target web repository and automatically creates reviews in the following cases:

- when a pull request (or a merge request, depending on service terminology) is made within one or several specified branches of the repository,

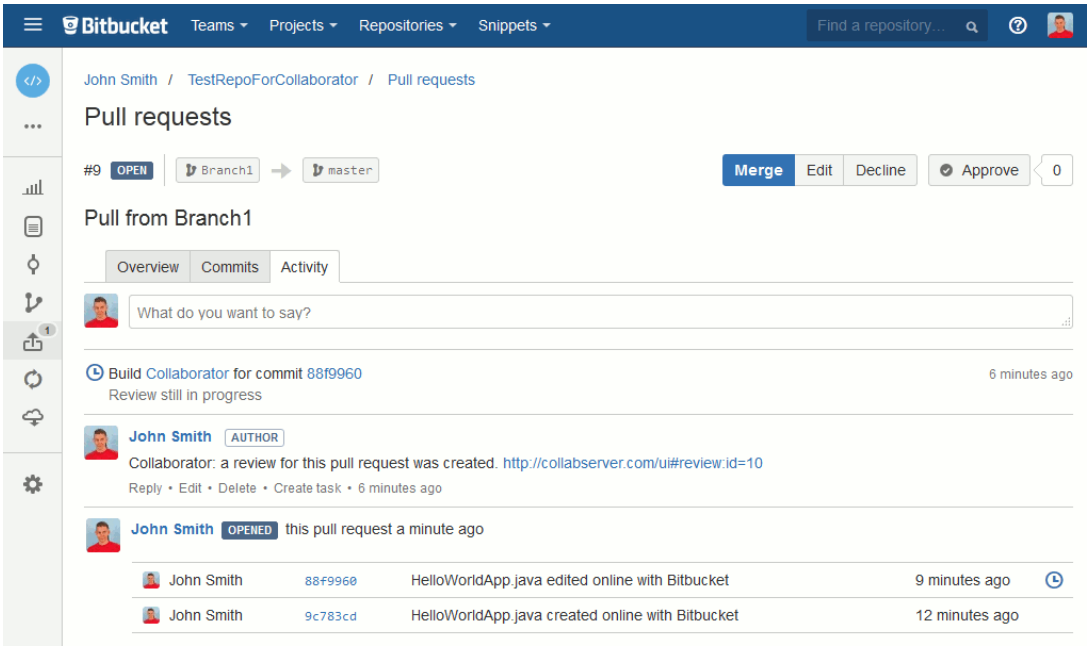
- when a direct push to one or several specified branches of the repository is performed. The latter can be suppressed by the "Ignore pushes for branches" configuration setting.

Additionally Collaborator reviews are registered as repository's continuous integration tests ( ***status checks*** in terms of GitHub, ***builds*** in terms of Bitbucket and Bitbucket Server and ***pipelines*** in terms of GitLab). Thus the current status of corresponding review will be displayed in the pull request. Moreover, for protected branches, Collaborator reviews must be accomplished in order to apply the changes.

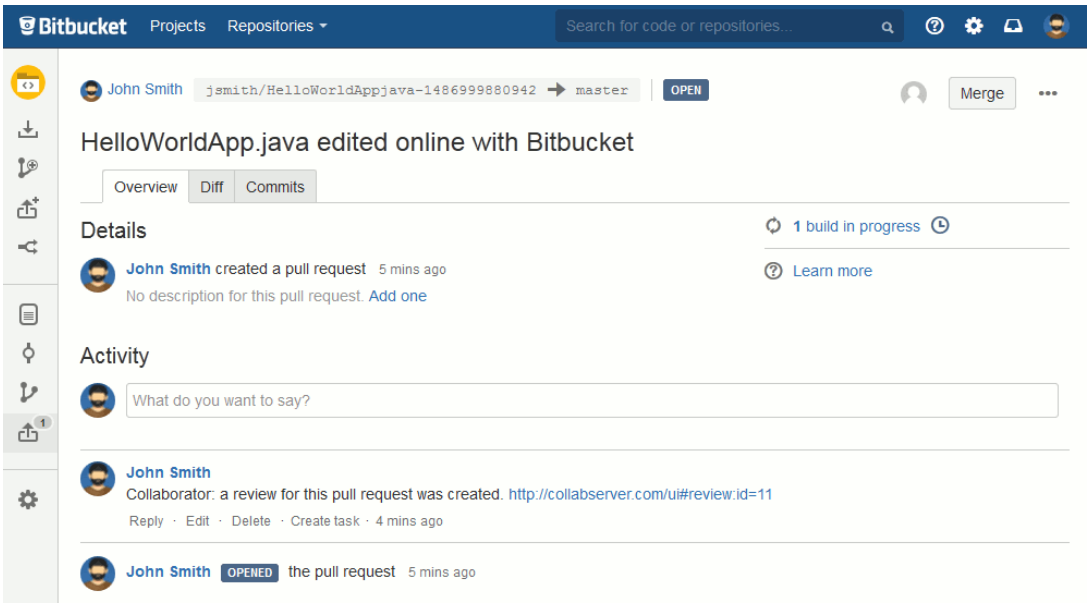
Click to see examples



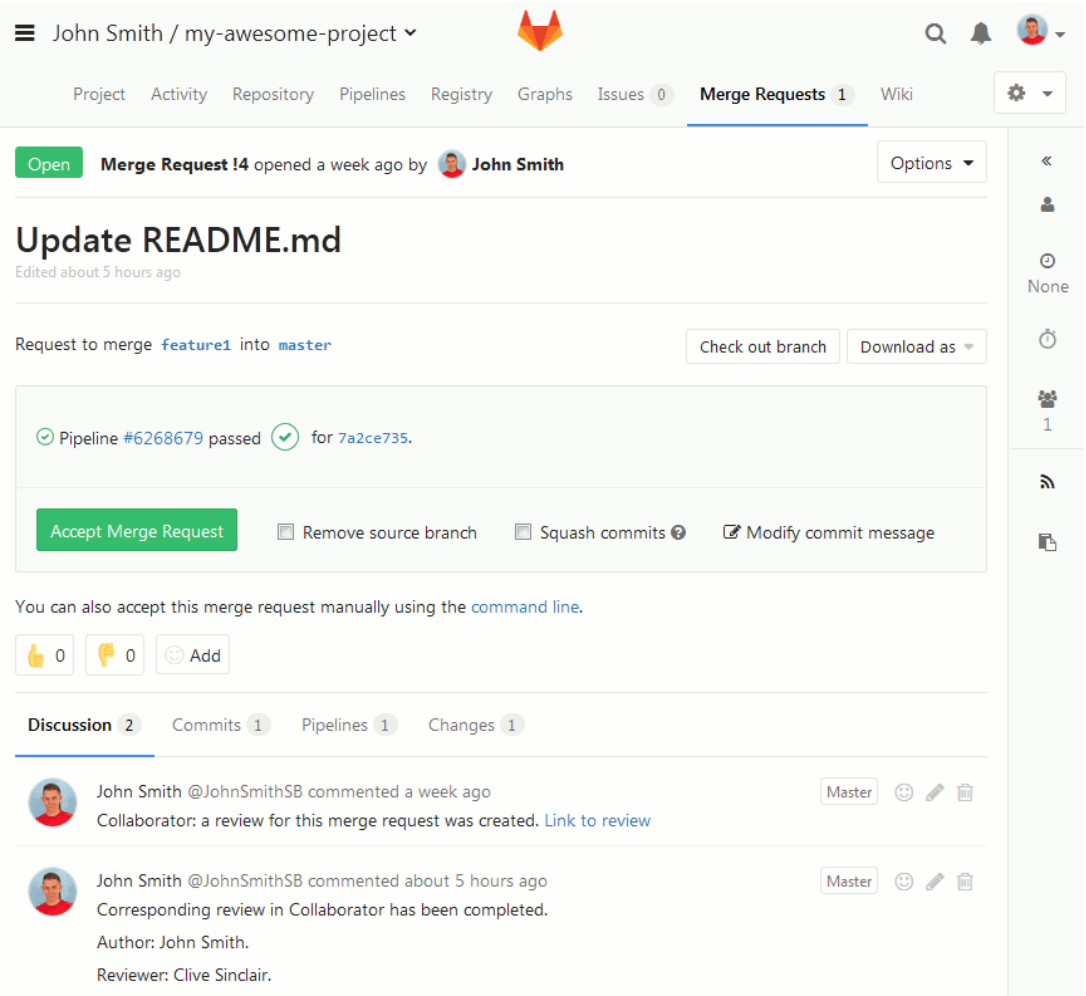

*An example of pull request in a GitHub repository*



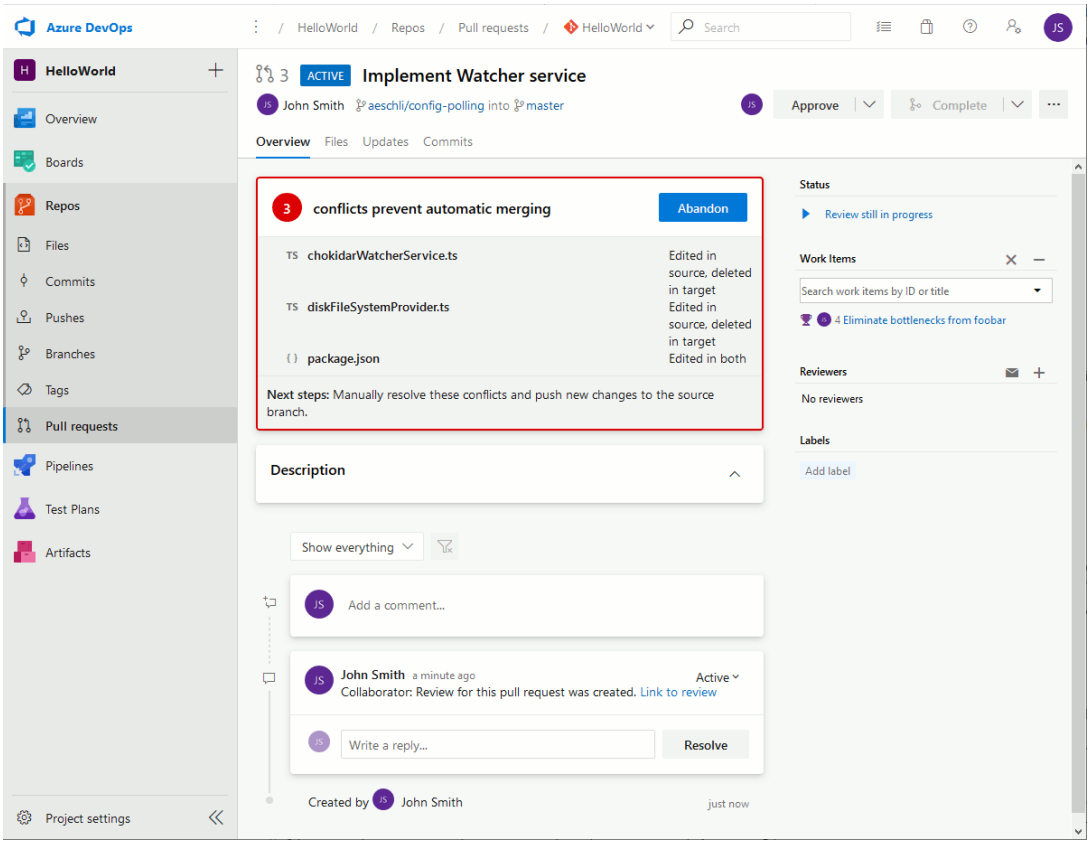

*An example of pull request in a Bitbucket repository*




*An example of pull request in a Bitbucket Server repository*




*An example of merge request in a GitLab repository*




*An example of pull request in a Azure DevOps Git repository*

### Whom the review is assigned to

When Collaborator gets a notification about changes in a repository, it also gets information about the source-control user, who made these changes. So, it attempts to find a corresponding user in Collaborator to assign the new review to that user. In order for the search to be successful, you and your teammates need to link their Collaborator and source-control accounts in Collaborator settings. See Link User Accounts 885 for details.

If Collaborator finds a Collaborator user that matches the source-control user, it assigns that Collaborator user as a review author and creator. If Collaborator does not find a matching user, it assigns the Collaborator administrator as a review author and creator.

Additionally, Collaborator can assign reviewers in the following cases:

- GitHub, GitLab: If repository contains a CODEOWNERS file that defines users responsible for certain files in a repository and integration can match those GitHub/GitLab users with Collaborator users,

- GitHub: If some specific users were added as reviewers when creating pull request on the GitHub side and integration can match those GitHub users with Collaborator users, and the **Auto assign reviewers** option is on.

- GitHub: If some team was added as reviewer when creating pull request on the GitHub side, or some team was specified in CODEOWNERS file and integration can match members of that team with Collaborator users, and the **Auto assign reviewers** option is on.

- Bitbucket, Bitbucket Server: If some specific users were added as reviewers when creating pull request on the Bitbucket side and integration can match those Bitbucket users with Collaborator users.

- Azure DevOps: If some specific users were added as reviewers when creating pull request on the Azure DevOps side and integration can match those Azure DevOps users with Collaborator users, and the **Auto assign reviewers** option is on.

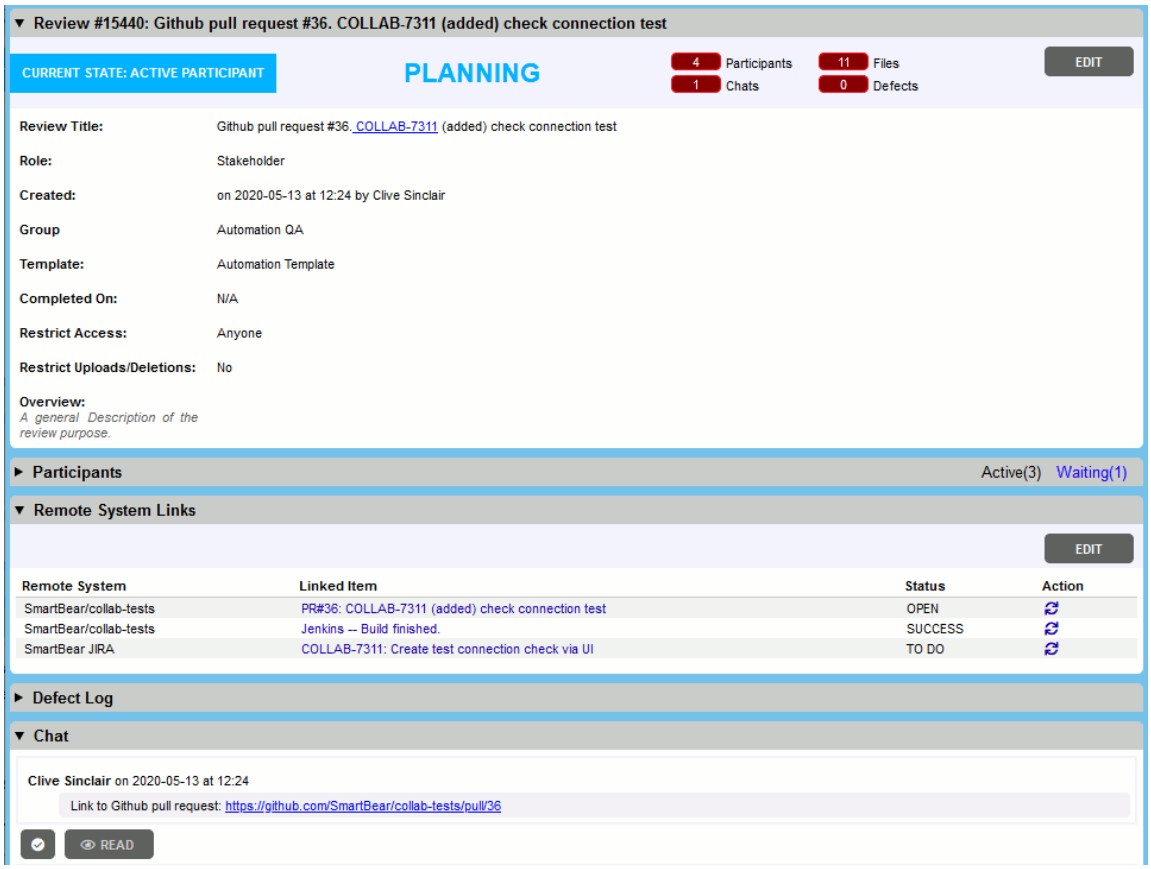
**What the review will contain**

The new review includes changes made to the remote repository. Namely —

- When creating a new review, Collaborator automatically uploads copies of the pull request's or commit's files to the review. That is, the review contains the modified files, not the file links or commit IDs. You can see these files in the Review Materials 357 section of the review.

- The Remote System Links 353 section contains the links to pull requests or commits, as well as their current statuses.

- The Chat 356 section would display a link to the remote repository, names of pull request branches, pull request description (if provided) and comments that users made to pull request/ commit on the remote repository side.




*An example of review created on changes in GitHub repository*

The initial phase of the newly created review will be Planning, so authors could check the review, add notes or upload additional materials before notifying other participants. Alternatively, Collaborator administrators can enable the Move pull request reviews to Inspection 192 setting to start reviews automatically. In this case Collaborator will verify if the review meets the workflow requirements (for example, has minimal number of participants) and will move it to the Inspection phase on success.

**When further commits are made**

- **If a review was created for a pull request**

  If your teammates make more changes to the branch after a review was created, and if the pull request is not closed, Collaborator uploads new file versions to the created review. This works until the pull request is closed. If the review remains open after the pull request is closed, then Collaborator will not upload files from subsequent commits to the review.

- **If a review was created for a commit (push event)**

  Collaborator will not upload new file versions to the review automatically.

**When files conflict**

Once a file conflict could be resolved automatically, the merge commit will not be added to the review. If a file conflict was resolved manually (that is, if some files have been changed to complete the merge), then Web-repository integration will update the review with the changes from the resolving commit.

**Note:** Changes from merge commits will not be displayed in the Diff Viewer only when the Default Revision Comparison of Diff Viewer 324 setting is set to 'Current Branch Changes Only'. You can find this setting under User Preferences 317 in Display tab.

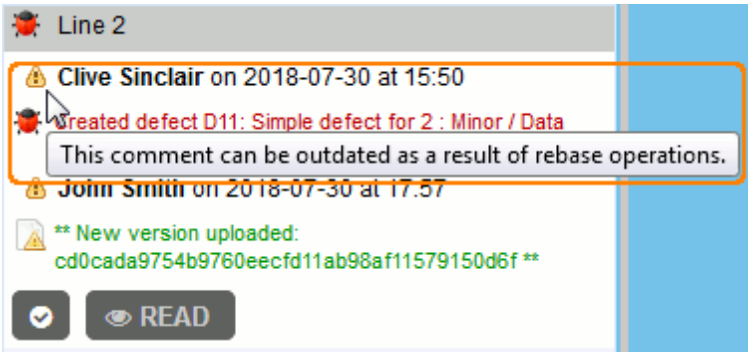
**When merge commit was added**

File changes made by merge commits will only be displayed in Separate view of the Review Materials section of the review. They will not be displayed in Overlay view. Besides, file changes made by merge commits are not taken into account when calculating overall LOC metrics and they do not affect the overall rework count of a file.

**When commit history is modified (rebasing or squashing commits)**

Collaborator will process the outdated commits and will reorder the list of file revisions to keep it up-to-date.

- If some file revisions have been removed, the comments or defects addressing that file revision will be promoted to resulting revision and will display a special icon ⚠ indicating that the comment/defect could be outdated.



- If some files have been removed, they will also be removed from the review materials. For audit purposes, comments and defects for removed files will be copied to the review's general chat.


**When a review is completed**

The integration may automatically merge or close the appropriate pull/merge request. This behaviour is controlled by the "When Review Completed" setting.

**When a review is completed and pull request is updated**

If review was completed and further commits are made in a branch which initiated a pull/merge request - depends on the "Allow to reopen review" setting of the remote system configuration. By default this setting is true, and review will be reopened.

**When a review is cancelled, deleted or rejected**

The integration may automatically close pull/merge request. Additionally it may automatically delete the corresponding feature branch upon closing or merging a pull request. This behaviour is controlled by the "When Review Cancelled/Deleted/Rejected" setting.

**When a pull/merge request is closed**

When a pull/merge request was closed by the remote repository, the integration posts a comment in the respective review.

**When pull request is reopened**

If review was completed and pull request being reopened - review will be reopened.

If review was cancelled/deleted/rejected and pull request being reopened - new review will be created.

**When repository contains  submodules**

Repository hosting integrations can display changes in the submodules of tracked repositories. (Other Collaborator clients[645] cannot do that and ignore submodule changes.)

Submodules must belong to the same owner/project/organization.

Commits that add a new submodule, or remove an existing submodule will be represented as changes in the `.gitmodules` file. Other changes in submodule files will be ignored in this commit.

Once submodule is added, further changes in submodule files will be displayed as file changes in submodule folder.

## Technical Details

- Integration creates reviews on any pull/merge requests - no matter whether it came from the same repository or from the forked repository.

- By default, webhooks created by the Easy Add Repository wizard do not check the if the server's SSL certificate is valid. This is done intentionally, in order to avoid handshake issues on Collaborator servers with self-signed or untrusted certificate authority (CA) root certificates. To enable SSL verification on existing webhooks, modify them on the remote repository server. To enable SSL verification for all new webhooks created by the Easy Add Repository wizard, use the `-Dcom.smartbear.collab.datamodel.remotesystem.webhooks.ssl.enable=true` Java VM Option[1232].

- In order to decrease number of calls to repository hosting servers, Collaborator caches some of the most often retrieved entities from APIs (commits, pull request diffs, commit diffs). By default cache contains up to 20000 entities for each active remote system integration. To change the default cache size or disable caching, use the server's `-Dcom.smartbear.collab.datamodel.remotesystem.cache.size` Java VM Option[1232].
  Usage statistics of cache and all outgoing API calls from Collaborator could be found in the `remoteSystemApi.log`[169] when the `debug` logging level is enabled (log4j.logger.rsApiLog = warn, RemoteSystemAPI) through the `log4j.properties` file.

- Currently, Collaborator incorrectly displays non-latin characters in pull request names, branch names, pull request descriptions.

## System Specifics

### GitHub

- If repository contains a CODEOWNERS file that defines users responsible for certain files in a repository and integration can match those GitHub users with Collaborator users, and the "Auto assign reviewers[841]" option is on, then it will automatically add those users as reviewers on Collaborator's side.

- You can specify GitHub users as reviewers when you are creating a pull request. If your teammates link[885] their GitHub accounts with their Collaborator accounts correctly, and the "Auto assign reviewers[841]" option is on, then Collaborator will automatically add those users as reviewers to the created review on the Collaborator side.

- Similarly, you can specify some team as reviewer when creating pull request on the GitHub side, or specify team in CODEOWNERS file. In this case, Collaborator tries to match members of that team with Collaborator users, and if the "Auto assign reviewers[841]" option is on, it will automatically add those users as reviewers on Collaborator's side.

- It is possible to additionally secure webhook calls by providing a Secret token while creating/modifying a webhook on GitHub server and a GitHub configuration[841] on the Collaborator side. This token will be used as server signature for verifying webhook events. The Secret token parameter is optional and is not specified by default.

- If your GitHub repositories use continuous integration, the Remote System Links section will also display current build statuses of your CI systems (Jenkins, Travis and so forth).

- Just after a new GitHub configuration has been created, the very first push to a repository will not be tracked. Subsequent pushes will be tracked as they should.

- To avoid creating reviews for merge commits of Merge requests their messages should be default or start with the 'Merge pull request #'

- If the remote repository server is not accessible, Collaborator will retry connecting after a certain time period and will perform several retry attempts. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on. Default wait time is 1 second and Collaborator will perform 3 retries. Wait time and maximal number of attempts could be configured via Java VM Options[1234]. If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to the remoteSystem.log[169]

### GitLab

- When configuring user remote accounts[885], user names must be entered as specified in the Full Name field of Profile settings.

- If repository contains a CODEOWNERS file that defines users responsible for certain files in a repository and integration can match those GitLab users with Collaborator users, then it will automatically add those users as reviewers on Collaborator's side.

- It is possible to additionally secure webhook calls by providing a Secret token while creating/modifying a webhook on GitLab server and a GitLab configuration[866] on the Collaborator side. This token will be used as server signature for verifying webhook events. The Secret token parameter is optional and is not specified by default.

- To avoid creating reviews for merge commits of Merge requests their messages should be default or start with the 'Merge branch'

- If your GitLab repository is configured to use merge request pipelines, Collaborator integration will use them as well. Otherwise, it will use regular pipelines.
- Currently, GitLab integration does not support connecting to the target GitLab server through the proxy server.

### GitLab Community / GitLab Enterprise

- When configuring user remote accounts 885, user names must be entered as specified in the Full Name field of Profile settings.
- If repository contains a CODEOWNERS file that defines users responsible for certain files in a repository and integration can match those GitLab users with Collaborator users, then it will automatically add those users as reviewers on Collaborator's side.
- It is possible to additionally secure webhook calls by providing a Secret token while creating/ modifying a webhook on GitLab server and a GitLab configuration 866 on the Collaborator side. This token will be used as server signature for verifying webhook events. The Secret token parameter is optional and is not specified by default.
- To avoid creating reviews for merge commits of Merge requests their messages should be default or start with the 'Merge branch'
- If your GitLab repository is configured to use merge request pipelines, Collaborator integration will use them as well. Otherwise, it will use regular pipelines.
- Currently, GitLab integration does not support connecting to the target GitLab server through the proxy server.
- By default, GitLab Community / GitLab Enterprise servers do not allow performing webhook requests to web services running in the same local network. If your Collaborator server resides on the same machine as GitLab server or within the same local network, then administrator of your GitLab server should enable the "Allow requests to the local network from hooks and services" option in the "Outbound requests" section inside the **Admin > Settings** area (<your-gitlab-server>/admin/application_settings).

### Bitbucket

- When configuring user remote accounts 885, user names must be entered as it is specified in the Name field of the Bitbucket settings.
- While creating pull request on Bitbucket, you can add specific Bitbucket users as reviewers to this request. If integration can match those Bitbucket users with some Collaborator users (that is, if User Remote Accounts 885 are properly configured), then it will automatically add those users as reviewers on Collaborator's side as well.

- In order to create reviews on pull requests from forked repositories the following conditions must comply: Forked repository of the same repository owner can be public or private. Forked repository of a different repository owner must be public repository.
- Just after a new Bitbucket configuration has been created, the very first push to a repository will not be tracked. Subsequent pushes will be tracked as they should.
- First push to newly created branch will be tracked only if it was branched out of main. Workaround: create new branch without commits - second push is tracked as it should.
- To avoid creating reviews for merge commits of Pull requests their messages should be default or start with the 'Merged in'
  - If the remote repository server is not accessible, Collaborator will retry connecting after a certain time period and will perform several retry attempts. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on. Default wait time is 1 second and Collaborator will perform 3 retries. Wait time and maximal number of attempts could be configured via Java VM Options 1234. If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to the remoteSystem.log 169
- Bitbucket will deprecate their API 1.0 starting from 29th of April 2019. You will need to upgrade to Collaborator version 11.5.11502 or later to use Bitbucket integration.

### Bitbucket Server

- Current version of Collaborator supports Bitbucket Server versions 5.4 and later.
- Due to limitation on Bitbucket Server side, integration is not able to create reviews if the paths of submitted files contain the percent symbol (%).
- While creating pull request on Bitbucket, you can add specific Bitbucket users as reviewers to this request. If integration can match those Bitbucket users with some Collaborator users (that is, if User Remote Accounts 885 are properly configured), then it will automatically add those users as reviewers on Collaborator's side as well.
- In order to create reviews on pull requests from forked repositories the following conditions must comply: Forked repository of the same repository owner can be public or private. Forked repository of a different repository owner must be public repository.
- To avoid creating reviews for merge commits of Pull requests their messages should be default or start with the 'Merged in'

- If the remote repository server is not accessible, Collaborator will retry connecting after a certain time period and will perform several retry attempts. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on. Default wait time is 1 second and Collaborator will perform 3 retries. Wait time and maximal number of attempts could be configured via Java VM Options 1234. If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to the remoteSystem.log 169

### Azure DevOps

- Team Foundation Version Control (TFVC) repositories of Azure DevOps are not supported.
- If a pull request is associated with some work item on Azure DevOps side, the link to work item will be added to the review's Remote System Links 353 section. Moreover, if the integration's " Auto close related work items 875" option is enabled, the respective work item will be automatically closed when pull request is merged.
- You can specify Azure DevOps users as reviewers when you are creating a pull request. If your teammates link 885 their Azure DevOps accounts with their Collaborator accounts correctly, and the "Auto assign reviewers 875" option is on, then Collaborator will automatically add those users as reviewers to the created review on the Collaborator side.
- To avoid creating reviews for merge commits of Merge requests their messages should be default or start with the 'Merge pull request #'
- If the remote repository server is not accessible, Collaborator will retry connecting after a certain time period and will perform several retry attempts. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on. Default wait time is 1 second and Collaborator will perform 3 retries. Wait time and maximal number of attempts could be configured via Java VM Options 1234. If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to the remoteSystem.log 169
- Currently, Azure DevOps Server (on-premises version of Azure DevOps) does not trigger webhook on comments made in pull requests. Therefore, these comments could not be copied into Collaborator reviews.

## 7.2 Configure Repository Hosting Service Integrations

This section describes how to setup integrations with remote repository hosting services supported by Collaborator.

- GitHub 841
- Bitbucket Cloud 850
- Bitbucket Server (On-Premise) 859
- GitLab 866

- Azure DevOps 875
- (For on-premise servers) Establish Trusted Connection 884
- (Common for all) Link User Accounts 885

## 7.2.1 Configure GitHub Integration

There are two ways to create a GitHub configuration: automatically via the Easy Add Repository wizard and manually via the Configure Remote Systems tab. The first approach is useful for adding new configurations, while the second allows adding new and modifying existing configurations. This section describes both of these ways.

Additionally, you can create auto-polling configuration to periodically look for new repositories on GitHub server and suggest creating integrations for them.

- Creating GitHub configuration via the Easy Add Repository wizard 841
- Creating GitHub configuration via the Configure Remote Systems tab 845
- Creating GitHub auto-polling configuration 848
- Enable/disable GitHub integration 850

> **Important.** In order to use integration, your Collaborator server must be accessible to the remote system and vice versa. Configure a firewall or enable tunneled connections to expose your local Collaborator server to the Internet.

### Creating GitHub configuration via the Easy Add Repository wizard

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Easy Add Repository** tab.

4. Select *GitHub* in the the **Add repository for** combobox and click **Next**.
   This will display the Easy Add Repository wizard. It helps to setup remote repository configuration on the Collaborator side and add the webhook on the GitHub side.


5. Specify the following settings in the wizard:

| Setting | Description |
|---|---|
| **Personal access token** | Required. The personal access token for the GitHub account to be tracked. |

| | |
|---|---|
| | Read Creating an access token for command-line use on GitHub documentation to learn how to obtain it. You will need to enable the following scopes: `repo`, `admin:org`, `admin:repo_hook` and `admin:org_hook`. |
| **Organization** | Required for organization repositories. Denotes the account name of GitHub organization. |
| **GitHub Enterprise host name** | Required for repositories hosted on GitHub Enterprise servers. Specifies the host name of your GitHub Enterprise server. |
| **Webhooks Secret token** | Optional. The secret token for webhook events. The secret token could be set when creating/modifying a webhook on GitHub server. If provided, must be at least 8 characters long.<br><br>To learn more about webhook settings on the GitHub side, see GitHub documentation:<br><br>https://developer.github.com/webhooks/ |
| **Scopes for user** | Optional. Defines which types of repositories to track.<br><br>Owner: Repositories that are owned by the specified user. (Default)<br><br>Collaborator: Repositories that the user has been added to as a collaborator.<br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user.<br><br>Organization member: Repositories that the user has access to through being a member of an organization. This includes every repository on every team that the user is on.<br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |
| **Override existing configurations** | Optional. Specifies whether to override existing configurations that track the same repository URI. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, "main" branch will be tracked. |

| | |
|---|---|
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **Status check required** | Specifies if the integration should enforce status checks for the tracked branches before merging pull request.<br><br>If this option is enabled, you cannot use regular expressions in the "Branches to track" setting, since this option requires exact name match. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and delete its branch: Merge pull request that corresponds to a review and delete the respective branch.<br><br>Close pull request: Close pull request that corresponds to a review.<br><br>Close pull request and delete its branch: Close pull request that corresponds to a review and delete the respective branch. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Close pull request: Close pull request that corresponds to a review.<br><br>Close pull request and delete its branch: Close pull request that corresponds to a review and delete the respective branch. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made, |

| | |
|---|---|
| | • when a comment is added to a pull request,<br>• when a comment is added to commit which is a part of a pull request |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge pull request" or "Merge pull request and delete its branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off 193 before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Auto assign reviewers** | Whether to assign Collaborator reviewers when some specific users or teams were added as pull request reviewers or specified as code owners on the GitHub side and integration can match those GitHub users with Collaborator users. |

After specifying these values, you can click **Load repositories**. This will display a list of repositories available for the specified user or organization. Select which of them to track and click **Create for selected**. To stop suggesting integrations for some specific repositories, select these repositories in the list and click **Ignore selected**.

Alternatively, you can click **Create for all** and add configurations for all repositories that are available for the specified user or organization.

For every repository a separate configuration will be created in Collaborator and a webhook will be added in GitHub.

Now you need to link some Collaborator user account to the GitHub account 885 of repository owner.

## Creating GitHub configuration via the Configure Remote Systems tab

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Configure Remote Systems** tab.
4. In the **New Remote System Configuration** section, select *GitHub* and click **Create.**
5. Collaborator will display a page with configuration settings. Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | Required. The configuration name as it will be displayed in Collaborator's user interface. |
| **GitHub repo URI** | Required. The URI of GitHub repository to be tracked.<br><br>For instance: *https://github.com/torvalds/linux.git*<br><br>You can copy it from the Clone URI field of the repository's main page on GitHub. |
| **GitHub API token** | Required. The personal access token for the GitHub account to be tracked.<br><br>Read Creating an access token for command-line use on GitHub documentation to learn how to obtain it. You will need to enable the following scopes: `repo` and `admin:org`. |
| **Webhooks secret token** | Optional. The secret token for webhook events. The secret token could be set when creating/modifying a webhook on GitHub server. If provided, must be at least 8 characters long.<br><br>To learn more about webhook settings on the GitHub side, see GitHub documentation:<br><br>https://developer.github.com/webhooks/ |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, "main" branch will be tracked. |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **Status check required** | Specifies if the integration should enforce status checks for the tracked branches before merging pull request. |

| | |
|---|---|
| | If this option is enabled, you cannot use regular expressions in the "Branches to track" setting, since this option requires exact name match. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and delete its branch: Merge pull request that corresponds to a review and delete the respective branch.<br><br>Close pull request: Close pull request that corresponds to a review.<br><br>Close pull request and delete its branch: Close pull request that corresponds to a review and delete the respective branch. |
| **When Review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Close pull request: Close pull request that corresponds to a review.<br><br>Close pull request and delete its branch: Close pull request that corresponds to a review and delete the respective branch. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request,<br><br>• when a comment is added to commit which is a part of a pull request |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge pull request" or "Merge pull request and delete its branch". |

| | |
|---|---|
| | If enabled Collaborator will wait for the completed review to be signed off[193] before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Auto assign reviewers** | Whether to assign Collaborator reviewers when some specific users or teams were added as pull request reviewers or specified as code owners on the GitHub side and integration can match those GitHub users with Collaborator users. |
| **Webhook status** | Indicates current status of repository webhook:<br><br>Webhook is absent - A webhook is not created.<br><br>Webhook isn't active - A webhook is created, but is inactive.<br><br>Up and running - A webhook is active.<br><br>To create or activate a webhook, you can press the **Update webhook** button. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create a configuration for the GitHub repository and add webhook for that repository on GitHub side.

Now you need to link some Collaborator user account to the GitHub account[885] of repository owner.

To learn more about webhook settings on the GitHub side, see GitHub documentation:

https://developer.github.com/webhooks/

## Creating GitHub auto-polling configuration

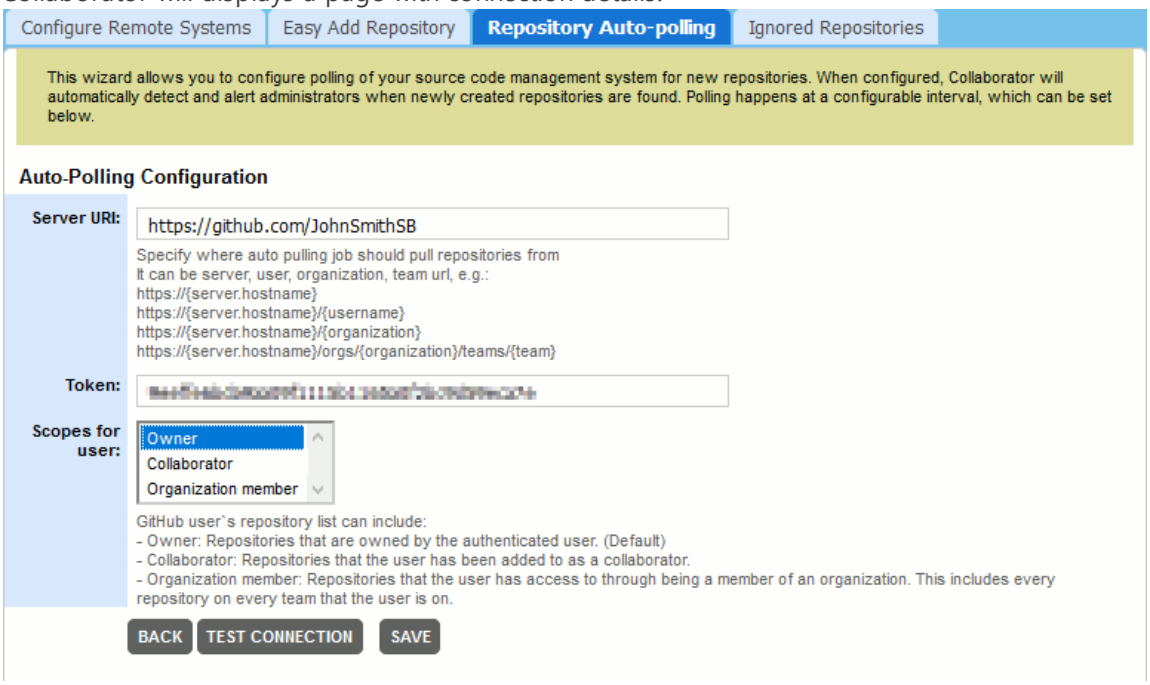
Sections above describe how to add integrations for existing repositories. When some new repositories are created on a server, you may integrate them manually. Alternatively, you can setup GitHub auto-polling configuration that will periodically look for new repositories on the specified server, organization, team or user and suggest creating integrations for them.

1. Log in to Collaborator as administrator. (To integrate with GitHub repositories, you need administrator privileges in Collaborator).

2. On the Collaborator main toolbar, click **ADMIN**, and then select **Repository Hosting Services** from the tree on the left. Then switch to the **Repository Auto-Polling** tab.

3. On the tab, select **GitHub** in the **Add configuration** box and click **Next**:

4. Collaborator will displays a page with connection details.




Fill in the edit boxes:

| Setting | Description |
|---|---|
| **Server URI** | Required. The URI of GitHub server, user, organization or team to be polled for new repositories. |
| **Token** | Required. The personal access token of the GitHub account.<br><br>Read Creating an access token for command-line use on GitHub documentation to learn how to obtain it. You will need to enable the following scopes: `repo` and `admin:org`. |
| **Scopes for user** | Optional. Defines which types of repositories to fetch.<br><br>Owner: Repositories that are owned by the specified user. (Default)<br><br>Collaborator: Repositories that the user has been added to as a collaborator.<br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user.<br><br>Organization member: Repositories that the user has access to through being a member of an organization. This includes every repository on every team that the user is on.<br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

5. After you specified the values, click **Save**. This will create an auto-polling configuration and display it in the **Auto-Polling Configurations List**.

6. Scroll down the **Repository Auto-Polling** tab and check that the **Enable Auto-Polling** setting is on and optionally change the **Auto-Polling Interval** setting.

Now Collaborator will automatically check if any new repositories were found on the specified server. Once found, it will notify administrators and suggest creating integrations with these newly created repositories via the Easy Add Repository wizard 841.

## Enable/Disable GitHub Integration

Once a new repository configuration is created, it is enabled automatically. However, you can enable and disable integration with GitHub servers manually. To do this:

1. Navigate to the **Admin > Remote System Integrations** screen.

2. Locate the **Enable GitHub Integration** setting and change it to **Yes** or **No**, respectively.

### 7.2.2 Configure Bitbucket Integration

There are two ways to create a Bitbucket configuration: automatically via the Easy Add Repository wizard and manually via the Configure Remote Systems tab. The first approach is useful for adding new configurations, while the second allows adding new and modifying existing configurations. This section describes both of these ways.

Additionally, you can create auto-polling configuration to periodically look for new repositories on Bitbucket server and suggest creating integrations for them.

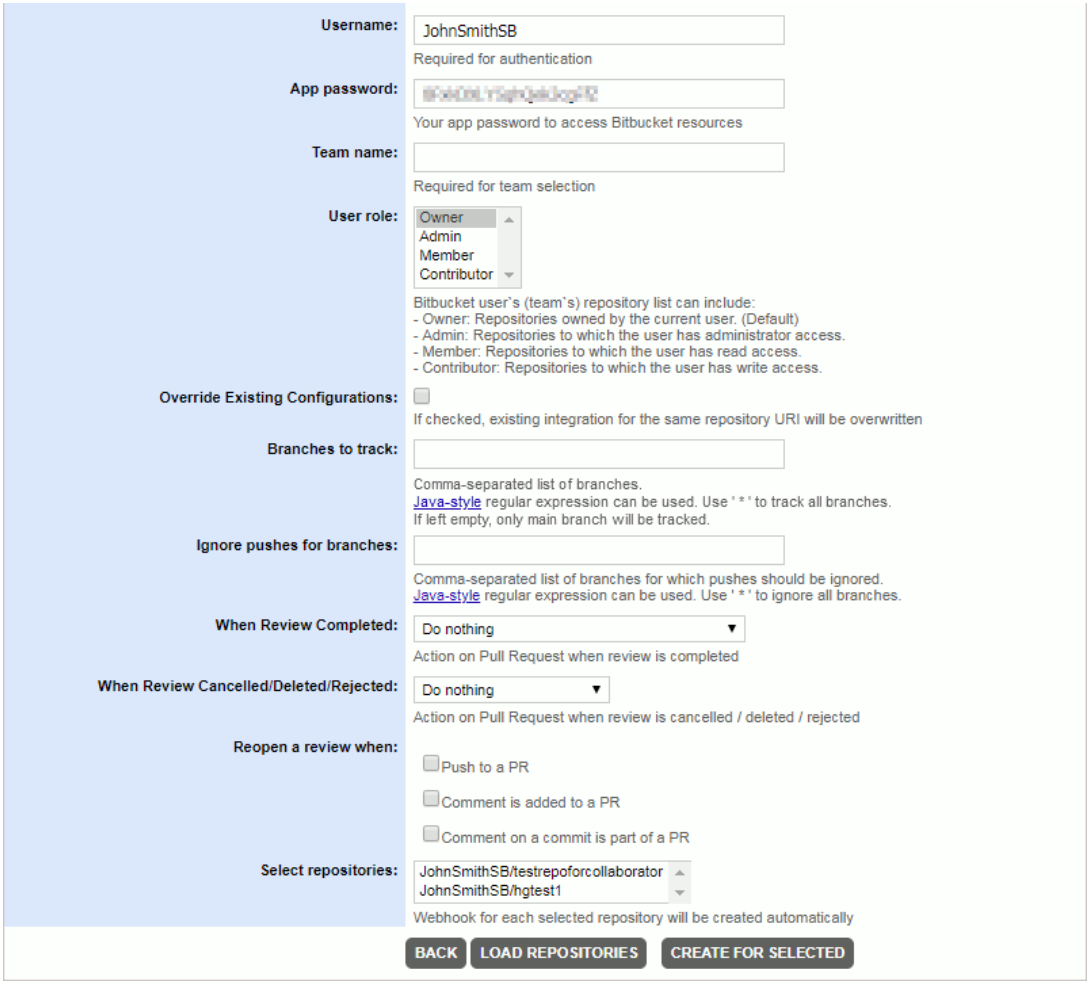
- Creating Bitbucket configuration via the Easy Add Repository wizard 851
- Creating Bitbucket configuration via the Configure Remote Systems tab 855
- Creating Bitbucket auto-polling configuration 857
- Enable/disable Bitbucket integration 859

**Important.** In order to use integration, your Collaborator server must be accessible to the remote system and vice versa. Configure a firewall or enable tunneled connections to expose your local Collaborator server to the Internet.

## Creating Bitbucket configuration via the Easy Add Repository wizard

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Easy Add Repository** tab.

4. Select *Bitbucket* in the the **Add repository for** combobox and click **Next**.
   This will display the Easy Add Repository wizard. It helps to setup remote repository configuration on the Collaborator side and add the webhook on the Bitbucket side.




5. Specify the following settings in the wizard:

| Setting | Description |
|---|---|
| **Username** | Required. Specifies the user account name for single-user repositories, or an account name to use for authentication in Team repositories. |
| **App password** | Required. The App password for the Bitbucket account to be tracked. |

| | |
|---|---|
| | Read App passwords on Bitbucket documentation to learn how to create App passwords. For integration with Collaborator you need to enable Read, Write, and Admin permissions for **Repositories** category and Read and Write permissions for **Pull requests** and **Webhooks** categories. |
| **Team name** | Optional. Required for team repositories. Denotes the team account name. |
| **User role** | Optional. Defines which types of repositories to track.<br><br>Owner: Repositories that are owned by the specified user. (Default)<br><br>Admin: Repositories to which the user has administrator access.<br><br>Member: Repositories to which the user has read access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user.<br><br>Contributor: Repositories to which the user has write access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |
| **Override existing configurations** | Optional. Specifies whether to override existing configurations that track the same repository URI. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, main branch in the repository will be tracked ("main" for Git repositories and "default" for Mercurial repositories). |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |

| | |
|---|---|
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and close its branch: Merge pull request that corresponds to a review and close the respective branch.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request,<br><br>• when a comment is added to commit which is a part of a pull request |

After specifying these values, you can click **Load repositories**. This will display a list of repositories available for the specified user. Select which of them to track and click **Create for selected**. To stop suggesting integrations for some specific repositories, select these repositories in the list and click **Ignore selected**.

Alternatively, you can click **Create for all** and add configurations for all repositories that are available for the specified user.

For every repository a separate configuration will be created in Collaborator and a webhook will be added in Bitbucket.

Now you need to link some Collaborator user account to the Bitbucket account 885 of repository owner.

## Creating Bitbucket configuration via the Configure Remote Systems tab

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Configure Remote Systems** tab.
4. In the **New Remote System Configuration** section, select *Bitbucket* and click **Create.**
5. Collaborator will display a page with configuration settings. Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | Required. The configuration name as it will be displayed in Collaborator's user interface. |
| **Bitbucket repo URI** | Required. The URI of Bitbucket repository to be tracked.<br><br>For instance: *https://bitbucket.org/jakenherman/awesome-java*<br><br>Both Mercurial and Git repositories are supported. You should copy the repository URI from the browser's address bar, rather than from the repository's clone URI. |
| **App password** | Required. The App password for the Bitbucket account to be tracked.<br><br>Read App passwords on Bitbucket documentation to learn how to create App passwords. For integration with Collaborator you need to enable Read, Write, and Admin permissions for **Repositories** category and Read and Write permissions for **Pull requests** and **Webhooks** categories. |
| **Username for authentication** | Required. Specifies an account name to use for authentication. For single-user repositories, account name is taken from the repo URI. |

| | |
|---|---|
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, main branch in the repository will be tracked ("main" for Git repositories and "default" for Mercurial repositories). |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and close its branch: Merge pull request that corresponds to a review and close the respective branch.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request, |

| | |
|---|---|
| | • when a comment is added to commit which is a part of a pull request |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge pull request" or "Merge pull request and delete its branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off[193] before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Webhook status** | Indicates current status of repository webhook:<br><br>Webhook is absent - A webhook is not created.<br><br>Webhook isn't active - A webhook is created, but is inactive.<br><br>Up and running - A webhook is active.<br><br>To create or activate a webhook, you can press the **Update webhook** button. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create a configuration for the Bitbucket repository and add webhook for that repository on Bitbucket side.

Now you need to link some Collaborator user account to the Bitbucket account[885] of repository owner.

To learn more about webhook settings on the Bitbucket side, see Bitbucket documentation:

https://confluence.atlassian.com/bitbucket/manage-webhooks-735643732.html

## Creating Bitbucket auto-polling configuration

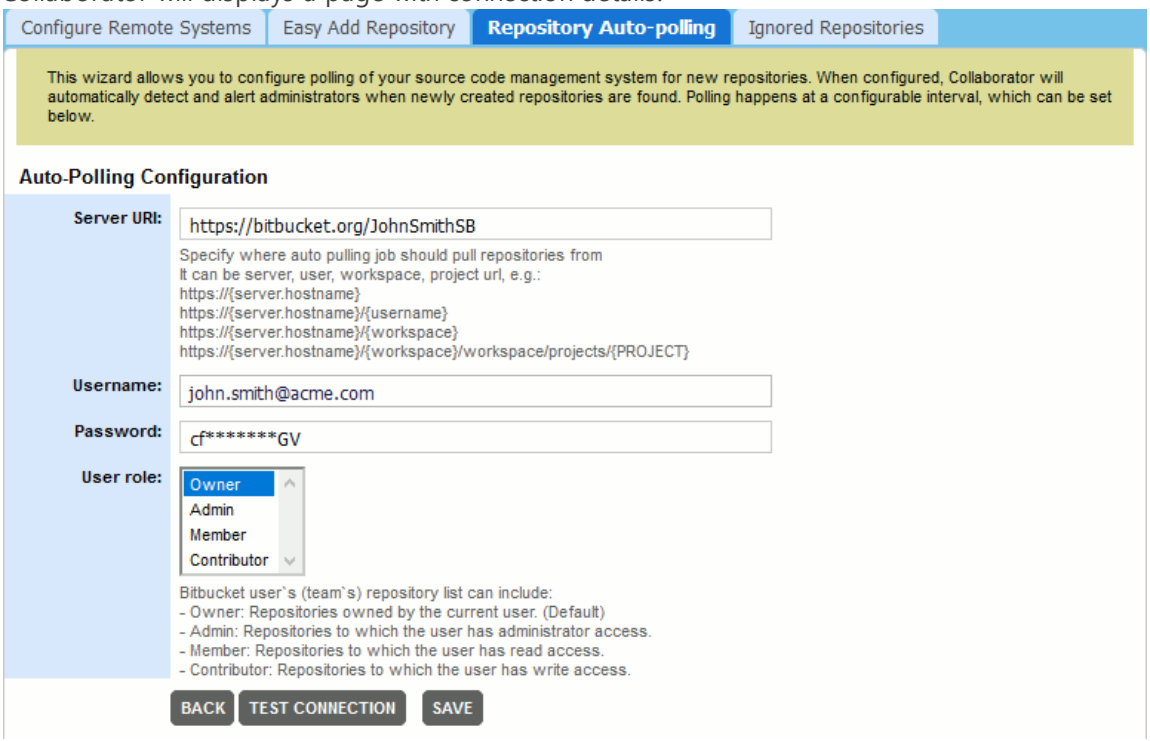
Sections above describe how to add integrations for existing repositories. When some new repositories are created on a server, you may integrate them manually. Alternatively, you can setup Bitbucket auto-polling configuration that will periodically look for new repositories on the specified server, user, workspace or project and suggest creating integrations for them.

1. Log in to Collaborator as administrator. (To integrate with Bitbucket repositories, you need administrator privileges in Collaborator).

2. On the Collaborator main toolbar, click **ADMIN**, and then select **Repository Hosting Services** from the tree on the left. Then switch to the **Repository Auto-Polling** tab.

3. On the tab, select **Bitbucket** in the **Add configuration** box and click **Next**:

4. Collaborator will displays a page with connection details.




Fill in the edit boxes:

| Setting | Description |
|---|---|
| **Server URI** | Required. The URI of Bitbucket server, user, workspace or project to be polled for new repositories. |
| **Username** | Required. Specifies the user account name for single-user repositories, or an account name to use for authentication in Team repositories. |
| **Password** | Required. The password for the Bitbucket account to be tracked. |
| **User role** | Optional. Defines which types of repositories to track.<br><br>Owner: Repositories that are owned by the specified user. (Default)<br><br>Admin: Repositories to which the user has administrator access.<br><br>Member: Repositories to which the user has read access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |

| | |
|---|---|
| | Contributor: Repositories to which the user has write access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

5. After you specified the values, click **Save**. This will create an auto-polling configuration and display it in the **Auto-Polling Configurations List**.

6. Scroll down the **Repository Auto-Polling** tab and check that the **Enable Auto-Polling** setting is on and optionally change the **Auto-Polling Interval** setting.

Now Collaborator will automatically check if any new repositories were found on the specified server. Once found, it will notify administrators and suggest creating integrations with these newly created repositories via the Easy Add Repository wizard 851.

## Enable/Disable Bitbucket Integration

Once a new repository configuration is created, it is enabled automatically. However, you can enable and disable integration with Bitbucket servers manually. To do this:

1. Navigate to the **Admin > Remote System Integrations** screen.

2. Locate the **Enable Bitbucket Integration** setting and change it to **Yes** or **No**, respectively.

### 7.2.3 Configure Bitbucket Server Integration

There are two ways to create a Bitbucket Server configuration: automatically via the Easy Add Repository wizard and manually via the Configure Remote Systems tab. The first approach is useful for adding new configurations, while the second allows adding new and modifying existing configurations. This section describes both of these ways.

Additionally, you can create auto-polling configuration to periodically look for new repositories on Bitbucket server and suggest creating integrations for them.

- Creating Bitbucket Server configuration via the Easy Add Repository wizard 860
- Creating Bitbucket Server configuration via the Configure Remote Systems tab 862
- Creating Bitbucket auto-polling configuration 865
- Enable/disable Bitbucket integration 866

**Important:** Do not confuse *Bitbucket Server (On-Premise)* and *Bitbucket* configurations. Bitbucket Server is a self-hosted version of Bitbucket.

**Important.** In order to use integration, your Collaborator server must be accessible to the remote system and vice versa. Configure a firewall or enable tunneled connections to expose your local Collaborator server to the Internet.

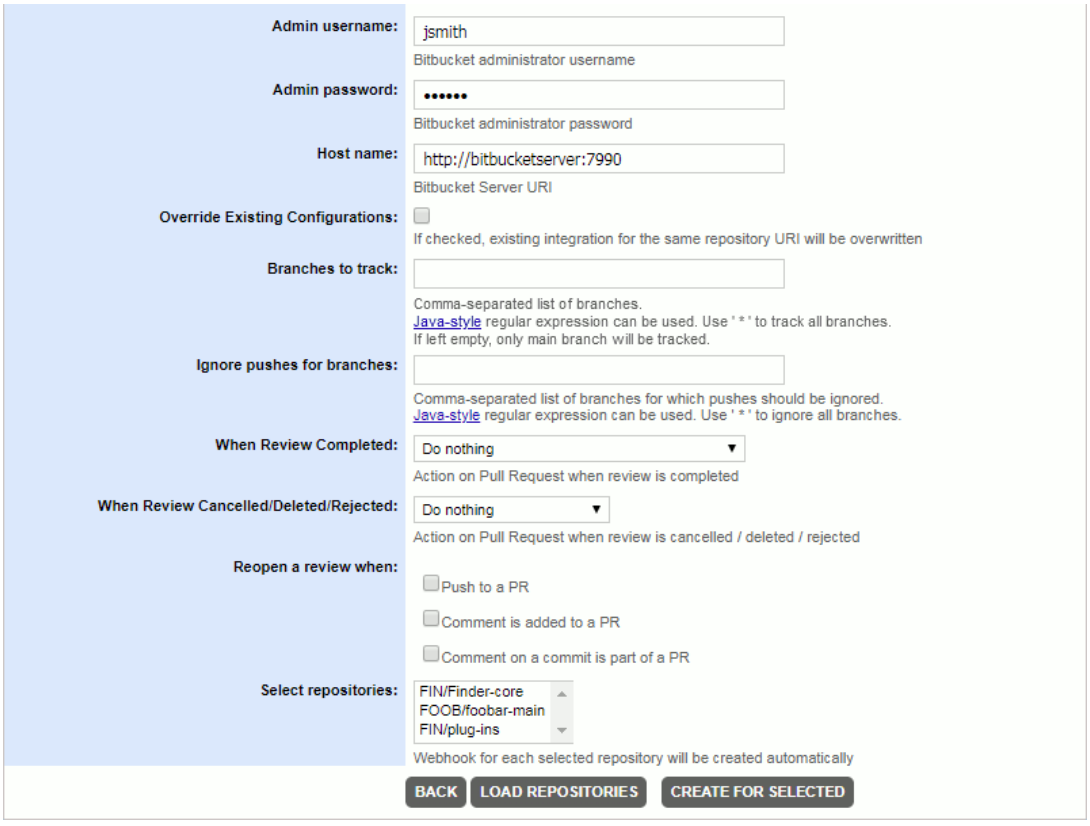
## Creating Bitbucket Server configuration via the Easy Add Repository wizard

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.

2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**

3. Switch to the **Easy Add Repository** tab.

4. Select *Bitbucket Server (On-Premise)* in the the **Add repository for** combobox and click **Next**. This will display the Easy Add Repository wizard. It helps to setup remote repository configuration on the Collaborator side and add the webhook on the Bitbucket Server side.




5. Specify the following settings in the wizard:

| Setting | Description |
|---|---|
| **Admin username** | Required. Username of Bitbucket Server administrator. |
| **Admin password** | Required. Password of Bitbucket Server administrator. |
| **Host name** | Required. Denotes the host name of your Bitbucket Server. |
| **Override existing configurations** | Optional. Specifies whether to override existing configurations that track the same repository URI. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, main branch in the repository will be tracked ("main" for Git repositories and "default" for Mercurial repositories). |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and close its branch: Merge pull request that corresponds to a review and close the respective branch.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected. |

| | |
|---|---|
| | Do nothing: Do not perform any action.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request,<br><br>• when a comment is added to commit which is a part of a pull request |

After specifying these values, you can click **Load repositories**. This will display a list of repositories available for the specified user. Select which of them to track and click **Create for selected**. To stop suggesting integrations for some specific repositories, select these repositories in the list and click **Ignore selected**.

Alternatively, you can click **Create for all** and add configurations for all repositories that are available for the specified user.

For every repository a separate configuration will be created in Collaborator and a webhook will be added in Bitbucket Server.

Now you need to link some Collaborator user account to the Bitbucket account[885] of repository owner.

## Creating Bitbucket configuration via the Configure Remote Systems tab

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Configure Remote Systems** tab.
4. In the **New Remote System Configuration** section, select *Bitbucket Server (On-Premise)* and click **Create.**
   **Important:** Do not confuse *Bitbucket Server (On-Premise)* and *Bitbucket* configurations. Bitbucket Server is a self-hosted version of Bitbucket.
5. Collaborator will display a page with configuration settings. Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | Required. The configuration name as it will be displayed in Collaborator's user interface. |
| **Bitbucket repo URI** | Required. The URI of Bitbucket repository to be tracked.<br><br>For instance: *http://hostname/projects/PROJ/repos/my_repo*<br><br>Both Mercurial and Git repositories are supported. You should copy the repository URI from the browser's address bar, rather than from the repository's clone URI. |
| **Admin username** | Required. Username of Bitbucket Server administrator. |
| **Admin password** | Required. Password of Bitbucket Server administrator. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, main branch in the repository will be tracked ("main" for Git repositories and "default" for Mercurial repositories). |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge pull request: Merge pull request that corresponds to a review.<br><br>Merge pull request and close its branch: Merge pull request that corresponds to a review and close the respective branch. |

| | |
|---|---|
| | Decline pull request: Decline pull request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Decline pull request: Decline pull request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request,<br><br>• when a comment is added to commit which is a part of a pull request |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge pull request" or "Merge pull request and delete its branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off[193] before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Webhook status** | Indicates current status of repository webhook:<br><br>Webhook is absent - A webhook is not created.<br><br>Webhook isn't active - A webhook is created, but is inactive.<br><br>Up and running - A webhook is active.<br><br>To create or activate a webhook, you can press the **Update webhook** button. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create a Bitbucket Server (On-Premise) configuration on the Collaborator side and add the webhook on the Bitbucket Server side.

Now you need to link some Collaborator user account to the Bitbucket account[885] of repository owner.

To learn more about webhook settings on the Bitbucket side, see Bitbucket documentation:

https://confluence.atlassian.com/bitbucket/manage-webhooks-735643732.html

## Creating Bitbucket auto-polling configuration

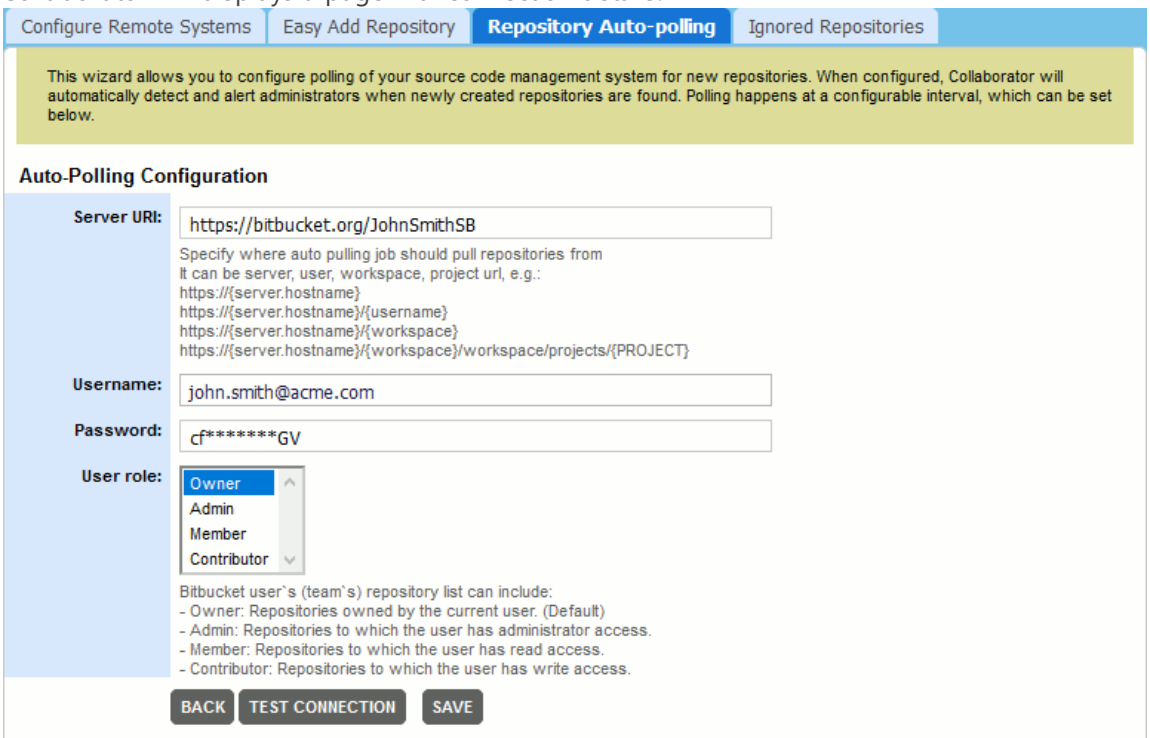
Sections above describe how to add integrations for existing repositories. When some new repositories are created on a server, you may integrate them manually. Alternatively, you can setup Bitbucket auto-polling configuration that will periodically look for new repositories on the specified server, user, workspace or project and suggest creating integrations for them.

1. Log in to Collaborator as administrator. (To integrate with Bitbucket repositories, you need administrator privileges in Collaborator).

2. On the Collaborator main toolbar, click **ADMIN**, and then select **Repository Hosting Services** from the tree on the left. Then switch to the **Repository Auto-Polling** tab.

3. On the tab, select **Bitbucket** in the **Add configuration** box and click **Next**:

4. Collaborator will displays a page with connection details.




Fill in the edit boxes:

| Setting | Description |
|---|---|
| **Server URI** | Required. The URI of Bitbucket server, user, workspace or project to be polled for new repositories. |

| | |
|---|---|
| **Username** | Required. Specifies the user account name for single-user repositories, or an account name to use for authentication in Team repositories. |
| **Password** | Required. The password for the Bitbucket account to be tracked. |
| **User role** | Optional. Defines which types of repositories to track.<br><br>Owner: Repositories that are owned by the specified user. (Default)<br><br>Admin: Repositories to which the user has administrator access.<br><br>Member: Repositories to which the user has read access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user.<br><br>Contributor: Repositories to which the user has write access.<br><br>Webhooks will be created for those repositories where allowed by the repository permissions for the specified user. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

5. After you specified the values, click **Save**. This will create an auto-polling configuration and display it in the **Auto-Polling Configurations List**.

6. Scroll down the **Repository Auto-Polling** tab and check that the **Enable Auto-Polling** setting is on and optionally change the **Auto-Polling Interval** setting.

Now Collaborator will automatically check if any new repositories were found on the specified server. Once found, it will notify administrators and suggest creating integrations with these newly created repositories via the Easy Add Repository wizard 860.

## Enable/Disable Bitbucket Integration

Once a new repository configuration is created, it is enabled automatically. However, you can enable and disable integration with Bitbucket servers manually. To do this:

1. Navigate to the **Admin > Remote System Integrations** screen.

2. Locate the **Enable Bitbucket Integration** setting and change it to **Yes** or **No**, respectively.

### 7.2.4 Configure GitLab Integration

There are two ways to create a GitLab configuration: automatically via the Easy Add Repository wizard and manually via the Configure Remote Systems tab. The first approach is useful for

adding new configurations, while the second allows adding new and modifying existing configurations. This section describes both of these ways.

Additionally, you can create auto-polling configuration to periodically look for new repositories on GitHub server and suggest creating integrations for them.

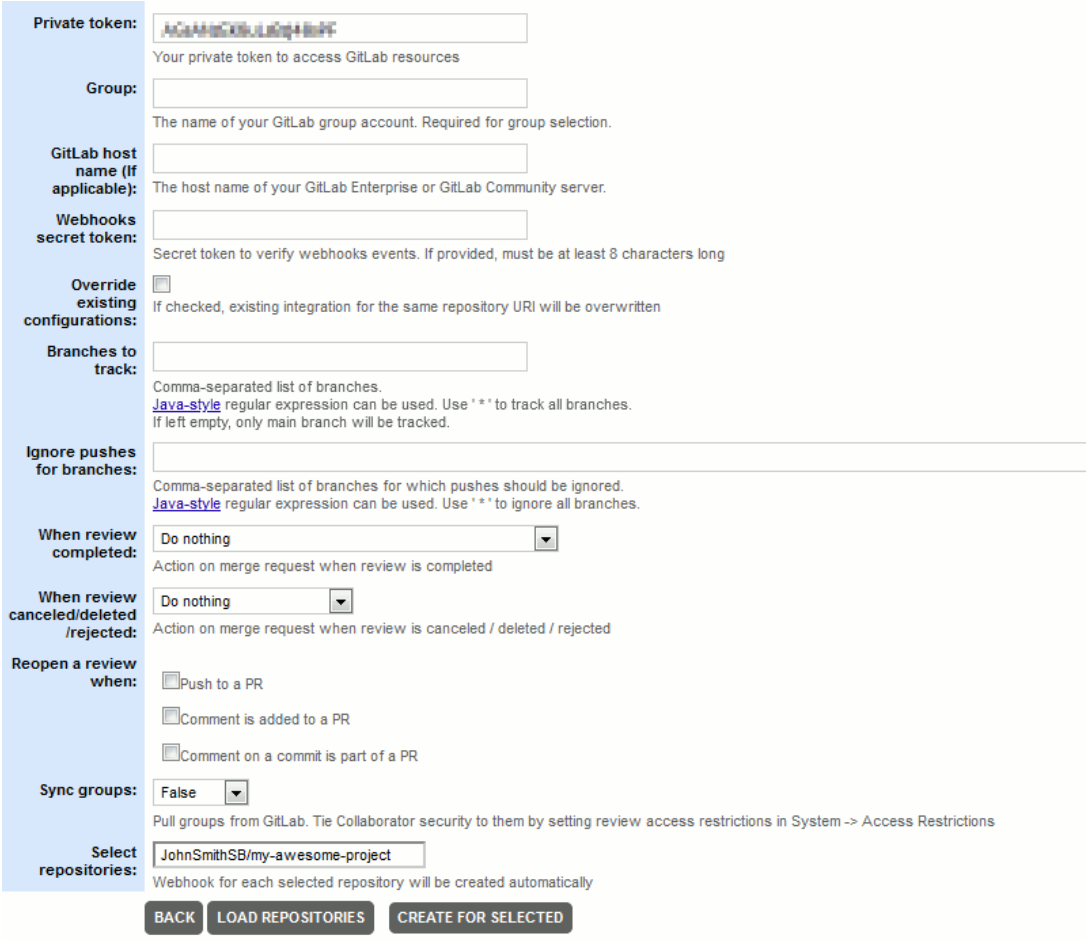
- Creating GitLab configuration via the Easy Add Repository wizard 867
- Creating GitLab configuration via the Configure Remote Systems tab 871

> **Important.** In order to use integration, your Collaborator server must be accessible to the remote system and vice versa. Configure a firewall or enable tunneled connections to expose your local Collaborator server to the Internet.

## Creating GitLab configuration via the Easy Add Repository wizard

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Easy Add Repository** tab.

4. Select *GitLab* in the the **Add repository for** combobox and click **Next**.
   This will display the Easy Add Repository wizard. It helps to setup remote repository configuration on the Collaborator side and add the webhook on the GitLab side.




5. Specify the following settings in the wizard:

| Setting | Description |
|---|---|
| **Private token** | Required. The private token for the GitLab account to be tracked.<br><br>You can copy it from the Access Tokens section of User Settings on GitLab. You will need to enable the `api` scope for this token. |
| **Group** | Required for group repositories. Denotes the account name of GitLab group. |
| **GitLab Community host name** | Required for repositories on self-hosted servers. Specifies the host name of your GitLab Enterprise or GitLab Community server. |

| | |
|---|---|
| **Webhooks secret token** | Optional. The secret token for webhook events. The secret token could be set when creating/modifying a webhook on GitLab server. If provided, must be at least 8 characters long.<br><br>To learn more about webhook settings on the GitLab side, see GitLab documentation:<br><br>https://gitlab.com/help/user/project/integrations/webhooks.md |
| **Override existing configurations** | Optional. Specifies whether to override existing configurations that track the same repository URI. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, "main" branch will be tracked. |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a merge request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge the merge request: Merge the request that corresponds to a review.<br><br>Merge the merge request and delete source branch: Merge the request that corresponds to a review and delete the respective branch.<br><br>Close merge request: Close merge request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a merge request was cancelled, deleted or rejected. |

| | |
|---|---|
| | Do nothing: Do not perform any action.<br><br>Close merge request: Close merge request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a merge request is made,<br><br>• when a comment is added to a merge request,<br><br>• when a comment is added to commit which is a part of a merge request |
| **Sync groups** | Specifies whether Collaborator should synchronize its native groups with group information from the GitLab server. Effective if the global **Enable Group Sync** 216 setting is turned on.<br><br>Once enabled, on every logging-in, Collaborator will check user membership in groups on the GitLab server, create new groups (if needed), and automatically add this user to the corresponding groups on the Collaborator server. See Syncing Groups and Their Members 239 for details.<br><br>To ensure that Collaborator users do not have access to reviews that contain code from the repositories that they do not have access to, use the Restrict Access to Review Content 188 settings. |

After specifying these values, you can click **Load repositories**. This will display a list of repositories available for the specified user or group. Select which of them to track and click **Create for selected**. To stop suggesting integrations for some specific repositories, select these repositories in the list and click **Ignore selected**.

Alternatively, you can click **Create for all** and add configurations for all repositories that are available for the specified user or group.

For every repository a separate configuration will be created in Collaborator and a webhook will be added in GitLab.

Now you need to link some Collaborator user account to the GitLab account 885 of repository owner.

## Creating GitLab configuration via the Configure Remote Systems tab

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Configure Remote Systems** tab.
4. In the **New Remote System Configuration** section, select *GitLab* and click **Create.**
5. Collaborator will display a page with configuration settings. Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | Required. The configuration name as it will be displayed in Collaborator's user interface. |
| **GitLab repo URI** | Required. The URI of GitLab repository to be tracked.<br><br>For instance: *https://gitlab.com/gitlab-org/gitlab-ee.git*<br><br>You can copy it from the Clone URI field of the repository's main page on GitLab. |
| **Private token** | Required. The private token for the GitLab account to be tracked.<br><br>You can copy it from the Access Tokens section of User Settings on GitLab. You will need to enable the `api` scope for this token. |
| **Webhooks Secret token** | Optional. The secret token for webhook events. The secret token could be set when creating/modifying a webhook on GitLab server. If provided, must be at least 8 characters long.<br><br>To learn more about webhook settings on the GitLab side, see GitLab documentation:<br><br>https://gitlab.com/help/user/project/integrations/webhooks.md |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on merge requests and direct pushes. Separate multiple branch names with commas. |

| | |
|---|---|
| | You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>If this field is empty, "main" branch will be tracked. |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a merge request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Merge the merge request: Merge the request that corresponds to a review.<br><br>Merge the merge request and delete source branch: Merge the request that corresponds to a review and delete the respective branch.<br><br>Close merge request: Close merge request that corresponds to a review. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a merge request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Close merge request: Close merge request that corresponds to a review. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a merge request is made,<br><br>• when a comment is added to a merge request,<br><br>• when a comment is added to commit which is a part of a merge request |

| | |
|---|---|
| **Sync groups** | Specifies whether Collaborator should synchronize its native groups with group information from the GitLab server. Effective if the global **Enable Group Sync** 216 setting is turned on.<br><br>Once enabled, on every logging-in, Collaborator will check user membership in groups on the GitLab server, create new groups (if needed), and automatically add this user to the corresponding groups on the Collaborator server. See Syncing Groups and Their Members 239 for details.<br><br>To ensure that Collaborator users do not have access to reviews that contain code from the repositories that they do not have access to, use the Restrict Access to Review Content 188 settings. |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge the merge request" or "Merge the merge request and delete source branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off 193 before merging the respective request. Otherwise, merge request will be merged immediately, even if the review have not been signed yet. |
| **Webhook status** | Indicates current status of repository webhook:<br><br>Webhook is absent - A webhook is not created.<br><br>Webhook isn't active - A webhook is created, but is inactive.<br><br>Up and running - A webhook is active.<br><br>To create or activate a webhook, you can press the **Update webhook** button. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create a configuration for the GitLab repository and add webhook for that repository on GitLab side.

Now you need to link some Collaborator user account to the GitLab account 885 of repository owner.

To learn more about webhook settings on the GitLab side, see GitLab documentation:

https://gitlab.com/help/user/project/integrations/webhooks.md

## Creating GitLab auto-polling configuration

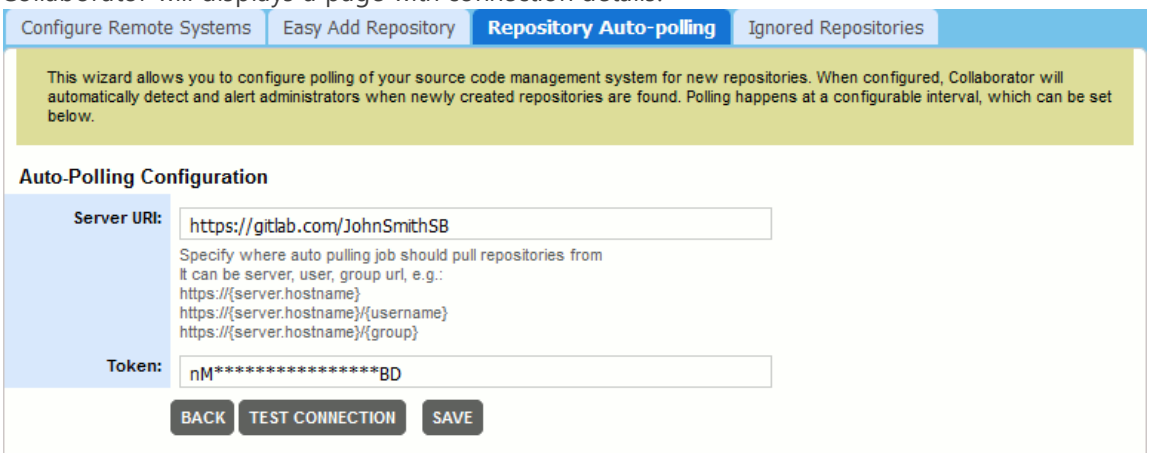
Sections above describe how to add integrations for existing repositories. When some new repositories are created on a server, you may integrate them manually. Alternatively, you can setup GitLab auto-polling configuration that will periodically look for new repositories on the specified server, user or group and suggest creating integrations for them.

1. Log in to Collaborator as administrator. (To integrate with GitLab repositories, you need administrator privileges in Collaborator).

2. On the Collaborator main toolbar, click **ADMIN**, and then select **Repository Hosting Services** from the tree on the left. Then switch to the **Repository Auto-Polling** tab.

3. On the tab, select **GitLab** in the **Add configuration** box and click **Next**:

4. Collaborator will displays a page with connection details.




Fill in the edit boxes:

| Setting | Description |
|---|---|
| **Server URI** | Required. The URI of GitLab server, user or group to be polled for new repositories. |
| **Token** | Required. The private token for the GitLab account to be tracked.<br><br>You can copy it from the Access Tokens section of User Settings on GitLab. You will need to enable the `api` scope for this token. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

5. After you specified the values, click **Save**. This will create an auto-polling configuration and display it in the **Auto-Polling Configurations List**.

6. Scroll down the **Repository Auto-Polling** tab and check that the **Enable Auto-Polling** setting is on and optionally change the **Auto-Polling Interval** setting.

Now Collaborator will automatically check if any new repositories were found on the specified server. Once found, it will notify administrators and suggest creating integrations with these newly created repositories via the Easy Add  Repository wizard 867.

## Enable/Disable GitLab Integration

Once a new repository configuration is created, it is enabled automatically. However, you can enable and disable integration with GitLab servers manually. To do this:

1. Navigate to the **Admin > Remote System Integrations** screen.

2. Locate the **Enable GitLab Integration** setting and change it to **Yes** or **No**, respectively.

### 7.2.5 Configure Azure DevOps Integration

There are two ways to create a Azure DevOps configuration: automatically via the Easy Add Repository wizard and manually via the Configure Remote Systems tab. The first approach is useful for adding new configurations, while the second allows adding new and modifying existing configurations. This section describes both of these ways.

Additionally, you can create auto-polling configuration to periodically look for new repositories on Azure DevOps server and suggest creating integrations for them.

- Creating Azure DevOps configuration via the Easy Add Repository wizard 841
- Creating Azure DevOps configuration via the Configure Remote Systems tab 845
- Creating Azure DevOps auto-polling configuration 883
- Enable/disable Azure DevOps integration 884

> **Important.** In order to use integration, your Collaborator server must be accessible to the remote system and vice versa. Configure a firewall or enable tunneled connections to expose your local Collaborator server to the Internet.
> Team Foundation Version Control (TFVC) repositories are not supported.

## Creating Azure DevOps configuration via the Easy Add Repository wizard

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.

2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**

3. Switch to the **Easy Add Repository** tab.

4. Select *Azure DevOps Git* in the the **Add repository for** combobox and click **Next**.
   This will display the Easy Add Repository wizard. It helps to setup remote repository configuration on the Collaborator side and add the webhook on the Azure DevOps side.


5. Specify the following settings in the wizard:

| Setting | Description |
|---|---|
| **Host name** | Optional. Specifies the host name where your Azure DevOps Server resides. For example `dev.azure.com`, `myorg.visualstudio.com`, or URI of your on-premise server.<br><br>If host name is omitted, `dev.azure.com` will be used. |
| **Username** | Required. Specifies the account name to use for authentication. |

| | |
|---|---|
| **Personal access token** | Required. The personal access token for a Azure DevOps Server account that has access to your remote repository.<br><br>Read Authenticate access with personal access tokens on Microsoft documentation to learn how to obtain it.<br><br>You will need to enable the following scopes: `Code:Full`, `Code:Status` and `Work Items:Read, write, & manage`. |
| **Organization name** | Specifies the name of your organization account. Required for `dev.azure.com` and on-premise hosts. |
| **Override existing configurations** | Optional. Specifies whether to override existing configurations that track the same repository URI. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. Note that wildcards and regular expressions cannot be used if you enable the Status check required setting (see below).<br><br>If this field is empty, "main" branch will be tracked. |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches.<br><br>Note that wildcards and regular expressions cannot be used if you enable the Status check required setting (see below). |
| **Status check required** | Specifies if Collaborator should enforce status checks for the tracked branches before merging pull requests.<br><br>If this option is enabled, you cannot use regular expressions in the *Branches to track* setting, since this option requires exact name match. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected. |

| | |
|---|---|
| | Do nothing: Do not perform any action.<br><br>Abandon pull request: Abandon pull request that corresponds to a review.<br><br>Abandon pull request and delete its branch: Abandon pull request that corresponds to a review and delete the respective branch. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Complete pull request: Complete pull request that corresponds to a review.<br><br>Complete pull request and delete its branch: Complete pull request that corresponds to a review and delete the respective branch.<br><br>Abandon pull request: Abandon pull request that corresponds to a review.<br><br>Abandon pull request and delete its branch: Abandon pull request that corresponds to a review and delete the respective branch. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following:<br><br>• when a push to a pull request is made,<br><br>• when a comment is added to a pull request. |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge the pull request" or "Merge the pull request and delete its branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off[193] before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Auto close related work items** | Automatically complete related work items when pull request is merged. |

| | |
|---|---|
| **Auto assign reviewers** | Whether to assign Collaborator reviewers when some specific users were added as pull request reviewers on the Azure DevOps side and integration can match those Azure DevOps users with Collaborator users. |

After specifying these values, you can click **Load repositories**. This will display a list of repositories available for the specified user or organization. Select which of them to track and click **Create for selected**. To stop suggesting integrations for some specific repositories, select these repositories in the list and click **Ignore selected**.

Alternatively, you can click **Create for all** and add configurations for all repositories that are available for the specified user or organization.

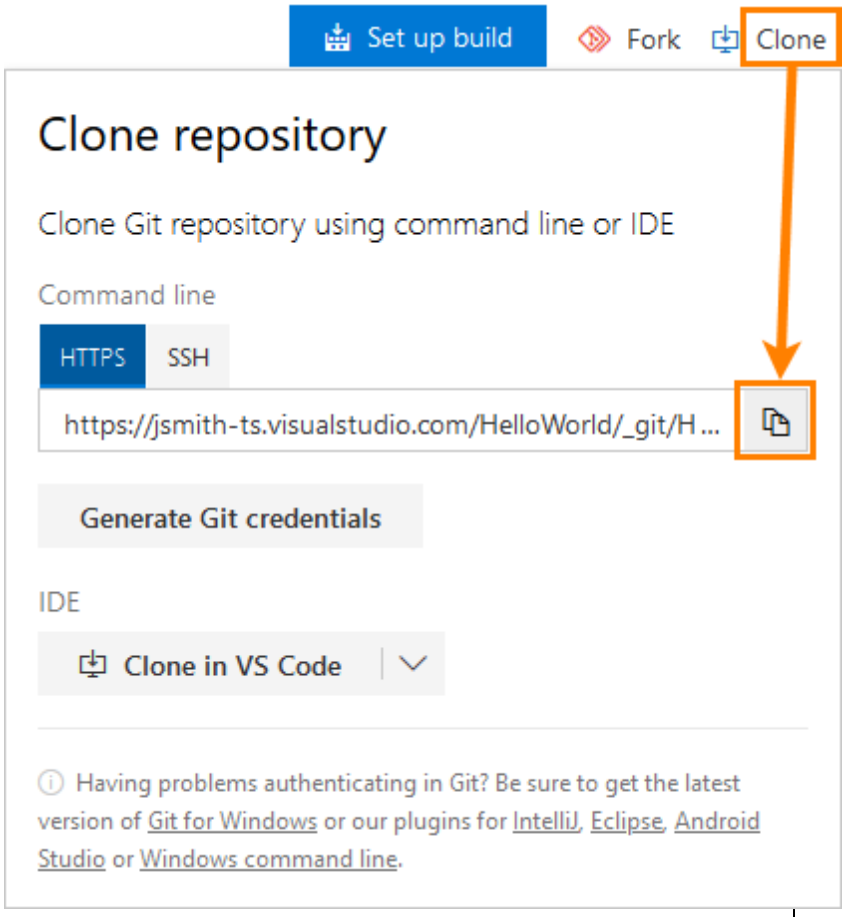
For every repository a separate configuration will be created in Collaborator and a webhook will be added in Azure DevOps.

Now you need to link some Collaborator user account to the Azure DevOps account 885 of repository owner.

## Creating Azure DevOps configuration via the Configure Remote Systems tab

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Repository Hosting Services**
3. Switch to the **Configure Remote Systems** tab.
4. In the **New Remote System Configuration** section, select *Azure DevOps Git* and click **Create.**
5. Collaborator will display a page with configuration settings. Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | Required. The configuration name as it will be displayed in Collaborator's user interface. |
| **Azure DevOps repository URI** | Required. The URI of the Azure DevOps repository to be tracked, for example, https://dev.azure.com/myorg/myproject/_git/myrepo or https://myorg.visualstudio.com/myproject/_git/myrepo.<br><br>You can copy the URI from the repository page in Azure DevOps Server: |

| | |
|---|---|
| |  |
| **Username** | Required. Specifies the account name to use for authentication. |
| **Personal access token** | Required. The personal access token for a Azure DevOps Server account that has access to your remote repository.<br><br>Read Authenticate access with personal access tokens on Microsoft documentation to learn how to obtain it.<br><br>You will need to enable the following scopes: `Code:Full`, `Code: Status` and `Work Items:Read, write, & manage`. |
| **Branches to track** | Optional. The names of branches to track changes and create reviews on pull requests and direct pushes. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. Note that wildcards and regular expressions cannot be used if you enable the Status check required setting (see below). |

| | |
|---|---|
| | If this field is empty, "main" branch will be tracked. |
| **Ignore pushes for branches** | Optional. Specifies branches for which Collaborator will not create reviews on direct pushes. You can enter one or several branch names. Separate multiple branch names with commas.<br><br>You may use Java-style regular expressions to match specific branch names, or you may use the '*' wildcard (alone, or separated by comma) to match all branches. Note that wildcards and regular expressions cannot be used if you enable the Status check required setting (see below). |
| **Status check required** | Specifies if the integration should enforce status checks for the tracked branches before merging pull request.<br><br>If this option is enabled, you cannot use regular expressions in the "Branches to track" setting, since this option requires exact name match. |
| **When review completed** | Optional. Specifies what action to perform when a review corresponding to a pull request was accomplished.<br><br>Do nothing: Do not perform any action.<br><br>Complete pull request: Complete pull request that corresponds to a review.<br><br>Complete pull request and delete its branch: Complete pull request that corresponds to a review and delete the respective branch.<br><br>Abandon pull request: Abandon pull request that corresponds to a review.<br><br>Abandon pull request and delete its branch: Abandon pull request that corresponds to a review and delete the respective branch. |
| **When review cancelled/deleted/ rejected** | Optional. Specifies what action to perform when a review corresponding to a pull request was cancelled, deleted or rejected.<br><br>Do nothing: Do not perform any action.<br><br>Abandon pull request: Abandon pull request that corresponds to a review.<br><br>Abandon pull request and delete its branch: Abandon pull request that corresponds to a review and delete the respective branch. |
| **Reopen a review when** | Optional. Specifies in what cases Collaborator should reopen completed reviews. May include any combination of the following: |

| | |
|---|---|
| | • when a push to a pull request is made,<br>• when a comment is added to a pull request. |
| **Wait for signature (if enabled) before merge** | Optional. Effective, if "When review completed" setting is set to "Merge pull request" or "Merge pull request and delete its branch".<br><br>If enabled Collaborator will wait for the completed review to be signed off[193] before merging the respective pull request. Otherwise, pull request will be merged immediately, even if the review have not been signed yet. |
| **Auto close related work items** | Automatically complete related work items when pull request is merged. |
| **Auto assign reviewers** | Whether to assign Collaborator reviewers when some specific users were added as pull request reviewers on the Azure DevOps side and integration can match those Azure DevOps users with Collaborator users. |
| **Webhook status** | Indicates current status of repository webhook:<br><br>Webhook is absent - A webhook is not created.<br><br>Webhook isn't active - A webhook is created, but is inactive.<br><br>Up and running - A webhook is active.<br><br>To create or activate a webhook, you can press the **Update webhook** button. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create a configuration for the Azure DevOps repository and add webhook for that repository on Azure DevOps side.

Now you need to link some Collaborator user account to the Azure DevOps account[885] of repository owner.

To learn more about webhook settings on the Azure DevOps side, see Microsoft documentation:

https://docs.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops

## Creating Azure DevOps auto-polling configuration

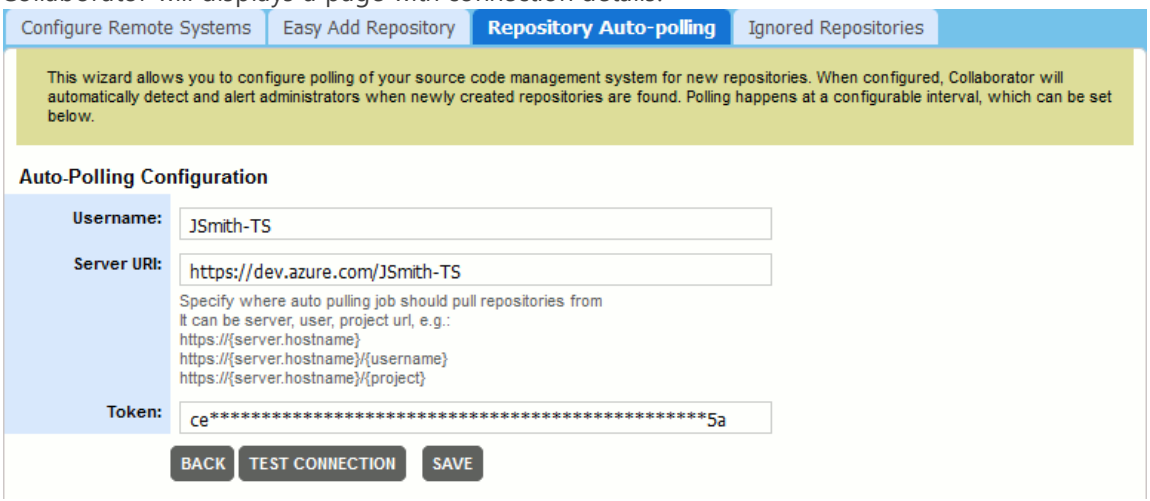
Sections above describe how to add integrations for existing repositories. When some new repositories are created on a server, you may integrate them manually. Alternatively, you can setup Azure DevOps auto-polling configuration that will periodically look for new repositories on the specified server, user or project and suggest creating integrations for them.

1. Log in to Collaborator as administrator. (To integrate with Azure DevOps repositories, you need administrator privileges in Collaborator).

2. On the Collaborator main toolbar, click **ADMIN**, and then select **Repository Hosting Services** from the tree on the left. Then switch to the **Repository Auto-Polling** tab.

3. On the tab, select **Azure DevOps** in the **Add configuration** box and click **Next**:

4. Collaborator will displays a page with connection details.




Fill in the edit boxes:

| Setting | Description |
|---|---|
| **Username** | Required. Specifies the account name to use for authentication. |
| **Server URI** | Required. The URI of GitHub server, user or project to be polled for new repositories. |
| **Token** | Required. The personal access token for a Azure DevOps Server account.<br><br>Read Authenticate access with personal access tokens on Microsoft documentation to learn how to obtain it.<br><br>You will need to enable the following scopes: `Code:Full`, `Code:Status` and `Work Items:Read, write, & manage`. |

After specifying these values, you can click **Test connection** to verify if you entered data correctly.

5. After you specified the values, click **Save**. This will create an auto-polling configuration and display it in the **Auto-Polling Configurations List**.

6. Scroll down the **Repository Auto-Polling** tab and check that the **Enable Auto-Polling** setting is on and optionally change the **Auto-Polling Interval** setting.

Now Collaborator will automatically check if any new repositories were found on the specified server. Once found, it will notify administrators and suggest creating integrations with these newly created repositories via the Easy Add Repository wizard 875.

## Enable/Disable Azure DevOps Integration

Once a new repository configuration is created, it is enabled automatically. However, you can enable and disable integration with Azure DevOps servers manually. To do this:

1. Navigate to the **Admin > Remote System Integrations** screen.

2. Locate the **Enable Azure DevOps Git Integration** setting and change it to **Yes** or **No**, respectively.

### 7.2.6 Establish Trusted Connection

When your on-premises hosting service (GitHub Enterprise, GitLab Enterprise, Bitbucket Server or Azure DevOps Server) use SSL connection, its certificates may not be trusted by Collaborator server. In this case you will need to import that certificate as trusted.

To establish trust, you need to import the public key of your on-premises server as a trusted certificate to Collaborator keystore file:

1. Get the certificate file from your on-premises server or network administrator.

2. Locate the keystore file which you have generated while configuring Collaborator HTTPS connection 112.

Default location is `<Collaborator Server>/tomcat/conf/collab.ks`, yet that could be changed while generating keystore.

3. Use Java's keytool utility to import the certificate in Collaborator keystore file. You can find keytool in the `$JAVA_HOME/bin` directory:

```
$JAVA_HOME/bin/keytool -import -alias on-prem-server -keystore
<collab-keystore-path> -trustcacerts -file <certificate-file-name>
```

For more information on command-line arguments of the keytool utility, see keytool documentation.

4. Most likely you will be prompted to confirm the validity of the certificate. It is imperative for the security of the overall system that you verify the key matches the trusted material. Before accepting the certificate, you should contact the administrator that sent you the certificates and verify that the certificate fingerprints that you see match the certificate fingerprints that they intended to send you.

5. The final step is to configure Collaborator to use the newly created keystore. Open the `<Collaborator Server>/ccollab-server.vmoptions` file in a text editor, and add the following lines to it:

```
-Djavax.net.ssl.trustStore=<collab-keystore-path>
-Djavax.net.ssl.trustStorePassword=<collab-keystore-password>
```

6. Restart the Collaborator server.

### 7.2.7 Link User Accounts

Collaborator automatically creates reviews 826 for changes made to web-based repositories 840. At that, it can automatically make the person, who is changing files in the repo, an author 279 of the review. The problem is that by default Collaborator has no information about source control accounts.

You and your teammates can link source-control accounts with their Collaborator accounts. After that, when creating a new review, Collaborator will be able to find a respective Collaborator user, will make that user the creator of the review and will assign an author role to that use in the review.

If Collaborator fails to match the repository user and a Collaborator user, it will make the Collaborator Administrator the creator and author of the new review.

#### How do you link accounts?

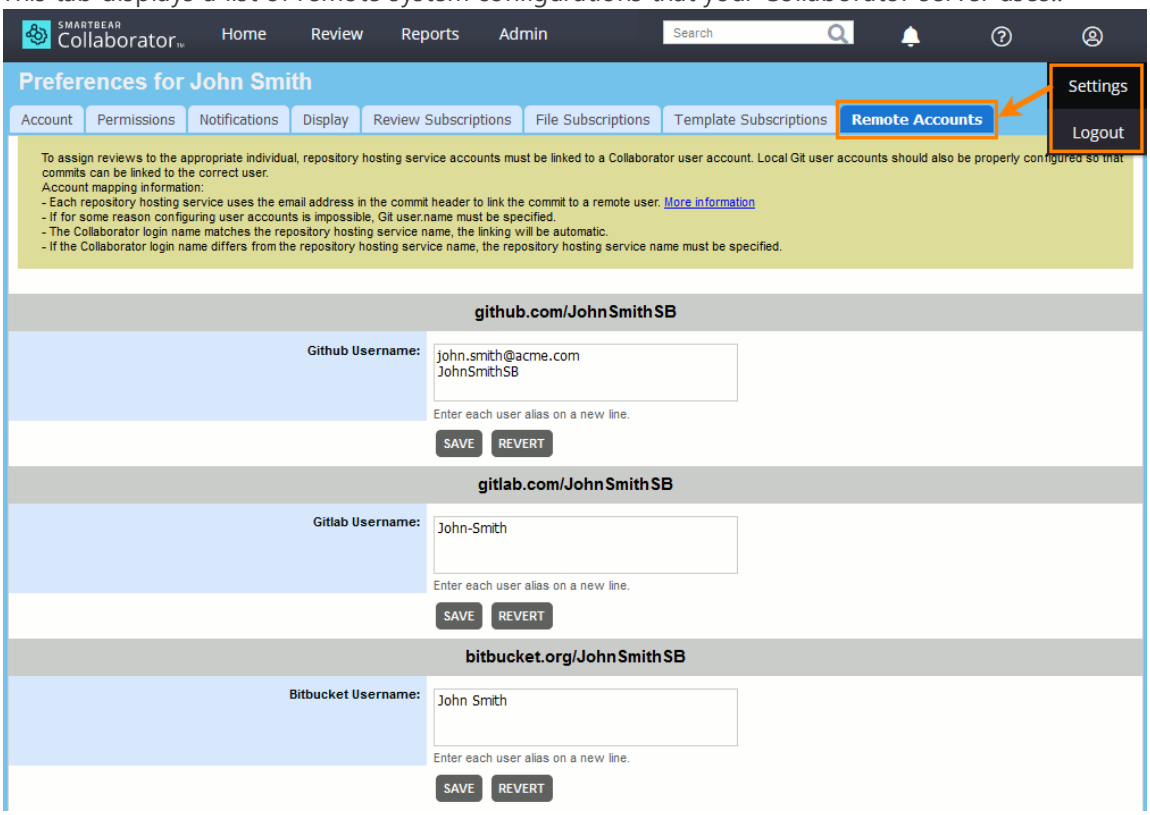
Make sure, that you have specified your user name and email address on your local machine (either globally, or for specific repository). To learn how to do this, see Git or Mercurial instructions, respectively.

Specify the same user name and email address in your remote repository profile.

If the **name or email address** of a Collaborator user account 317 coincides with the name or email address of a source-control user account, Collaborator will link these two accounts automatically.

If the names differ, then you should link accounts manually:

1. In Collaborator, go to **Settings** and then switch to the **Remote Accounts** tab.
   This tab displays a list of remote system configurations that your Collaborator server uses.:




2. Specify your account names for those web-based repositories, on which you have accounts. To specify several names or e-mails, separate them by comma.
   **For Bitbucket users:** enter your Bitbucket name exactly as it is specified in the Name field of the Bitbucket settings.
   **For GitLab users:** enter your GitLab name exactly as it is specified in the Full Name field of the Profile settings.
   Press **Save** to apply the changes.

# 8 Issue-Tracking Integrations

Collaborator can integrate with any external issue-tracking systems such as JIRA, TeamTrack, Bugzilla, FogBugz and so on. This gives you the possibility to associate issues with reviews and vice versa, hyperlink mentioned issues, move or mirror defects from Collaborator to your issue-tracking system.

The integration saves the time and efforts needed to synchronize your issue-tracking items and Collaborator reviews, and helps you be concentrated on your business tasks rather than on the integration procedures.

## In This Section



# 8.1 Issue-Tracking Integrations: Overview

Collaborator provides advanced support for JIRA issue-tracking and Team Foundation Server work item systems, giving you the possibility to easily integrate these products into your work processes. It automatically synchronies reviews and the issues/work items addressed in those reviews, appends the issues to the review's Remote System Links section and retrieves the item's current status.



For other issue-tracking systems, your Collaborator administrator should establish integration as described in General Approach to Issue-Tracking Integrations 906.

## Requirements

**Server versions:**

- JIRA server 5.0 and higher (SaaS or on-prem, including Atlassian Data Center).

- Team Foundation Server 2010 and higher.

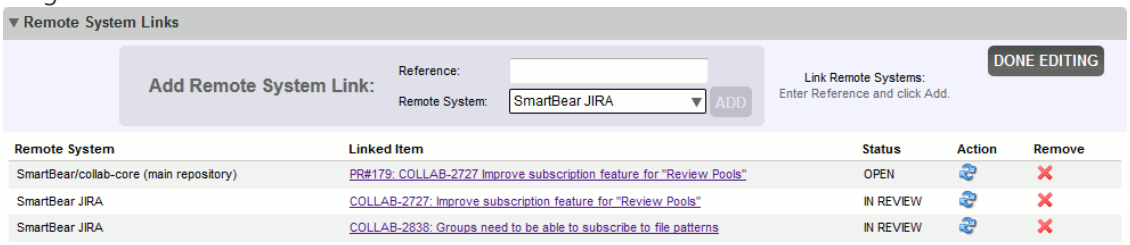
**Setup and configure integration** - Your Collaborator administrator should create a new Remote System Integration configuration for your issue-tracking/work item system. The configuration defines what server to use, access credentials, list of projects to track and so on. Each single configuration instance tracks changes on a single server. To track multiple issue-tracker/work item servers, your administrators should create a separate configuration for each server. For detailed instructions on how to setup integration, see Configuring Issue-Tracking Integrations 896.

## How Issue-Tracking Integration Works

Collaborator checks if the uploaded review materials are associated with any tickets/work items. Depending on system, Collaborator either retrieves this information from the changelist service data or parses review title, custom fields, changelist comments, comments and defects to match ticket/work item identifiers.

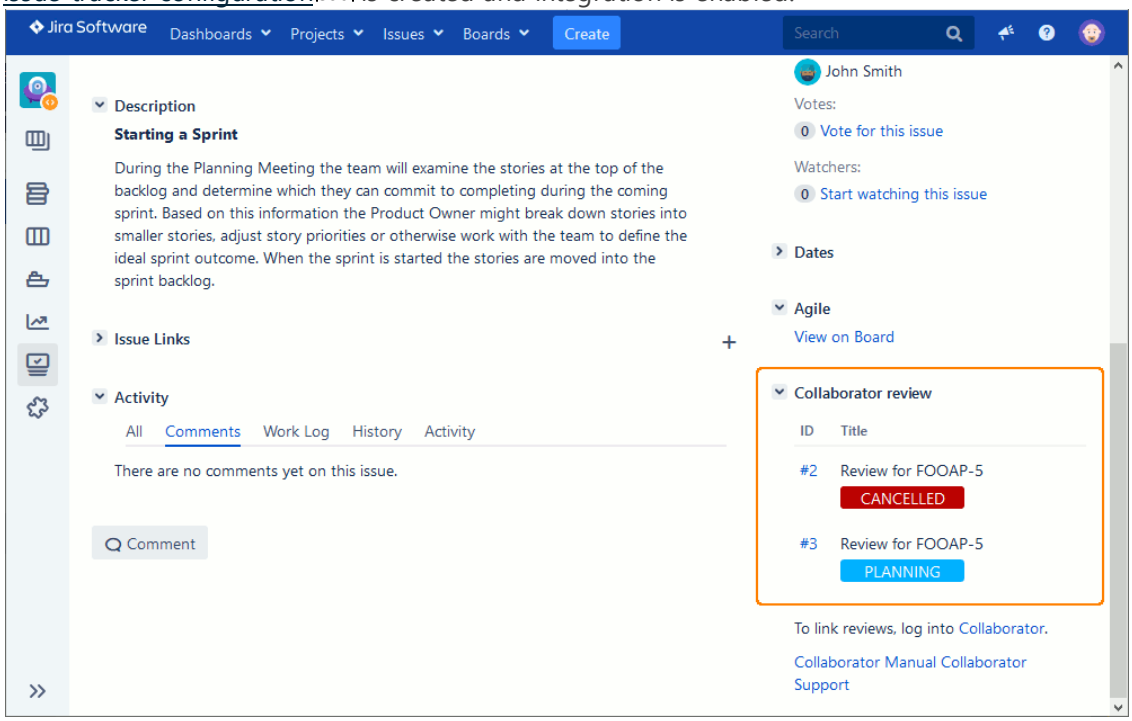
Once Collaborator finds that the review materials are associated with some ticket or work item, it performs the following:

- **Links ticket or work item to a review.** All linked tickets/work items, as well as their current statuses, will be displayed in the review's Remote System Links 353 section. This feature is enabled automatically, when the respective issue-tracker configuration 896 is created and integration is enabled.

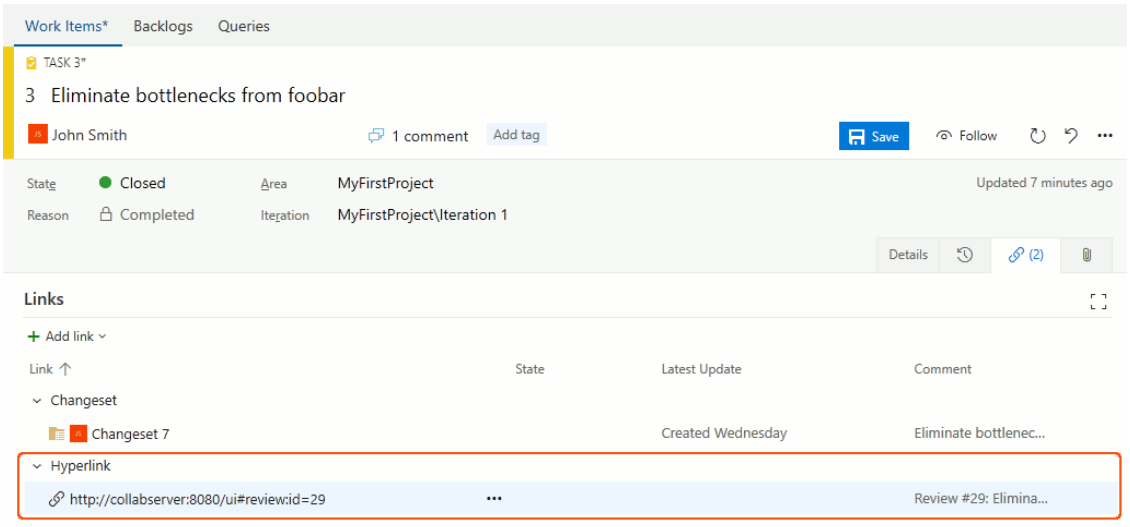



*Review with links to JIRA tickets*

- **Links review to a ticket or work item.** A link will be appended to a ticket/work item, if an issue-tracking system permits it. This feature is enabled automatically, when the respective issue-tracker configuration[896] is created and integration is enabled.

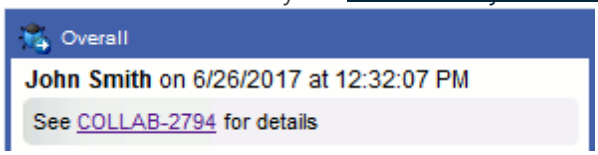



*JIRA ticket with link to review*

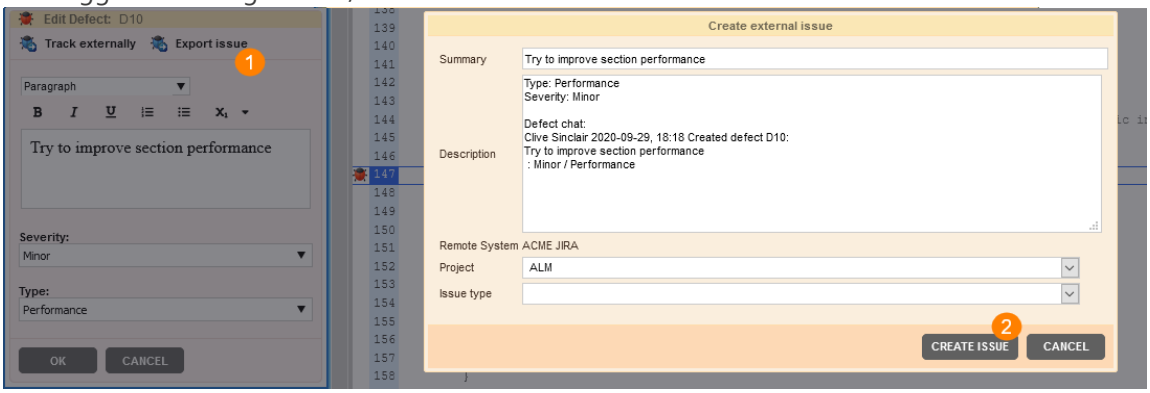



*Team Foundation Server work item with link to review*

- **Converts ticket identifier into hyperlink.** (JIRA Only.) This is similar to creating automatic links manually, however requires less effort, since appropriate links will be configured automatically on the basis of the specified issue-tracking server. This feature is enabled by default and controlled by the Automatically Add Remote System Links 246 template setting.

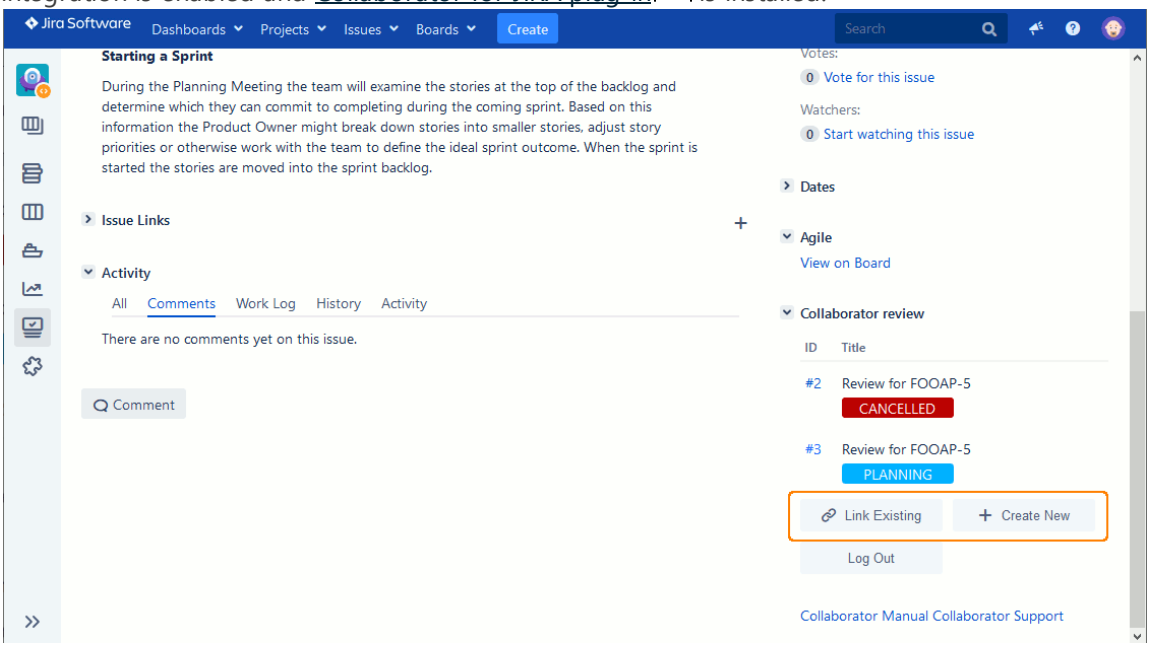



- **Allows to create tickets for external defects.** (JIRA Only.) You can create tickets in your issue-tracker directly from the Collaborator reviews. This feature is disabled by default. To enable defect export, go to the Review Templates 246 screen, edit the appropriate template, change the "Automatically Create New Work Item for External Defects 246" setting for that specific system field to "Enabled" and select which project will be pre-selected in the dialog that suggests creating a ticket/work item.




*Creating JIRA ticket from review*

- **Allows to create new reviews or link existing reviews form issue-tracker.** (JIRA Only.) You can create new Collaborator reviews or link existing reviews directly form the issue-tracker ticket. This feature is enabled, when the respective issue-tracker configuration[896] is created, integration is enabled and Collaborator for JIRA plug-in[892] is installed.




*Creating reviews from JIRA ticket*

## Technical Details

- Item statuses in the Remote System Links are updated upon webhook events (if a remote system supports webhooks), or periodically (default interval is every 2 minutes and can be adjusted via VM option[1232]).
- The "Project List" field of JIRA configurations can contain up to 2000 characters, whereas for JIRA (Legacy) configurations it can contain only 255 characters.
- Connections established by JIRA integrations do respect proxy settings[89] if they are specified for Collaborator server.
- Team Foundation Server work items do not have unique identifier except for their ordinal number. Because of this, Collaborator cannot determine work item identifiers within review's text fields (title, custom fields, comments and others) and, consequently, cannot automatically convert them into hyperlinks.
- Currently, creating tickets for external defects is only available for JIRA issue-tracking systems.

- When creating tickets for external defects on JIRA servers, the dialog may display additional fields depending on server's custom fields configuration. Currently, integration is able to retrieve and display standard custom fields. Advanced custom fields are not supported yet. If advanced custom fields are mandatory for creating new tickets, Collaborator will fail to create it.

### Related Topics

#### General Approach to Issue-Tracking Integrations 906

This topic describes general approach to integration with any issue-tracking system. Integration can be done in several ways depending on your needs.

## 8.2 Collaborator for JIRA Plug-in

The Collaborator for JIRA plug-in automatically links Collaborator reviews to JIRA tickets. Additionally, the plug-in allows users to manually link or create new Collaborator reviews from within JIRA. The plug-in works with both JIRA Cloud (SaaS) and JIRA Server (on-prem) instances of JIRA.

### Requirements

In order to use the plug-in the following preparations should be done:

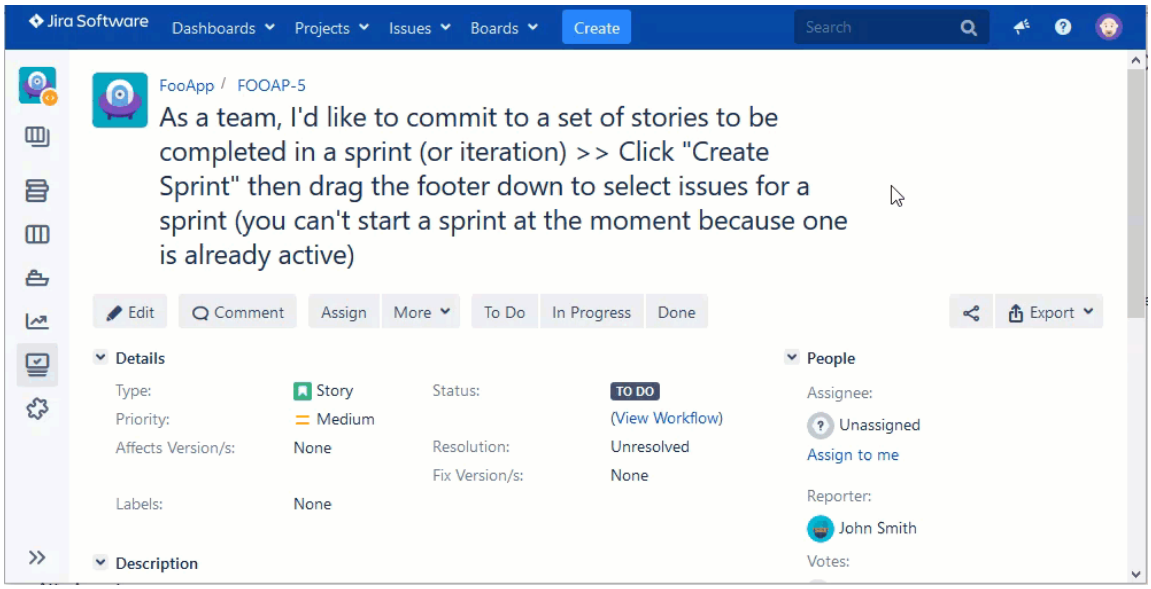
- **JIRA integration be running on the Collaborator side.** Your Collaborator administrator should create and configure 897 integration between Collaborator server and your JIRA instance.
- **The plug-in be installed and configured on the JIRA side.** Your JIRA administrator should install the plug-in 899 on your JIRA instance and configure which JIRA projects 901 should display Collaborator information.

### Supported environment

- JIRA server 5.0 and higher (SaaS or on-prem, including Atlassian Data Center).
- Collaborator server 12.x and higher.

### Working With

Once installed and configured, the plug-in adds a Collaborator sidebar block to a JIRA ticket screen.

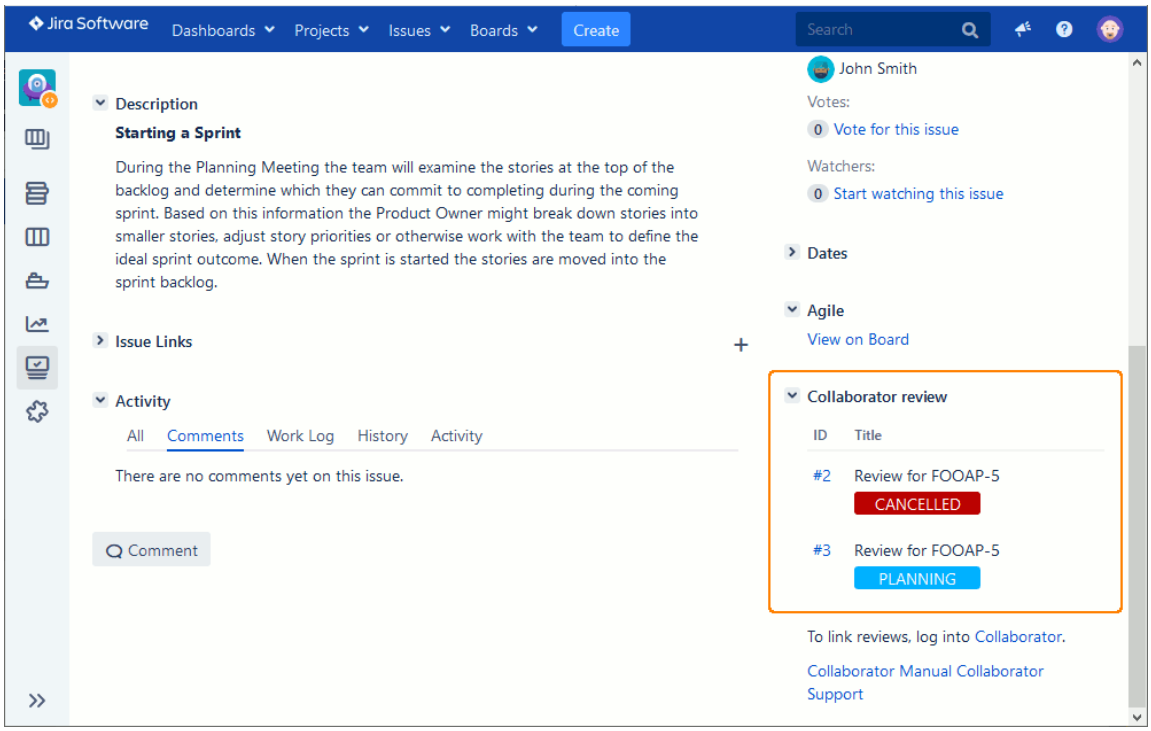



*Collaborator reviews in JIRA tickets*

In this block, you can:

- **See linked Collaborator reviews and their statuses.**

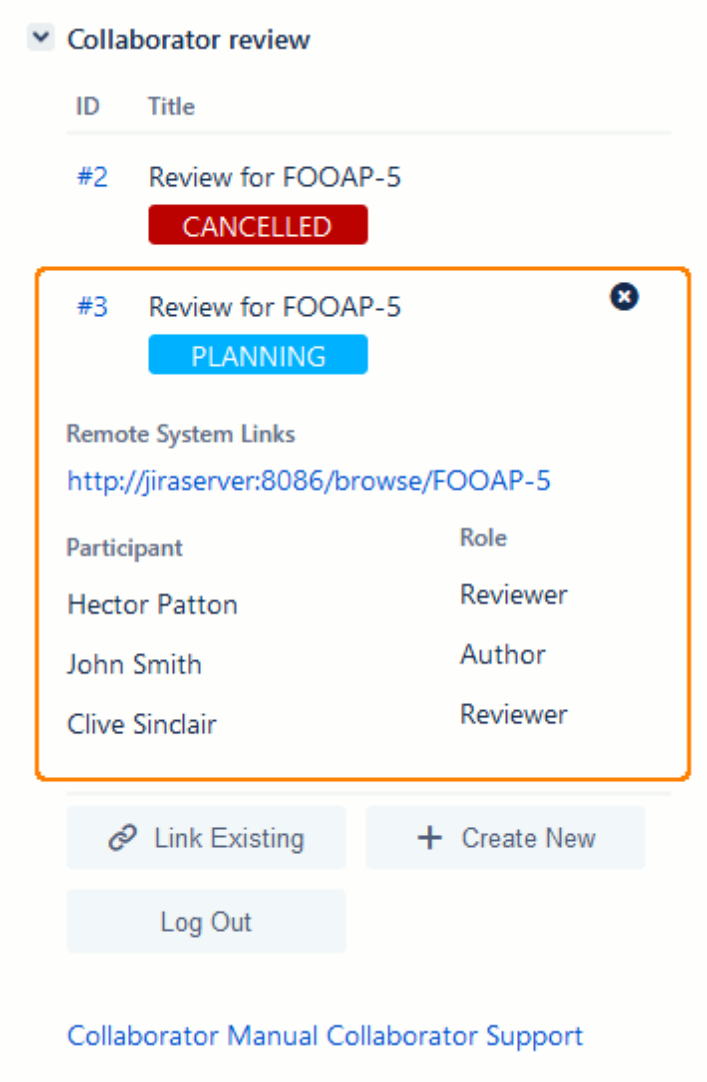
Just scroll to the Collaborator sidebar block and expand it.




*JIRA ticket with link to review*

- **See review details.**

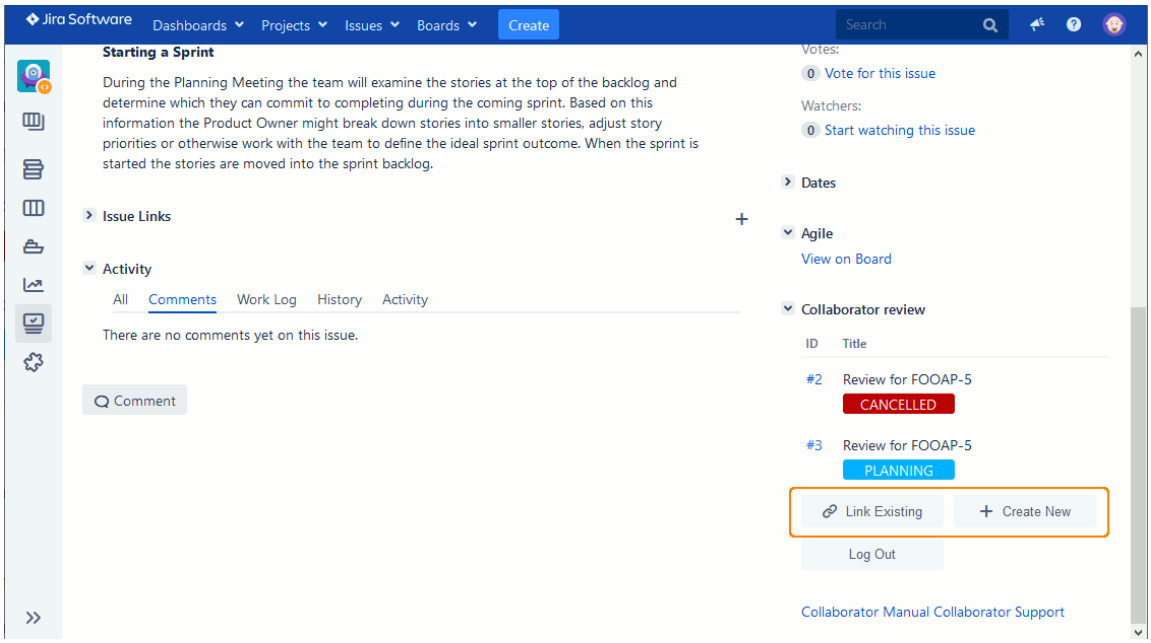
Click on the desired review in the list of reviews.




*Participants of linked review*

- **Create new or link existing reviews.**

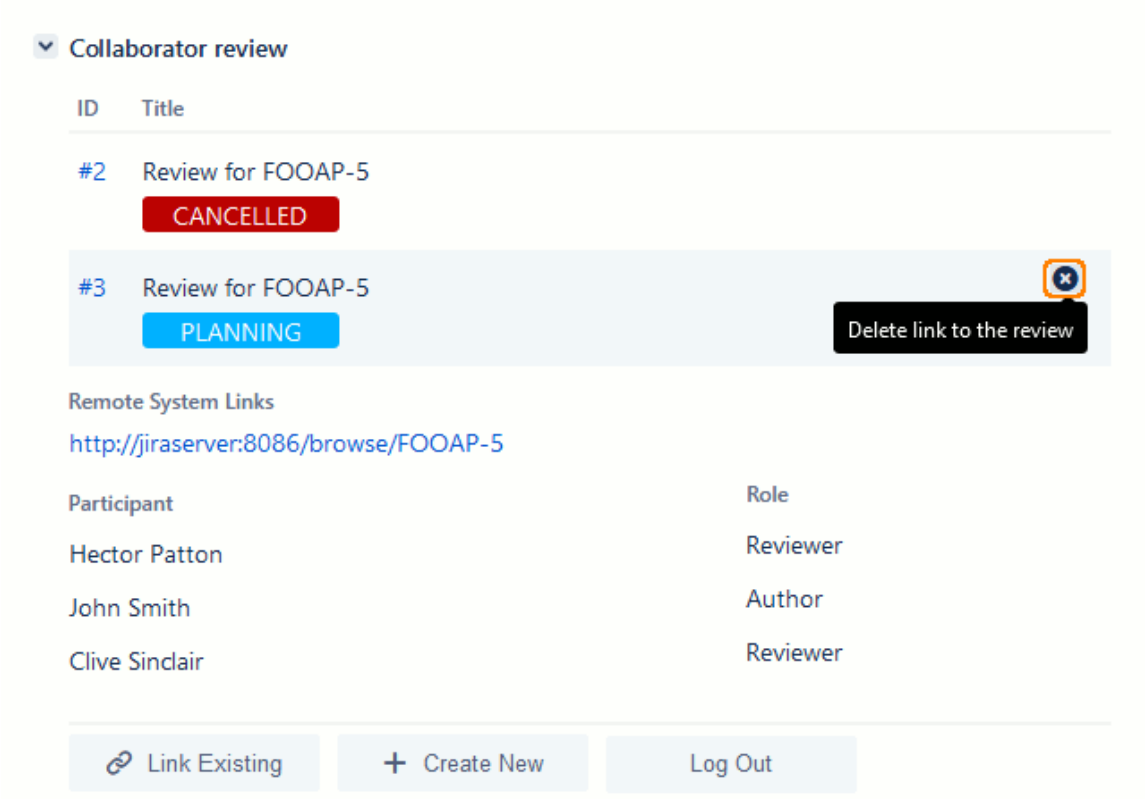
Click **Create New** or **Link Existing** buttons, respectively.




*Creating reviews from JIRA ticket*

- **Remove links to reviews.**

Click on the desired review in the list and then click on the X sign.




*Delinking reviews*

## Technical Details

- By default, detailed review information is displayed respecting the Restrict Access to Review 188 setting. To make this information be available to any JIRA user (even to those having no Collaborator accounts), your JIRA administrators could enable the Show reviews to everyone 901 project setting.

## Version History

1.0.1 - Fixed compatibility with JIRA Server 7.x

1.0.0 - Initial release of the Collaborator for JIRA Plug-in.

# 8.3 Configuring Issue-Tracking Integrations

This section describes how to setup integrations with remote issue-tracking systems supported by Collaborator.

- Configuring JIRA Integration 897
- Configuring TFS Work Item Integration 903

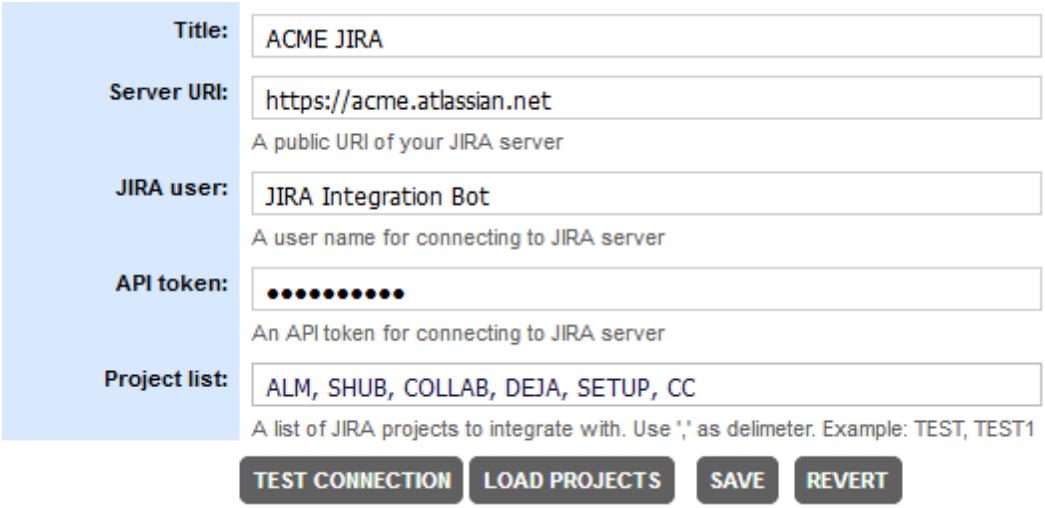
For other issue-tracking systems, your Collaborator administration should establish integration as described in General Approach to Issue-Tracking Integrations 906.

## 8.3.1 Configuring JIRA Integration

This topic describes how to setup Collaborator integration with JIRA issue-tracking system. To learn general principles of how any issue-tracking integration operates, see Issue-Tracking Integrations: Overview 887.

### Configure Collaborator server

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.
2. In Collaborator, go to **Admin > Remote System Integrations > Issue-Tracking Services**
3. In the **New Remote System Configuration** section, select *JIRA* and click **Create.** This will display the configuration settings.




Specify the setting values:

| Setting | Description |
|---|---|
| **Title** | The configuration name as it will be displayed in Collaborator's user interface. |
| **Server URI** | The JIRA server's URL and port.<br><br>For instance: *https://jira.acme.com* |
| **JIRA user** | The user name that Collaborator will use for connecting to JIRA. |

| | |
|---|---|
| | The specified user must have read-write permissions to JIRA projects. Administrator permissions are NOT required. You may specify a regular user, or create a dedicated user for this integration. |
| **API token** | The API token or password to use for connecting to JIRA.<br><br>For **JIRA Cloud (SaaS)** instances you need to specify API token. To generate token, log in to https://id.atlassian.com/manage/api-tokens, click **Create API token**, specify token caption and click **Create**.<br><br>See Atlassian documentation for detailed information about API tokens.<br><br>For **JIRA Server (on-prem)** instances you need to specify password. |

After specifying these values, you can click **TEST CONNECTION** to verify if you entered data correctly.

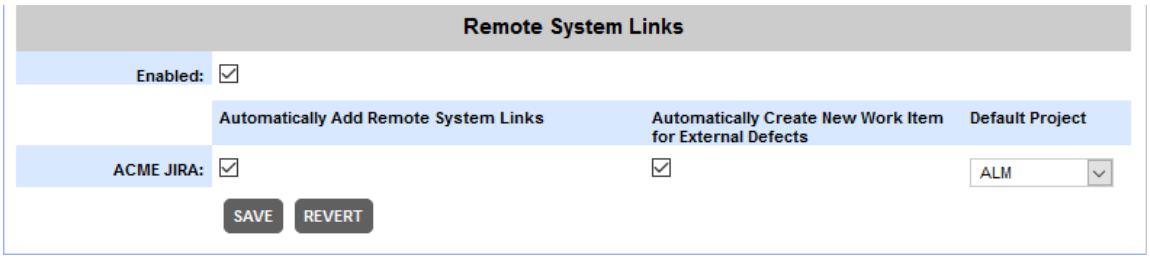
| Setting | Description |
|---|---|
| **Project list** | A string containing the keys of JIRA projects to integrate with.<br><br>**Tip:** Click **LOAD PROJECTS** to read project keys from the JIRA instance specified by the *Server URI* setting.<br><br>Project keys are case-insensitive. The "Project List" field can contain up to 2000 characters. |

4. After you specified the values, click **Save**. This will create and enable JIRA integration.

5. Switch to the **Admin > Remote System Integrations** page. The **Integration Status** section allows you to quickly enable or disable integrations with remote systems.

6. Locate the **Enable JIRA Integration** setting and verify that it is set to **Yes**.

7. Once the JIRA configuration is enabled, automatic linking for the specified remote systems will be enabled in all active templates. However, Collaborator administrators may enable or disable linking per each template through its Automatically Add Remote System Links 246 setting. Besides that, administrators can enable the ability to export review defects to the selected issue-tracking system. To enable defect export, go to the Review Templates 246 screen, edit the appropriate template, change the "Automatically Create New Work Item for External Defects 246 " setting for that specific system field to "Enabled" and select which project will be pre-selected in the dialog that suggests creating a ticket/work item.

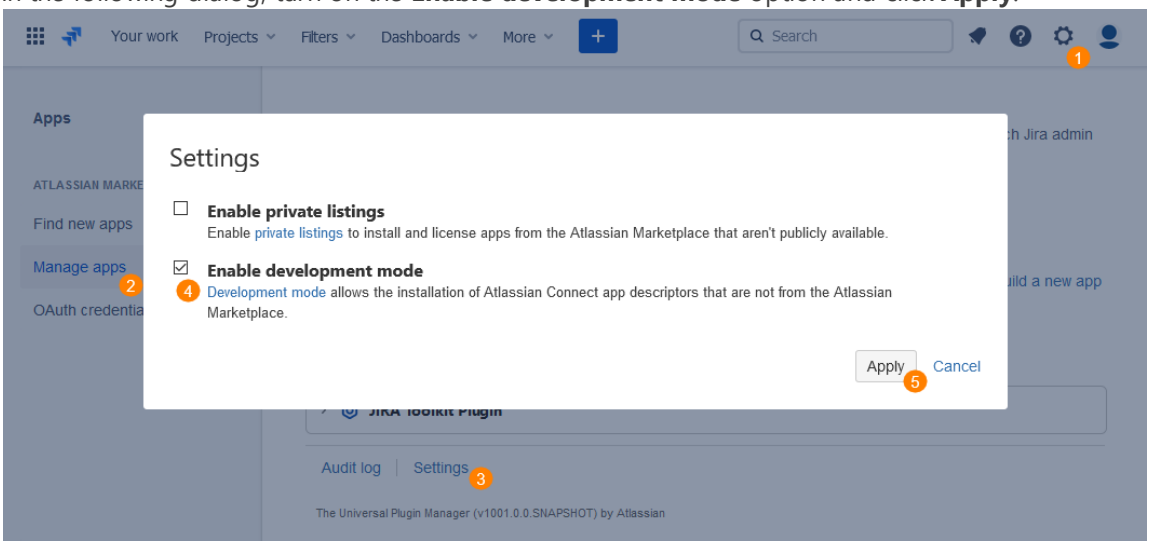



## Configure JIRA server

To display review information on the JIRA side you need to install Collaborator plug-in[892] on your JIRA instance and configure your projects to use that plug-in. The plug-in works with both JIRA Cloud (SaaS) and JIRA Server (on-prem) instances of JIRA.

### Install plug-in on JIRA Cloud (SaaS)

1. Log into your JIRA instance as an administrator.
2. Click the admin drop-down and choose **Apps**.
3. Go to the **Manage apps** section and click **Settings**.
4. In the following dialog, turn on the **Enable development mode** option and click **Apply**.

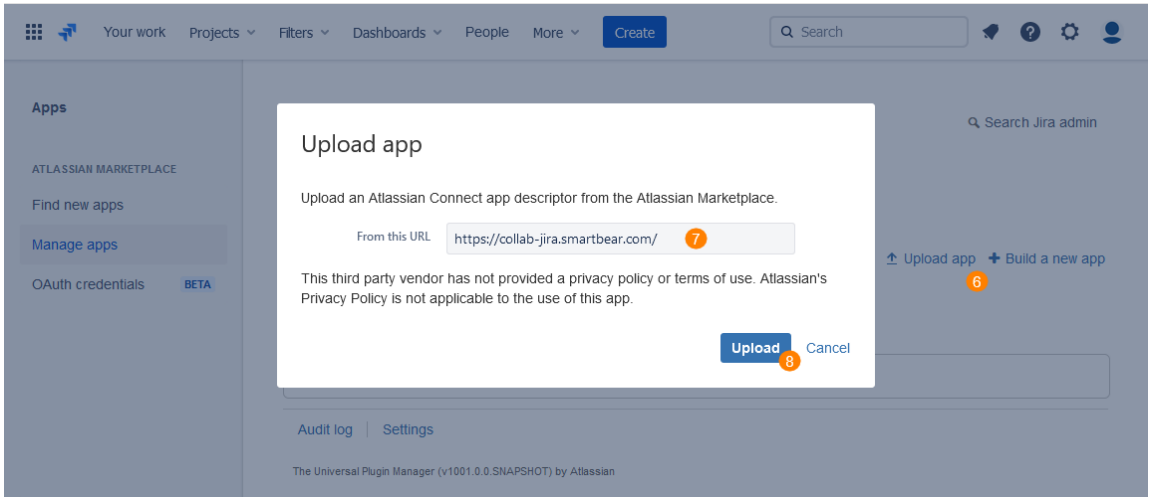



*Enabling development mode*

This will allow to upload and install custom apps.

5. Click **Upload app** button.
6. In the following dialog, specify https://collab-jira.smartbear.com in the **From this URL** field.

7. Click **Upload** to upload and install the plug-in.

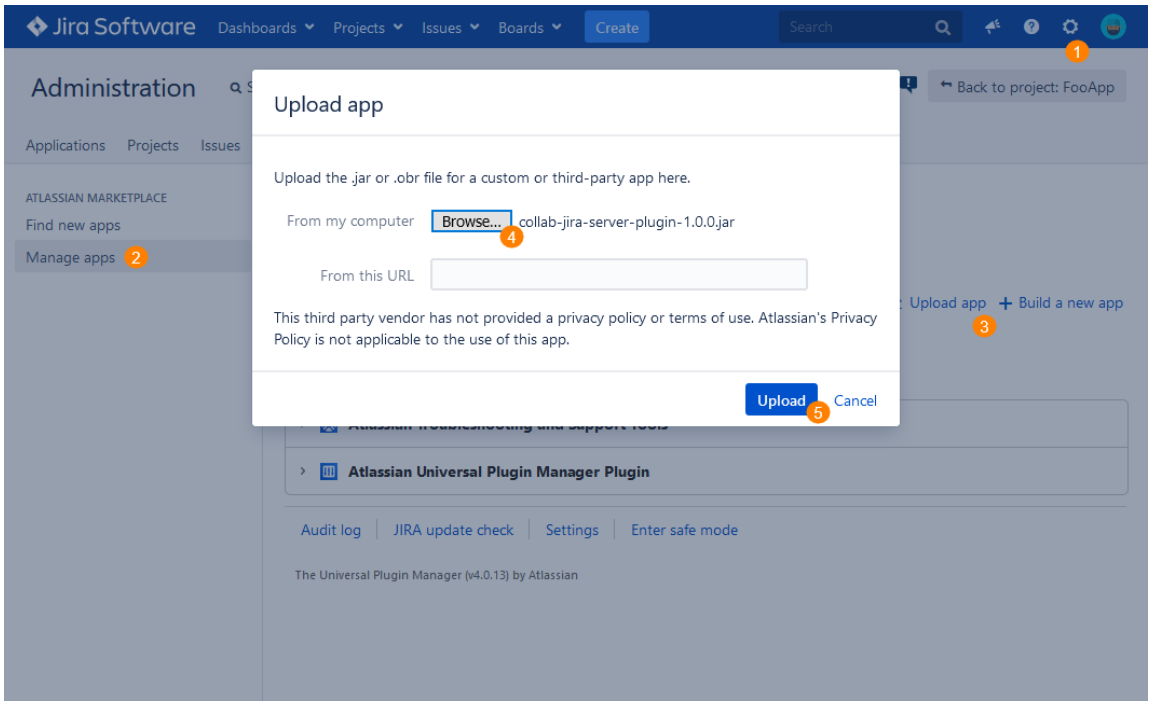



*Installing plug-in on JIRA cloud*

Once installed, the plug-in on will appear on the **Manage apps** page of the JIRA administration console.

## Install plug-in on JIRA Server (on-prem)

1. Visit the Collaborator JIRA Plug-in download page and get the most recent version of server plug-in.
2. Log into your JIRA instance as an administrator.
3. Click the admin drop-down and choose **Manage apps**.
4. Go to the **Manage apps** section and click **Upload app**.
5. In the following dialog, click **Browse** and specify path to the downloaded plug-in .jar file.
6. Click **Upload** to upload and install the plug-in.




Once installed, the plug-in on will appear on the **Manage apps** page of the JIRA administration console.

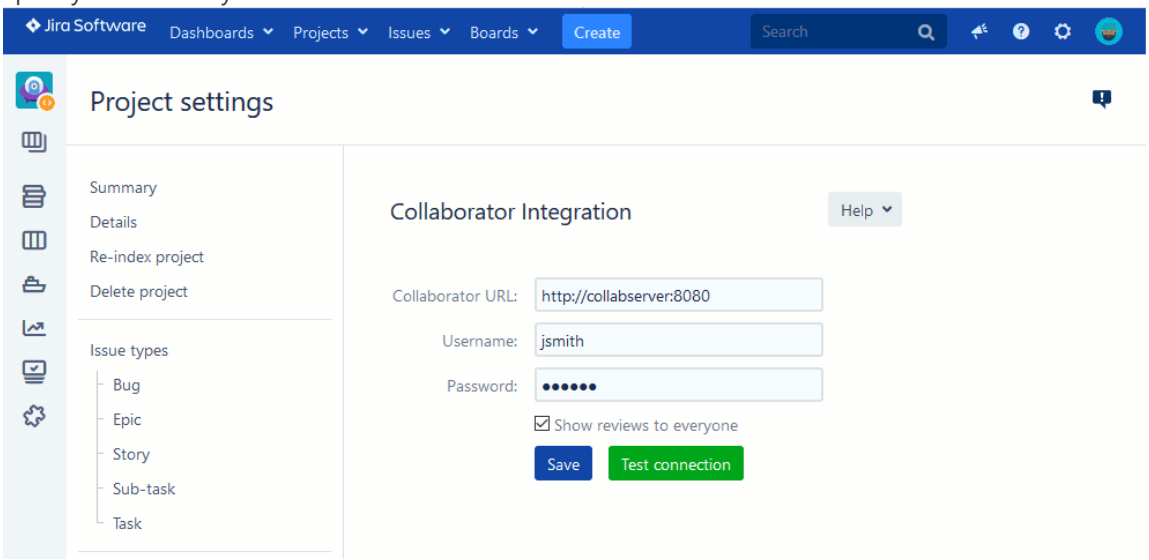
## Configure JIRA Projects

Now you need to configure your JIRA projects to display information about Collaborator reviews using the new plug-in.

For each project that should integrate with Collaborator, perform the following steps:

1. Log into your JIRA instance as an administrator.
2. Select the desired project in the **Projects** drop-down list.
3. Click **Project settings** and scroll to the **Collaborator Integration** section.

4. Specify the URL of your Collaborator server and credentials to use for connection:




5. Enable or disable the **Show reviews to everyone** option to control whether review information could be available to users that are not participants of that review:

- If disabled (default), the Collaborator panel will contain links and statuses to all linked reviews, however detailed review information would be displayed respecting the Restrict Access to Review 188 setting.
- If enabled, the Collaborator panel will contain links and statuses to all linked reviews. This list will be visible to any JIRA user and would contain the following information: link to the review on Collaborator server, current status, list of participants and their roles.

6. Click **Save**.

## Establish trusted connection

When your on-premises JIRA Server use SSL connection, its certificates may not be trusted by Collaborator server. In this case you will need to import that certificate as trusted.

To establish trust, you need to import the public key of your on-premises server as a trusted certificate to Collaborator keystore file:

1. Get the certificate file from your on-premises server or network administrator.

2. Locate the keystore file which you have generated while configuring Collaborator HTTPS connection 112.

Default location is `<Collaborator Server>/tomcat/conf/collab.ks`, yet that could be changed while generating keystore.

3. Use Java's keytool utility to import the certificate in Collaborator keystore file. You can find keytool in the `$JAVA_HOME/bin` directory:

```
$JAVA_HOME/bin/keytool -import -alias on-prem-server -keystore
<collab-keystore-path> -trustcacerts -file <certificate-file-name>
```

For more information on command-line arguments of the keytool utility, see keytool documentation.

4. Most likely you will be prompted to confirm the validity of the certificate. It is imperative for the security of the overall system that you verify the key matches the trusted material. Before accepting the certificate, you should contact the administrator that sent you the certificates and verify that the certificate fingerprints that you see match the certificate fingerprints that they intended to send you.

5. The final step is to configure Collaborator to use the newly created keystore. Open the `<Collaborator Server>/ccollab-server.vmoptions` file in a text editor, and add the following lines to it:

```
-Djavax.net.ssl.trustStore=<collab-keystore-path>
-Djavax.net.ssl.trustStorePassword=<collab-keystore-password>
```

6. Restart the Collaborator server.

Now the integration between Collaborator and JIRA is configured and running.

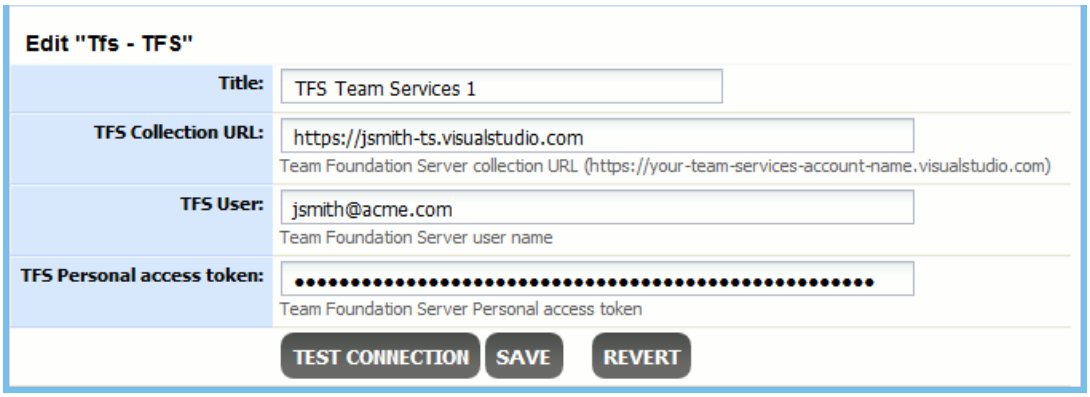
### 8.3.2 Configuring TFS Work Item Integration

This topic describes how to setup Collaborator integration with TFS work item system. To learn general principles of how any issue-tracking integration operates, see Issue-Tracking Integrations: Overview 887.

1. Open the Collaborator login page in a browser and log in to Collaborator as an administrator.

2. In Collaborator, go to **Admin > Remote System Integrations > Issue-Tracking Services**

3. In the **New Remote System Configuration** section, select *TFS* and click **Create.** This will display the configuration settings.




4. Specify the setting values:

Remote system configuration settings depend on whether you use a self-hosted version or a SaaS version of Team Foundation Server known as Visual Studio Team Services. If you use the latter, then you might need to set-up alternate authentication credentials at first and then use those credentials in your TFS configuration. See Client Configuration for Visual Studio Team Services 741 for configuration instructions.

| Setting | Description |
|---|---|
| **Title** | The configuration name as it will be displayed in Collaborator's user interface. |
| **TFS Collection URL** | For self-hosted version of Team Foundation Server, specify the URL of Team Foundation Project Collection to work with.<br><br>For instance: *http://tfs.acme.com/my-collection*<br><br>For SaaS version of Team Foundation Server, specify the URL of your Visual Studio Team Services account (without project or collection names).<br><br>For instance: *http://jsmith-ts.visualstudio.com* |
| **TFS User** | The name of Team Foundation user. For SaaS version of Team Foundation Server, specify either user name or alternate primary user name.<br><br>The specified user must have read-write permissions to TFS projects. Administrator permissions are NOT required. You may specify a regular user, or create a dedicated user for this integration. |
| **TFS Personal access token** | The personal access token of Team Foundation user. For SaaS version of Team Foundation Server, specify either personal access token or alternate credentials password. |

After specifying these values, you can click **TEST CONNECTION** to verify if you entered data correctly.

6. After you specified the values, click **Save**. This will create and enable TFS integration.

7. Switch to the **Admin > Remote System Integrations** page. The **Integration Status** section allows you to quickly enable or disable integrations with remote systems.

8. Locate the **Enable TFS Integration** setting and verify that it is set to **Yes**.

## Establish trusted connection

When your on-premises server use SSL connection, its certificates may not be trusted by Collaborator server. In this case you will need to import that certificate as trusted.

To establish trust, you need to import the public key of your on-premises server as a trusted certificate to Collaborator keystore file:

1. Get the certificate file from your on-premises server or network administrator.

2. Locate the keystore file which you have generated while configuring Collaborator HTTPS connection[112].

Default location is `<Collaborator Server>/tomcat/conf/collab.ks`, yet that could be changed while generating keystore.

3. Use Java's keytool utility to import the certificate in Collaborator keystore file. You can find keytool in the `$JAVA_HOME/bin` directory:

```
$JAVA_HOME/bin/keytool -import -alias on-prem-server -keystore
<collab-keystore-path> -trustcacerts -file <certificate-file-name>
```

For more information on command-line arguments of the keytool utility, see keytool documentation.

4. Most likely you will be prompted to confirm the validity of the certificate. It is imperative for the security of the overall system that you verify the key matches the trusted material. Before accepting the certificate, you should contact the administrator that sent you the certificates and verify that the certificate fingerprints that you see match the certificate fingerprints that they intended to send you.

5. The final step is to configure Collaborator to use the newly created keystore. Open the `<Collaborator Server>/ccollab-server.vmoptions` file in a text editor, and add the following lines to it:

```
-Djavax.net.ssl.trustStore=<collab-keystore-path>
-Djavax.net.ssl.trustStorePassword=<collab-keystore-password>
```

6. Restart the Collaborator server.

Now the integration between Collaborator and TFS work items is configured and running.

# 8.4 General Approach to Issue-Tracking Integrations

Collaborator is flexible enough to integrate with any external issue-tracking system: TeamTrack, Bugzilla, FogBugz and so forth. Integration can be done in several ways depending on the type of integration you require.

## Hyperlinked Issues

Collaborator can identify external issue references in titles, custom fields, comments, defects, and more, and automatically hyperlink them to the right web page in your external issue tracker.

For example, if your company uses the word "`Case`" followed by a number to identify an issue in an issue-tracking system. You can configure Collaborator to search for text in the (regular expression) form of "`\bcase\s*(\d+)`" and automatically hyperlink the number part (in the grouping symbols) to your issue-tracker.

To learn how to setup automatic hyperlinks, see this section 286 of administrator guide.

## Referring to issues in the Review Overview

It is common to want to associate one or more issues with each review. The easiest way to do this is to create a custom review field 256 and set the regular expression validator to ensure that only properly-formatted issues are entered in.

You should also set up the issue-hyperlink feature described above so that these fields are interactive.

## Externalizing: Moving a defect from Collaborator to an external issue tracker

Sometimes you find a defect during review that you do not want to fix just now. In this case you want to move the defect from Collaborator into an external issue tracker. You also want to record this state and audit trail in the review.

The externalized defects 442 feature allows you to do just that. See that section of the manual for details.

## It is unusual to integrate beyond this point

This is the extent to which most of our customers go with integration. Most people do not want review defects mirrored into an external issue tracker because these defects were never "delivered".

For example, if you had done the peer review side-by-side with someone, you would not enter in defects then -- you would just fix them!

Usually the QA department runs off the issue tracker for verification. How can QA verify something that was not necessarily externally broken (for example, some function was not checking input values carefully) or something just in the file (for example, some method was not documented properly).

### Mirroring defects from Collaborator into an external issue tracking system

You can use the server-side trigger 206 that runs every time a defect is added, edited, marked fixed, or deleted. Your trigger would locate the corresponding external issue record and update it. This integration takes a bit of work. We do not have it pre-built because everyone's external issue-tracking system is different.

For an example see this scripting tutorial 928.

# 9 External Integrations

Integrating Collaborator into other systems, scripts, and processes is easy. A wide variety of integration mechanisms is available.

## Reporting-writing tools 908

Collaborator comes with a few built-in reports 467, but almost everyone wants to make reports of their own.

You can use any external reporting system including Excel, Access, Crystal Reports, and Business Objects.

## Scripting 926

Scripting is the way to implement custom behaviors in Collaborator. You can run scripts on your user's machines, or from the server machine, possible invoked with a trigger. Almost any action you can do in the Web Client can be done from a script.

## Web Services 943

Web services provide external programmatic interfaces to Collaborator server. You can use web services from different client applications to run various commands on a server. Web service runs on a server and waits for special kind of requests. Upon receiving a request the server processes it and sends the results in response. You can use JSON  API services from various applications to interact with the Collaborator server.

# 9.1 Creating Custom Reports

Collaborator comes with a number of pre-built and customizable reports[467]. However, if you would like to automate a report for integration with external tools or applications, or if you would like to create your own reports, you can access the underlying database directly.

(Note: SmartBear does not support querying Hypersonic[60] databases. You will need to use one of the other databases[59] supported by Collaborator.)

## Creating Custom SQL Reports

In many cases, you may want report output slightly different than one of the existing customizable reports. In these cases, it may be simplest to begin with a report that outputs most of the data that you want, then modify its database query as needed.

1. Open a customizable report[467] that best matches the type of report you would like to create.
2. Choose the columns you wish to display in the report, and the criteria by which you would like to filter the results.
3. Click the "Save" button.
4. Verify that the first few results contain the data you expect.
5. Click the "SQL" link at the top-right of the Results pane. This will download a SQL query appropriate for your database which you can use as a starting point for your custom report.
6. Reference the Collaborator database schema[908] and documentation for your database's SQL dialect (MySQL, Oracle, or Microsoft SQL) to modify the query as you see fit.
7. Execute the query directly against the database using whatever SQL tool you like. Some tools that can run SQL queries include: Excel, Access, Crystal Reports, and Business Objects.

## Custom XSL Reports

You can alternatively use the Command-Line Client to retrieve XML data regarding a review. For samples of this, see: Example XPath and XSL[923].

### 9.1.1 Database Schema

This section covers the Collaborator database schema and some special features of the database we created specifically to support external custom reporting applications.

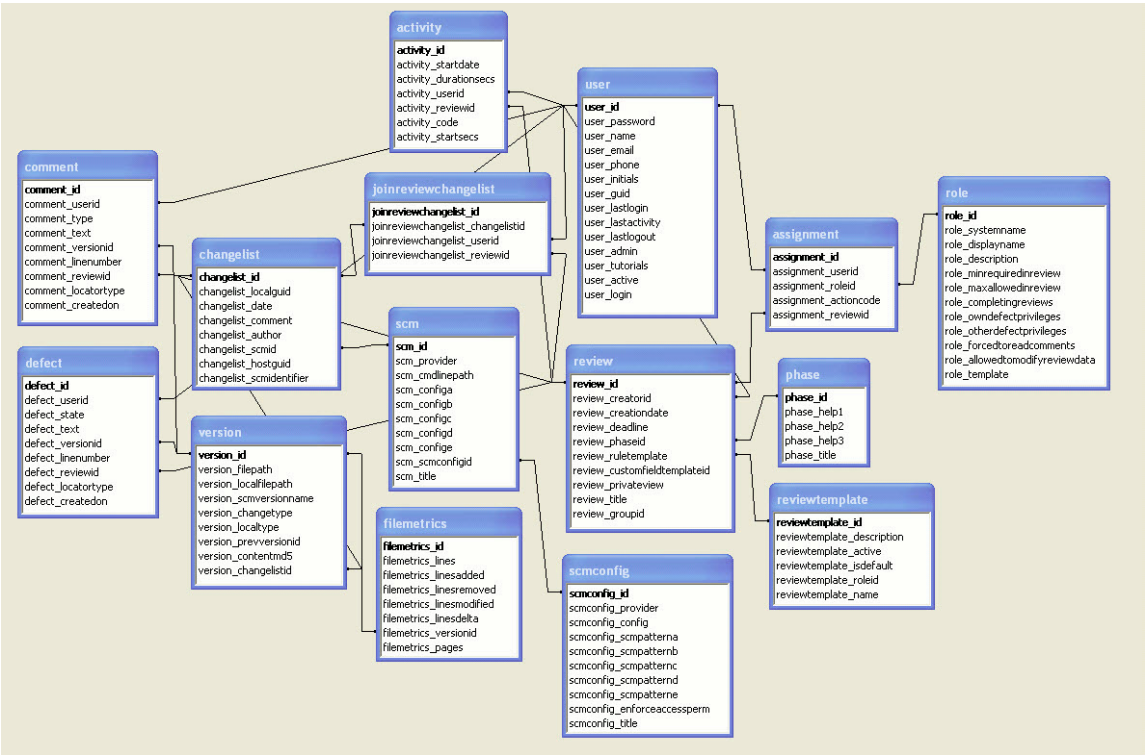
## The Database is Read-Only!

**Never change data in the database directly.** Although we support read-only access to the database for reporting and automation, if you make changes to data in the database yourself **you could irreparably destroy the integrity of the database.**

## Database Schema

The diagram below shows *only the major tables* in the Collaborator database and which files should be linked with which other fields when creating queries.

There are more tables which are either (a) purposefully undocumented or (b) described below but which do not have relationships to other tables. The diagram does not contain those tables in the interest of simplicity:




*Database Schema*

**Warning:** SmartBear reserves the right to change any of the table definitions whenever there is a minor point-release of the product.

Note: The table/view names above are approximations. In some cases, the names may be altered slightly (for example, vowels are removed).

We are committed to backwards-compatibility with the reporting views[914], and with our own client software[506].

Here is a brief description of each table:

| | |
|---|---|
| activity | Activity log of users' actions during review. This is used to compute metrics such as inspection rate.<br><br>Each row represents a slice of time where the user was active. There might be many slices for a given user in a single review. Each slice includes a duration (in seconds) and a "start time" that is encoded both as a database date/time (`activity_startdate`) and as a number of seconds since 1970-01-01 00:00:00 GMT (`activity_startsecs`).<br><br>The activity action code (`activity_code`) tells whether the user was acting in the capacity of an author (A), reviewer (R), or was doing rework (F) as opposed to actually reviewing. Also there is a code for when someone views the review but is not a participant (P).<br><br>It almost always best to use the reporting views[914] to access activity data so you do not get your query wrong. The technique and motivation behind this system is described in our metrics section[972]. |
| assignment | List of user <--> review combinations. For each user who was "assigned" to a review, notes the role associated with that user.<br><br>The `assignment_actioncode` field is used internally to understand what general status that user has in the review currently. You should not depend on this field as we might change it in the future. Refer to the `assignment_state` view for names and descriptions for these codes. |
| changelist | Every time a set of files is uploaded to the server a "changelist" record is created. Most changelists will be associated with a review through `joinreviewchangelist`, but some may not be for various reasons. Changelists are linked to `version` where the actual file content is represented. |

| | |
|---|---|
| | If the changelist is of files from a local hard drive, the "local GUID" field will contain a globally-unique made-up identifier for that upload. If the changelist was derived from something already checked into version control (for example, a Perforce or Subversion changelist), then this field will be blank.<br><br>The date, author, and check-in comment are all listed if known. To the extent possible this will match usernames with version control.<br><br>Also linked is the SCM table ID. This ties the changelist to a specific version control server. Changelists from different servers might match in other details but are actually unrelated.<br><br>If the changelist happened to have an associated identifier in a version control system, that is also recorded. Otherwise that field is blank. |
| comment | Represents a comment made by a user in some conversation. This includes not only actual chat but indirect events such as "marked read" and "created defect" and "new file uploaded".<br><br>Also included is the file (version) and line number the comment is associated with, however both of those fields are optional. They are linked to user ID and review ID as well. |
| defect | Represents a defect made by a user in some conversation.<br><br>Also included is the file (version) and line number the defect is associated with, however both of those fields are optional. They are linked to user ID and review ID as well.<br><br>A state field indicates whether the defect is still open or fixed. We will be adding more state to this field in the future. |
| filemetrics | Holds basic change metrics for file versions. Each filemetrics record is tied to one version record.<br><br>You should only depend on the values of this table for versions directly associated with changelists. The other metrics are often incomplete. There are technical reasons for this; we will not be changing this behavior. |
| groupdescription | Represents a group 226. Some of the groups are defined in the web UI and some are built-in internal groups automatically defined by Collaborator |
| groupusers | Joins the groupdescription table with the user table to represent the users that are direct members of a group. A user can be a member of zero, one, or many groups. |

| | |
|---|---|
| groupgroups | Joins the groupdescription table with itself to represent the groups that are direct members of a group. A group can be a member of zero, one, or many groups. |
| groupancestry | *The behavior of this table is intentionally undocumented.* |
| joinreviewchangelist | Joins reviews and changelists. A changelist can be associated with zero, one, or many reviews, and a review can be associated with zero, one, or many changelists. |
| metadatadescription metadatavaluecharacter metadatavaluedate metadatavalueinteger metadatavaluestring metadatavaluestringbig | All of these tables have to do with "meta-data" which means any data where the data schema itself is dynamic. Most notably, all review and defect custom fields[256] are a kind of meta-data.<br><br>You should not use the meta-data tables directly. Their relationships are very complex and we change how they work regularly as we add more features.<br><br>Instead, access meta-data through the reporting views described below[914]. This contains all the information you need for custom fields and formats it nicely as an added bonus. |
| metadataselectitem | Information for all the drop-down items in any custom field.<br><br>Each item is matched to a particular custom field. The "title" is the text displayed to end users. A "sequence" number defines the order of the elements (the IDs are not an order). Items can also be individually enabled or disabled. |
| notification | Holds the history of notification messages that have been sent out to clients. Clients might choose (or not) to get notifications by email, RSS feed, and so on.<br><br>This table is periodically cleaned out by the server. There will always be some backlog of events for each user (for example, for use in creating the RSS feed for a user), but you cannot depend on any particular number of events to be saved. |
| reportcategory reportfilter reporttemplate | *Internal server use. Do not depend on this table.* |
| review | Holds one record for each review in the system. |

| | |
|---|---|
| | The "creator" is the user who created the review, or the system administrator if the review was created automatically.<br><br>Use the review custom field view[914] to access review custom field data. |
| reviewtemplate | *Internal server use. Do not depend on this table.* |
| role | Represents all of the roles from all role-sets. Each role has a "standard" name that never changes and the custom name that was set by the user. |
| scm | Contains one record for each SCM server that has ever been reported by a client. It is OK if there are duplicate records for a given SCM system. This separates changelists from different systems. This table will probably change in the future. |
| user | One record for each user who can log into the system. User ID 1 is the special system administrator.<br><br>Key user information and preferences are stored here. Additional user preference information is stored as meta-data and is accessible from the special `userprefs` view.<br><br>The `user_initials` field is deprecated and should be ignored.<br><br>Passwords are stored in hashed form so that a casual observer cannot deduce a password. If you need to reset a password, set this field to:<br>`d41d8cd98f00b204e9800998ecf8427e` |
| version | One record for every version of every file that has ever been uploaded to the server. Join with `changelist` to see the group they were uploaded with.<br><br>Each version includes the full file path to the original document. If this file was retrieved from version control, this will be the version control server path, not the path on the user's local hard drive.<br><br>The version name is the version control-specific name of the version of the file. This is typically a number or set of numbers.<br><br>The version change-type indicates whether this represents a file addition, deletion, modification, and so on. Sometimes the system does not know. You should use this as a guide but not depend on it because there are exceptions to the "type" rules and we add new types periodically. The current values are:<br>`? - Unknown`<br>`A - Added`<br>`B - Branched`<br>`D - Deleted`<br>`I - Integrated` |

| | |
|---|---|
| | M - Modified<br>R - Reverted<br>U - Uploaded<br><br>Sometimes the version will have a "previous" version. This typically means the version that came before it in version control. This is used internally and is tricky; there are lots of exceptions and we can change exactly what this means.<br><br>The content MD5 is the MD5 sum of the raw content of the file. This can be used to link a version with the on-disk file content stored in the content cache. It can also be used as a check to see whether two versions are identical. |

## Reporting Views

For databases that support the concept of a "View," Collaborator creates a set of Views specifically for the purpose of external report-writers. You should use these Views whenever possible; we will make sure that the definitions of these Views remain the same even if we change the database schema in future versions.

You can also use these Views as a guide for how to create other custom queries.

Here are the special reporting Views:

| | |
|---|---|
| assignment_view | Contains columns from the assignment table which will be maintained in the event of a future schema change. |
| assignment_state | Contains names and descriptions for the codes used in the actioncode column of the assignment table. |
| defects_by_path | One row per review reporting file path, lines of code reviewed, and number of defects. |
| defectcustom | These are the custom fields you have defined for defects, one row for each defect. If you change the custom field definition, the layout of this table will change as well (automatically, and immediately). Warning: Because the exact custom field titles are used for column names, if you change the title of a custom field it will change the definition of this view.<br><br>With review workflows 246, each review might have a different subset of custom fields. In this case fields are NULL when they are not applicable. |

| | |
|---|---|
| defectcustom_compat | Same as defectcustom, but column names are in the form custom_id_N where "N" is the custom field ID as displayed in the custom fields admin page[256]. The columns are ordered exactly the same as defectcustom, so you can also tell which is which by comparing the two views. |
| defect_custom_dropdowns | This table contains a list of the possible values as defined in drop-down and multi-select Defect Custom Fields. |
| defect_state | Contains "defect state" codes and names. Join on this table if you would like to display more user-friendly defect state names. |
| defect_view | Contains columns from the defect table which will be maintained in the event of a future schema change. |
| participant_singleline_values | Shows Participant Custom Field values that users have selected in reviews. This table only shows values for "Single-Line Text" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value. |
| participant_multiline_values | Shows Participant Custom Field values that users have selected in reviews. This table only shows values for "Multi-Line Text" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value. |
| participant_select_values | Shows Participant Custom Field values that users have selected in reviews. This table shows values for "Drop-Down" and "Multi-Select" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value. In the case of Multi-Select fields, if a user selected multiple values then multiple rows will appear in the results, one for each selection. |
| phase | Represents the various phases a review can be in. |
| review_activity | For each unique combination of review, user, and role, reports the person-hours[972] spent.<br><br>This includes data for current review participants only. If a user was a participant but is not now, that person will not be included in this result. However that time will be included in the review_activity_summary view. |
| review_activity_summary | One row per review reporting author, reviewer, rework hours, and total person-hours[972] spent in the review. |

| | |
|---|---|
| | • total_person_hours -- a total of all time spent in the review<br>• author_rework_hours -- time spent by the author during the "rework" phase.<br>• author_hours -- time spent by the author outside of the rework phase.<br>• reviewer_hours -- total time spent by reviewers. (active & passive)<br>• active_reviewer_hours -- time spent by "active" reviewers (that is, reviewers required to finish a review.)<br>• passive_reviewer_hours -- time spent by "passive" reviewers (that is, those not required to finish a review.) |
| review_comment_summary | One row per review reporting author, reviewer, and total number of comments made in the review. |
| review_defect_summary | One row per review reporting the number of defects created in that review. |
| review_metrics_summary | One row per review reporting a variety of standard metrics such as inspection rate, defect rate, defect density, and the individual numbers used to form those ratios. Some values will contain NULL values because of divide-by-zero errors. |
| review_version_list | Lists all versions actually associated with a review, although with the associated review. |
| review_version_summary | One row per review reporting the number of files and line metrics 972 (total, added, modified, removed) for all files in the review. |
| reviewcustom | These are the custom fields you have defined for reviews, one row for each review. If you change the custom field definition, the layout of this table will change as well (automatically, and immediately).<br><br>With review workflows 246, each review might have a different subset of custom fields. In this case fields are NULL when they are not applicable. |

| | |
|---|---|
| reviewcustom_compat | Same as reviewcustom, but column names are in the form custom_id_N where "N" is the custom field ID as displayed in the custom fields admin page[256]. The columns are ordered exactly the same as reviewcustom, so you can also tell which is which by comparing the two views. |
| review_custom_dropdowns | This table contains a list of the possible values as defined in drop-down and multi-select Review Custom Fields. |
| review_view | Contains columns from the review table which will be maintained in the event of a future schema change. |
| role_view | Contains columns from the role table which will be maintained in the event of a future schema change. |
| userprefs | Additional user preferences, one row per user. Most user information and preferences are stored in the user table; the rest is here. |
| userprefs_compat | Same as userprefs, but column names are in the form custom_id_N where "N" is an internal ID used to store user preferences. The columns are ordered exactly the same as userprefs, so you can tell which is which by comparing the two views. |
| user_view | Contains columns from the user table which will be maintained in the event of a future schema change. |

### 9.1.2 Example SQL: Participant Custom Fields

## What Are Participant Custom Fields?

In Collaborator, we introduced Per-Participant Custom Fields[256]. This allows you to create a field in a review (via its review template) to which each review participant can assign his or her own value. This allows collection of any type of data that may be different for each participant.

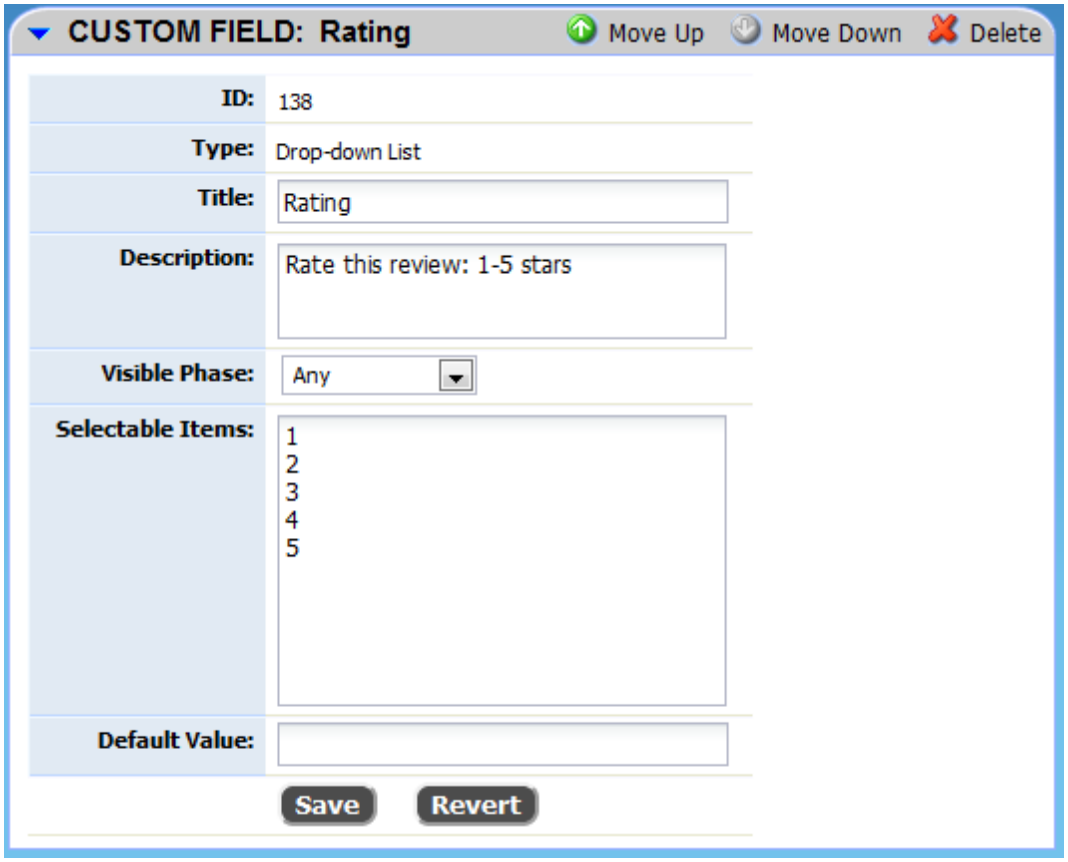
Representing this data in table/column style is sometimes less than useful, especially since, generally, you will want to perform some calculation on the data before displaying it. To aid you in this, we have created views in the database to help you retrieve these values and perform calculations on them.

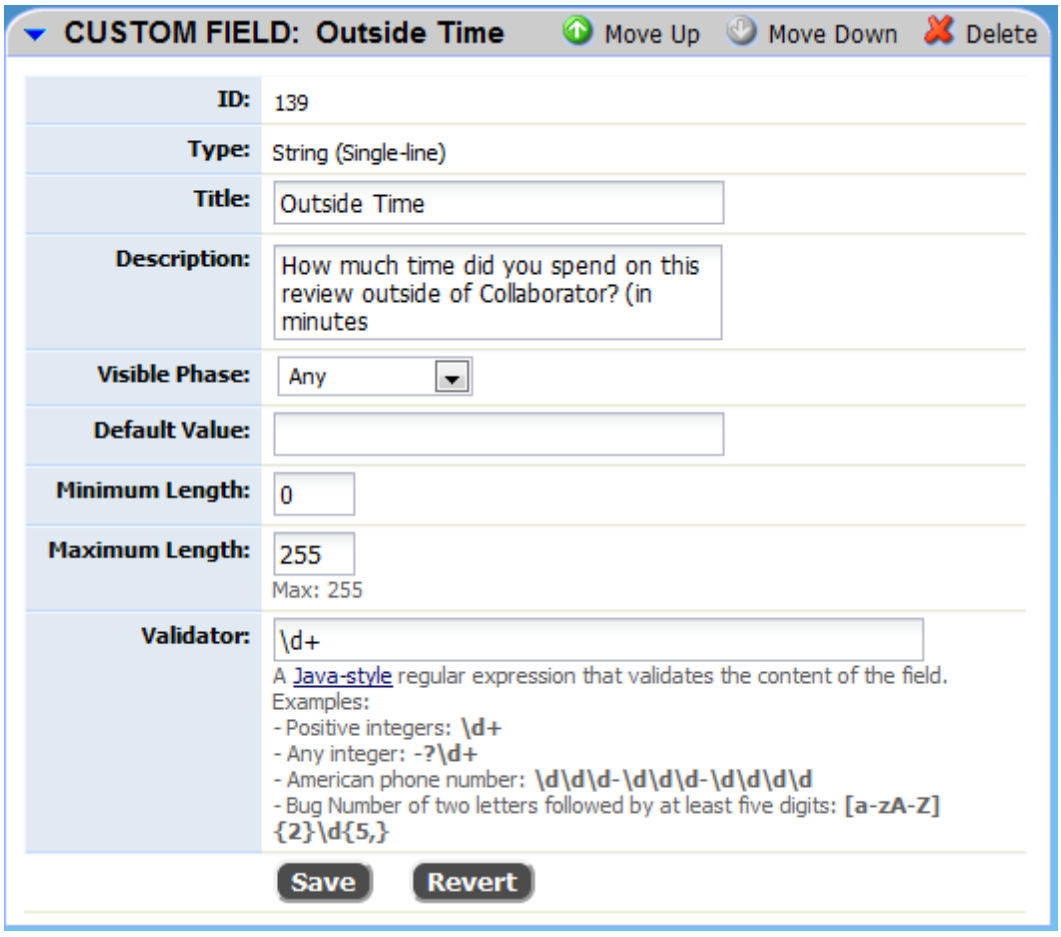
## Examples in MySQL

Below is a sample of how to query Participant Custom Fields in MySQL[60], using MySQL Workbench to connect directly to the database. The syntax will be similar for Oracle and MSSQL, though you may need to reference the documentation for your particular database. You will also need to use a SQL query tool that is able to connect to those databases.

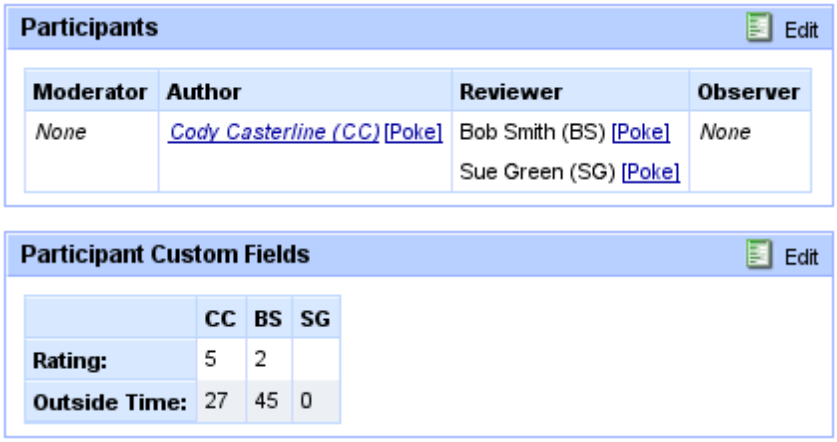
## Participant Custom Fields

In these examples, we will use the following Participant Custom Fields:





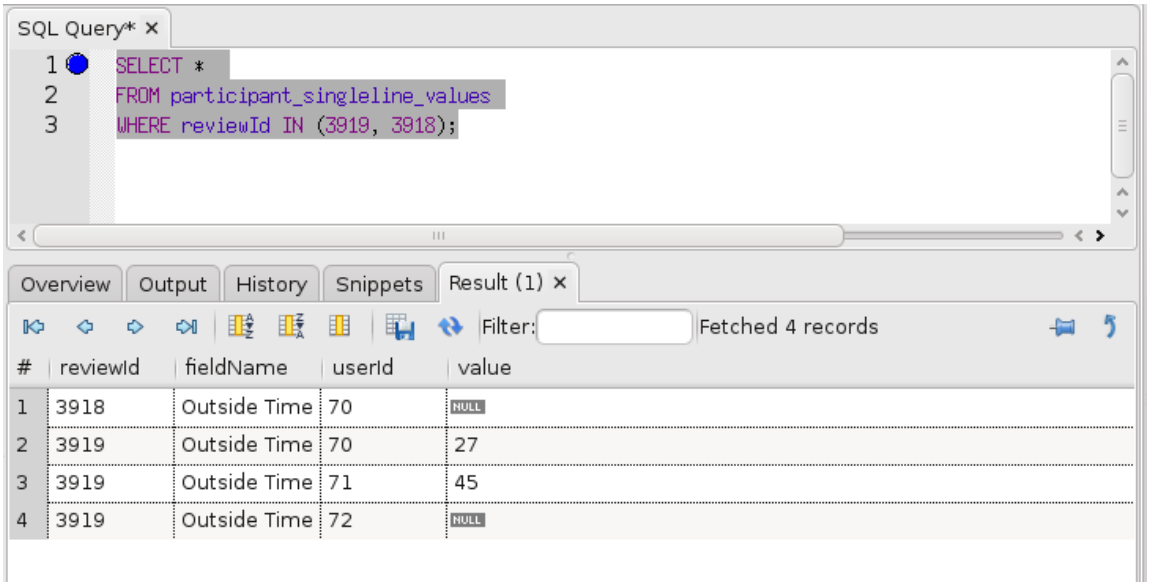

We will also use a review (#3919) with some sample data entered for the custom fields:



## Simple Data

The simplest method of selecting the data from the database is just to show the values as they were entered by users. To do this, we will use one of the views[908] for Participant Custom Fields that have been defined in the database.

```
SELECT *
FROM participant_singleline_values
WHERE reviewId IN (3919, 3918);
```




Note that, for the purposes of these examples, we are limiting the results to two reviews.

Review #3918 only has one participant (it is still in planing phase), who has not yet specified a value for Outside Time.

Review #3919 has three users, two of which have specified values for Outside Time.

## A Note About Database Data Types

Collaborator stores all values for Custom Fields as strings, or sequences of letters and numbers. This is fine for the purposes of displaying the data (as above), but if you want to perform calculations on the data, you will need to tell your database to convert the strings into a meaningful number type, such as an integer, or a floating point number.

In MySQL, you perform this data conversion using the CONVERT() function.

SQL also contains a 'NULL' value, which represents the absence of a specified value. If you want to replace NULL values with some meaningful default value, use the IFNULL() function. Generally, aggregate functions like SUM() and AVG() will exclude NULL values from their calculations, so you will not need this function.

## Sums

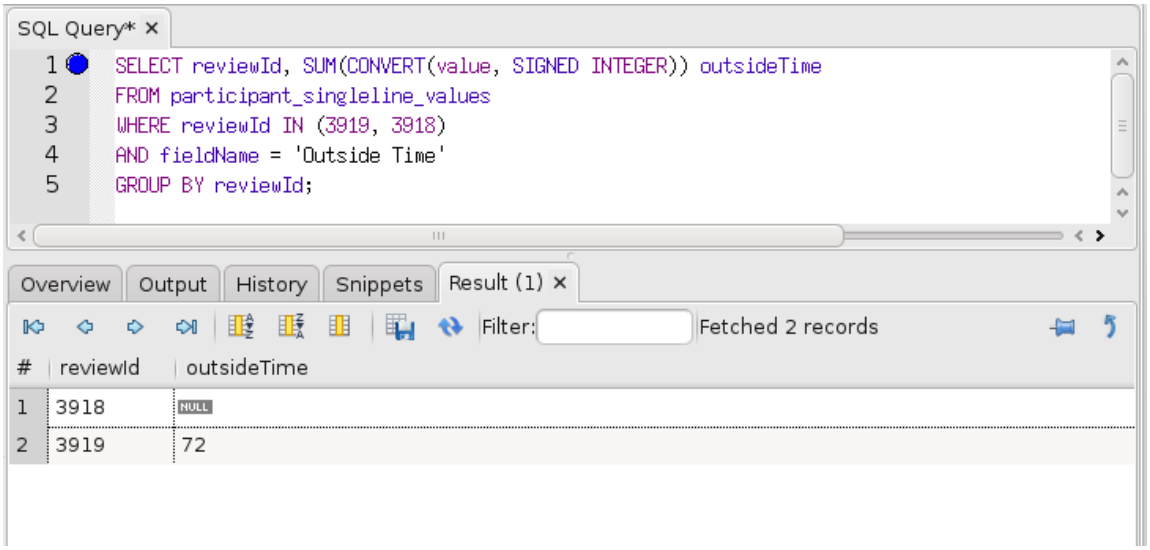
If we would like to find the sum of Outside Time for each review, we can modify the query to be:

```
SELECT reviewId, SUM(CONVERT(value, SIGNED INTEGER)) outsideTime
FROM participant_singleline_values
WHERE reviewId IN (3919, 3918)
AND fieldName = 'Outside Time'
GROUP BY reviewId;
```

In the SELECT line, we now explicitly list each of the columns we wish to select. We use the CONVERT() function to convert the values to (signed) integers, and then the SUM() function to add them up. We also specify that the summed value should be named "outsideTime".

In the WHERE clause, we have added an additional 'AND' clause. This makes sure that we only sum up values of the correct field type: 'Outside Time'.

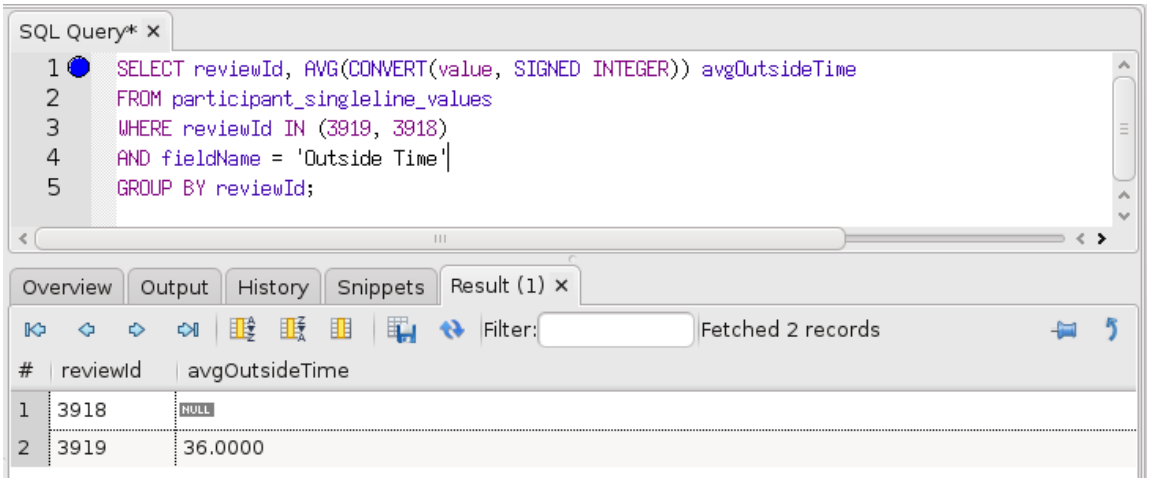
Because we are using an aggregate function (SUM()), we have to specify which pieces of data we want to perform that calculation on. In this case, we want the sum for each review, so we specify "GROUP BY reviewId". The database server first groups all results with the same review ID together, then sums the values for each of those groups.



## Averages

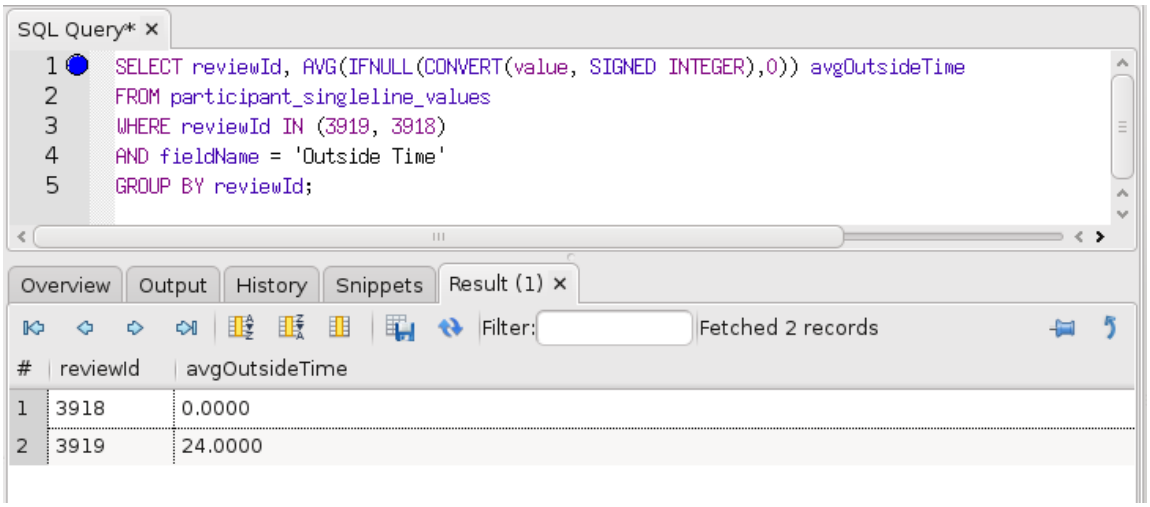
To perform averages, the query is nearly identical, we just replace the SUM() function with the average function, AVG():

```
SELECT reviewId, AVG(CONVERT(value, SIGNED INTEGER)) avgOutsideTime
FROM participant_singleline_values
WHERE reviewId IN (3919, 3918)
AND fieldName = 'Outside Time'
GROUP BY reviewId;
```




Note that these may or may not be the results you were expecting, depending on how you wish to treat NULL values. In the above results, NULL values have been excluded from the calculation of the average, so what we are really seeing is the average of 27 and 45. If you instead want to treat NULL values as if they were 0, you need to specify that in your query using the IFNULL() function. IFNULL(x,y) says that if x is NULL, return y. (Otherwise, it just returns x.) So we will modify our query like this to treat NULLs as if they were 0:

```
SELECT reviewId, AVG(IFNULL(CONVERT(value, SIGNED INTEGER),0))
avgOutsideTime
FROM participant_singleline_values
WHERE reviewId IN (3919, 3918)
AND fieldName = 'Outside Time'
GROUP BY reviewId;
```




Note that, now that we treat missing (NULL) values as zeros, the average Outside Time for Review #3919 is the average of 27, 45, and 0. Also note that the missing values for Review #3918 are also interpreted as zeros, so we get a 0 result instead of a NULL.

### 9.1.3 Example XPath and XSL

If you feel more comfortable handling XML conversions than connecting directly to a database, you can instead get data about reviews using the command ccollab admin review-xml. This command has one required argument, the review ID. By default, it will output the entire XML document for the review. However, there are optional arguments of --xpath and --xsl-file which allow you to query the resulting document for particular information.

#### XPath

XPath allows you to address XML elements in an XML document much like paths in a file system. The hierarchy in an XPath expression tells the XPath parser where in the XML document to find elements.

For example, to get the section of the review-xml document that contains participant custom fields, you would use an XPath expression like this:

```
/reviews/review/participant-custom-fields
```

Below is an example command invocation and its response:

```
$ ccollab admin review-xml 3919 --xpath /reviews/review/participant-
custom-fields
```

```
<participant-custom-fields>
 <user userId="70">
 <outside-time metaDataId="198" title="Outside Time">27</outside-
time>
 <affected-components metaDataId="199" title="Affected
Components">APIs
Business Logic
Database Back-end</affected-components>
 <rating metaDataId="197" title="Rating">5</rating>
 </user>
 <user userId="71">
 <outside-time metaDataId="198" title="Outside Time">45</outside-
time>
 <affected-components metaDataId="199" title="Affected Components"/>
 <rating metaDataId="197" title="Rating">2</rating>
 </user>
 <user userId="72">
 <outside-time metaDataId="198" title="Outside Time">0</outside-time>
 <affected-components metaDataId="199" title="Affected Components"/>
 <rating metaDataId="197" title="Rating"/>
 </user>
</participant-custom-fields>
```

## Finding Particular Values

To get all values of a particular type, you will need to specify a few more path elements. Note in the sample above that all of the Participant Custom Fields are grouped by the user that specified them. Within that user tag, each of the fields gets a unique XML tag which is a normalized form of the title of the custom field title. The value for that field is stored as text inside of that field's tag. So, to get the values for the Rating field, you would use the following XPath expression:

```
/reviews/review/participant-custom-fields/user/rating/text()
```

Below is an example invocation, and its output:

```
ccollab admin review-xml 3919 --xpath "/reviews/review/participant-
custm-fields/user/rating/text()"


5
2
```

Note that we had to quote the XPath expression because some command-line shells interpret parentheses as non-literal characters.

## XSLT

XSLT (Extensible Stylesheet Language Transformation) is a subset of XSL which can be used to transform an XML document into another format. You can use XPath expressions along with XSLT to perform more complicated data queries. For example, if we want to show all "rating" values and the user ID of the user who set them, we can use a file like this:

## File: sample.xslt

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
Transform">
<xsl:output method="text"/>
<xsl:template match="/">
 <!-- Output CSV of Participant Custom Fields -->

 <!-- Header: -->
 <xsl:text>UserID</xsl:text>
 <xsl:text>, </xsl:text>
 <xsl:text>Field Name</xsl:text>
 <xsl:text>, </xsl:text>
 <xsl:text>Field Value</xsl:text>
 <xsl:text>&#10;</xsl:text><!-- linefeed -->

 <xsl:for-each select="/reviews/review/participant-custom-fields/
user">
```

```
<xsl:value-of select="@userId"/>
<xsl:text>, </xsl:text>
<xsl:value-of select="rating/@title"/>
<xsl:text>, </xsl:text>
<xsl:value-of select="rating/text()"/>
<xsl:text>&#10;</xsl:text>
</xsl:for-each>

</xsl:template>
</xsl:stylesheet>
```

Here is how you would use the above XSL file, and some sample results:

```
$ ccollab admin review-xml 3919 --xsl-file sample.xslt

UserID, Field Name, Field Value
70, Rating, 5
71, Rating, 2
72, Rating,
```

Note that user #72 has not specified a value for Rating, so that value is empty.

# 9.2 Scripting

Scripting is the easiest way to implement custom behaviors with Collaborator.

## Command-Line Client

You can use the Command-Line Client to script actions on your user's machines, or install it on the same machine as the Collaborator Server and invoke it with a trigger 206.

The Command-Line Client can create, delete, and edit reviews, including templates and review custom fields. It can assign or remove participants within a review, and even move reviews to the next phase (for example, from "planning" to "inspection"). It can also create comments and defects, including defect custom fields, and mark defects as external.

You can use the Command-Line Client to create, edit, enable or disable user accounts. It can be used to set up user information like email addresses and phone numbers. It can also be used to set up author or file-based subscriptions.

## Batching Commands

Often when scripting you find that you need to run multiple Command-Line Client commands. It is OK to simply run them one after the other in your script, but it is more efficient to group them together using the `ccollab admin batch` command. The `batch` command takes as input an XML file which lists the commands to be run and optionally global options to use to run them.

It is faster to run multiple commands using the `batch` command because the Command-Line Client only has to connect and authenticate with the Collaborator Server once. Also the XML input format is useful if the data you are passing to the Command-Line Client contains characters difficult to encode on the command-line like line feeds, quotes, or multi-byte characters.

## Extracting Data

The `ccollab admin review-xml` command lets you extract data from one or more reviews in your script. You can supply an XSL file to format the output, or an XPATH expression to select only the specific data you need. You can even use XPATH functions to perform some computation.

You can also query the server for a list of reviews using the built-in reports 467, and then use the `ccollab admin wget` command to download the CSV output format 473 to be parsed by your script. Note the CSV output format of the reports does not consume a license. For example, to get a list of the reviews currently in progress run:

```
ccollab admin wget "/go?
page=ReportReviewList&formSubmittedreportConfig=1&reviewIdVis=y&revie
wTitleVis=y&data-format=csv&phaseFilter=inprogress"
```

## Prompting

You can disable all interactive prompting by specifying the `non-interactive` 516 global option.

If `non-interactive` 516 is not specified, the Command-Line Client will automatically prompt for server `url` 517, `user` 517, and `password` 516 as necessary when it connects to the Collaborator Server.

You can also use the special value "*ask*" when specifying the `review` parameter. Using "*ask*" will cause the Command-Line Client to interactively prompt the user with the list of reviews they are currently involved in.

If you are invoking more than one command, you can use the special value "*last*" in combination with "*ask*". Using "*last*" will cause the Command-Line Client to address the review most recently accessed by the Command-Line Client. Typically the first command in the script will use "*ask*" and then subsequent commands use "*last*" to access the same review.

## Opening a Browser

Collaborator is fundamentally a web-based tool and the actual reviewing goes on in a browser. Many Command-Line Client commands automatically open a browser to allow the user to take the next step. You can prevent the Command-Line Client from opening a browser by specifying the `no-browser` 516 global option.

You can also explicitly open a browser using the `ccollab browse` command. Note the `browse` command ignores the `no-browser` 516 global option.

## "Virtual" Script Users

Often when setting up scripting it is convenient to create a Collaborator user which the script will use to log in to the Collaborator server. Simply create a user in the normal way and give the credentials (login and password) to the script. The command-line scripting commands will not cause the "virtual" script user to consume a license (except `ccollab admin wget`, if the URL it is loading consumes a license).

## Downloading file version content

A special URL lets you download the binary contents of any file version efficiently. The URL is authenticated, so you have to use `ccollab admin wget`. Note the version content URL does not consume a license.

For example, to get a file version with ID `12345`:

```
ccollab admin wget "/data/server?versionid=12345"
```

The 'wget' command can also be used to:

Get all diffs in a review:

```
ccollab admin wget "/diff?context=<context>&reviewid=<review id>"
```

Get diffs from two specific versions:

```
ccollab admin wget "/diff?context=<context>&reviewid=<review
id>&versionids=<version a>,<version b>
```

### 9.2.1 Mirroring Defects to an external issue-tracker

In this tutorial we will set up a script that will automatically mirror defects found during a peer review in Collaborator to an external issue-tracking system. In this tutorial we will mirror defects to FogBugz, but the steps in this tutorial can be modified to mirror defects to any external issue-tracker.

## Outline

When a Defect is created in Collaborator the Defect activity trigger[207] invokes a script which creates a new "Case" in FogBugz with a link back to the review. Then the Defect in Collaborator is marked "external" with a link to the mirrored Case in FogBugz.

## Prerequisites

The script in this example is written in Perl, so the server must have the Perl runtime installed. The script invokes the Collaborator Command-Line Client, so that needs to be installed on your server machine as well. Be sure to install these prerequisites so that they are accessible and executable by the system user which is running the Collaborator server (this is especially important on Unix systems).

## Script Step 1: Set up constants

The script will need to know a few values to do it is work. These will be hard-coded in to the script:

```
# URL to the Collaborator Server
$CCOLLAB_URL = "http://localhost:8080";

# Login of Collaborator User who will mark the Defect as external
$CCOLLAB_USER = "admin";

# Password of Collaborator User who will mark Defect as external
$CCOLLAB_PASSWORD = "";

# URL to the FogBugz server
$FOGBUGZ_URL = "http://bugs";

# Email of user to use to create Case in to FogBugz
$FOGBUGZ_USER_EMAIL = "person\@yourcompany.com";

# Password of user to use to create Case in to FogBugz
$FOGBUGZ_USER_PASSWORD = "yourpasswordhere";
```

## Script Step 2: Read parameters from command-line

The Review ID, Defect ID, and Defect title will be supplied by Collaborator as parameters when it invokes the script from the trigger (we set that up in Step 9 932). The script needs to read those parameters from the command-line:

```
# read parameters from command-line
$reviewId = $ARGV[0];
$defectId = $ARGV[1];
$defectTitle = $ARGV[2];
```

## Script Step 3: Logon to FogBugz

The script will use the FogBugz web services API to mirror the Defect in to FogBugz. In order to use the API we first have to "logon" and acquire an "authentication token":

```
# logon to FogBugz
my $response = get("$FOGBUGZ_URL/api.php?
cmd=logon&email=$FOGBUGZ_USER_EMAIL&password=$FOGBUGZ_USER_PASSWORD")
;

# extract authentication token
$response =~ /<token>(?:<!\[CDATA\[)?([^\]]+)(?:\]\]>)?<\/token>/;
my $token = $1;
```

## Script Step 4: Create Case in FogBugz

The script creates a "Case" in FogBugz, setting the title of the Case to the text from the Collaborator Defect, and putting a link back to the Collaborator server in the description of the Case. Note that the Defect title and Case description have to be escaped because they are going in to a "get" URI. The script extracts the resulting Case ID and saves it for later:

```
# create new Case
$defectTitle = uri_escape($defectTitle);
$description = uri_escape("Defect D$defectId mirrored from
Collaborator Review $reviewId ($CCOLLAB_URL/index.jsp?
page=ReviewDisplay&reviewid=$reviewId)");
$response = get("$FOGBUGZ_URL/api.php?
token=$token&cmd=new&sTitle=$defectTitle&sEvent=$description");
```

```
# extract Case ID
$response =~ /ixBug="(\d+)"/;
my $case = $1;
```

## Script Step 5: Logoff from FogBugz

The script logs off from FogBugz, invalidating the authentication token:

```
# log off from FogBugz (invalidate authentication token)
$response = get("$FOGBUGZ_URL/api.php?token=$token&cmd=logoff");
```

## Script Step 6: Mark Collaborator Defect as "external"

The script runs the Command-Line Client `ccollab admin review defect mark-external` command to mark the Defect as external. The "external-name" of the Defect is the Case number in FogBugz (which will appear as a link after Step 8 932):

```
# mark Defect as "external" in Collaborator
$ccollabOptions = "--url $CCOLLAB_URL";
$ccollabOptions .= " --user $CCOLLAB_USER";
$ccollabOptions .= " --password \"$CCOLLAB_PASSWORD\"";
$ccollabOptions .= " --quiet --non-interactive";
system("ccollab $CCOLLAB_OPTIONS admin review defect mark-external
$defectId \"Case $case\"");
```

This step in the script works fine if you have configured Collaborator to work with MySQL. With any other database, the process is slightly more complicated - please contact technical support 32 for more details.

## Step 7: Test the script from the command-line

That is our whole script! Here it is in finished form: `mirror-defect.pl` (opens in a new window). Do not forget to replace the constants in the script (urls, users, passwords and so on) with the appropriate values for your environment, then copy the script to an accessible place on your server. The script needs to be readable by the system user which is running the Collaborator server.

Before you configure the Collaborator server to invoke the script automatically, test it manually by opening a console on your server machine running it on the command-line. **Be sure to log in to your server machine as the user which is running the Collaborator server.** First create a review and create a defect in the review, then run the script.

For example, if you created Review 1234 with Defect D5678, then run the script with this command:

```
C:\Perl\perl.exe "C:\Program Files\Collaborator Server\mirror-defect.
pl" 1234 5678 "mirrored Defect title"
```

You should see output similar to this:

```
Mirrored Defect D5678 in FogBugz as Case 91011
Connecting to Collaborator Server http://localhost:8080
Connected as: Collaborator Administrator (admin).
D5678 marked as external
```

When you refresh your browser in Collaborator you should see the Defect has been marked as external, with the description "Case 91011".

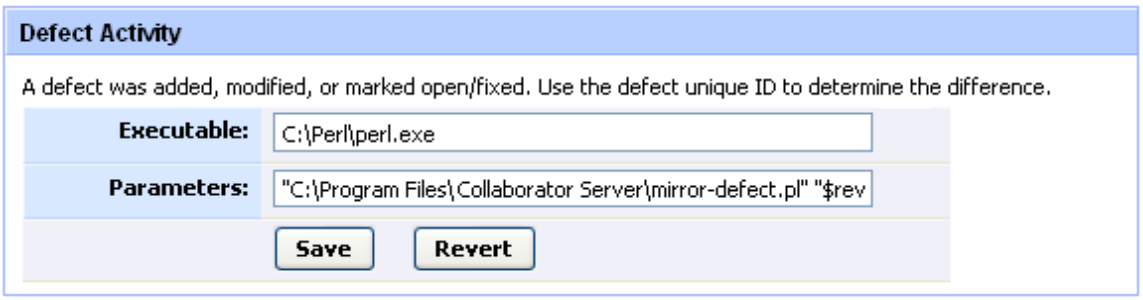
## Step 8: Set up Bug-Tracking integration link

Using the Collaborator built-in Bug-Tracking hyperlink function, set up the text "Case <number>" to automatically hyperlink to that Case number in FogBugz[906]. Now if you go back to look at Defect D5678 in Collaborator the text "Case 91011" should be a hyperlink to Case 91011 in FogBugz.

## Step 9: Invoke Script from a trigger

Make sure you have Step 7[931] working before you move on to this step. The last thing we need to do is tell the Collaborator server to automatically invoke the `mirror-defect.pl` script when a Defect is created. We do this with the Defect Activity trigger[207]. We will use these values:

```
Executable: C:\Perl\perl.exe
Parameters: "C:\Program Files\merge2.2\specs\collab\Baggage\mirror-
defect.pl" "$review.id" "$defect.id" "$defect.txt"
```




*Defect Activity Trigger*

Note that it is important to use the FULL PATH of both the Perl runtime and the `mirror-defect.pl` script. Also note that the parameters are enclosed in quotes in case they have spaces in them. The Review id[162], Defect id[164], and Defect text[164] come from substitution variables[161].

## Finished

That is it! Now when anyone creates a Defect in Collaborator it will be automatically mirrored to FogBugz, and the Defect will be "marked as external" with a link to the Case in FogBugz. Note that for performance reasons the trigger runs *after* the Defect is created in a separate thread, so you may have to refresh your browser a couple of seconds after creating the Defect to see it automatically externalized.

### 9.2.2 Syncing users from Perforce

In this tutorial we will write a script that will create a corresponding Collaborator user for every Perforce user. The script can be run periodically to pick up any new Perforce users.

This script performs the same actions as the `ccollab admin syncusers`[785] command.

## Outline

The script runs the Perforce `p4 users` command, parses the output, and then calls the Collaborator Command-Line Client `ccollab admin user create` command for every user. If the user already exists then the `ccollab admin user create` command fails and the script goes on to the next user.

## Prerequisites

The script in this example is written in Perl, so you must have the Perl runtime installed. The script invokes the Collaborator Command-Line Client, so that needs to be installed as well. Note you must be a Collaborator Administrator in order for this script to work.

## Script Step 1: Get users from Perforce

The script runs the `p4 users` command to get the a list of all user records from Perforce.

```
# Get users from Perforce
@p4Users = `p4 users`;
```

## Script Step 2: Parse fields from Perforce user record

The script extracts the user name, email, and full name from the Perforce user record.

```
#parse fields from Perforce user record
```

```
$p4User =~ /(\S+)\s*<([^>]*)>\s*\((.*?)\)\s*accessed.*/;

$user = $1;
$email = $2;
$fullName = $3;
```

### Script Step 3: Create user in Collaborator

The script runs the Command-Line Client `ccollab admin user create` command to create the user in Collaborator. If a user with that name already exists, the command fails and the script goes on to the next user.

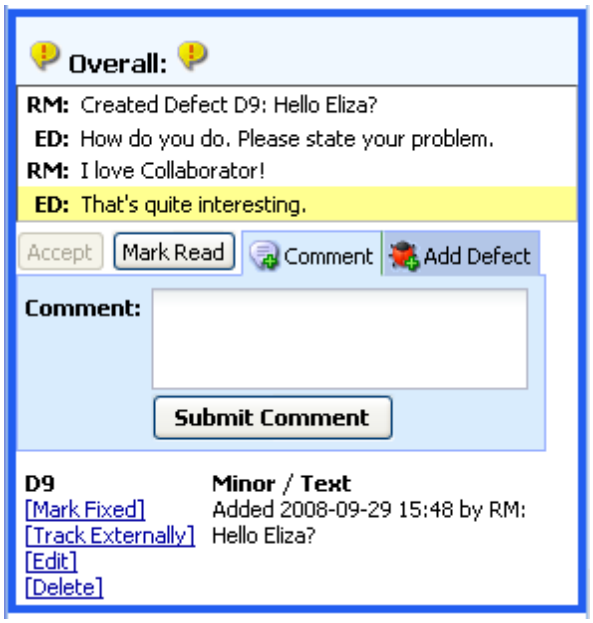
```
#create user in Collaborator - does nothing if user already exists
system("ccollab admin user create \"$user\" --email \"$email\" --
full-name \"$fullName\"");
```

### Finished

That is it! Here is the script in finished form: `syncusers-p4.pl` (opens in a new window). You can run this script periodically (for example, with a `cron` job) and it will keep your Perforce users and Collaborator in sync.

## 9.2.3 Conversation with Eliza

In this tutorial we will set up a script that will let a user have a conversation with the classic ELIZA program. Eliza will appear to chat using comments in a Review. Eliza will only respond to comments in conversations that have a Defect containing her name.




*Conversation with Eliza*

## Outline

When a Defect is created in Collaborator the Defect Activity trigger 207 launches a Perl script in the background. The script polls Collaborator for comments using the Command-Line Client. When it finds a new comment, it passes the comment to an implementation of the ELIZA program, and then posts Eliza's response to the conversation using the Command-Line Client.

The script runs indefinitely until the Defect is marked fixed or deleted, or the Review is finished.

## Prerequisites

The script in this example is written in Perl, so you must have the Perl runtime installed.

The Perl script invokes the Collaborator Command-Line Client, so that needs to be installed on your server machine. Be sure to install the Command-Line Client so that it is accessible and executable by the system user which is running the Collaborator server (this is especially important on Unix systems).

The script uses a clone of ELIZA written in Perl called Chatbot::Eliza which must be installed.

## Script Step 1: Set up constants

The script will need to know a few values to do it is work. These will be hard-coded in to the script:

```
# URL to the Collaborator Server
```

```
$CCOLLAB_URL = "http://localhost:8080";

# Login of Collaborator User who Eliza will use for chat
# should be a Collaborator administrator
$CCOLLAB_USER = "eliza";

# Password of Collaborator User who Eliza will use for chat
$CCOLLAB_PASSWORD = "eliza";

# Seconds to sleep before polling for new chat
$REFRESH_DELAY_SECONDS = 4;
```

## Script Step 2: Read parameters from command-line

The Review ID and Defect ID will be supplied by Collaborator as parameters when it invokes the script from the trigger (we set that up in Step 12 942). The script needs to read those parameters from the command-line:

```
# read parameters from command-line
$reviewId = $ARGV[0];
$defectId = $ARGV[1];
```

## Script Step 3: Create XSL Transform

The script needs to extract multiple pieces of information from the Collaborator server. To do that more efficiently it uses an XSL Transform to extract everything in a single call to ccollab admin review-xml (Step 6 938). The XSL is inlined in the Perl script using "here-document" syntax.

```
#xslt file we will use to extract info from Collaborator
$xslt = <<XSLT;
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
version='1.0'>

 <xsl:template match='//reviews/review'>

       <!-- get review phase -->
       <xsl:value-of select='general/phase'/><xsl:text>
```

```
</xsl:text>

      <!-- find defect -->
      <xsl:for-each select='defects/defect[\@defectId=$defectId]'>

            <!-- get defect status -->
            <xsl:value-of select='status'/><xsl:text>
</xsl:text>

            <!-- get defect text-->
            <xsl:value-of select='text'/><xsl:text>
</xsl:text>

      </xsl:for-each>

      <!-- find the conversation we are talking on (the one with the
defect) -->
      <xsl:for-each select='conversations/conversation[defects/
conversation-defect/\@defect-id=$defectId]'>

            <!-- find last comment in conversation -->
            <xsl:for-each select='comments/comment[last()]'>

                  <!-- get author -->
                  <xsl:value-of select='\@creator'/><xsl:text>
</xsl:text>

                  <!-- get text -->
                  <xsl:value-of select='text()'/><xsl:text>
</xsl:text>

            </xsl:for-each>

            <!-- get file-path (empty for overall comment) -->
```

```
            <xsl:value-of select='\@file-path'/><xsl:text>
</xsl:text>

            <!-- get line-number (empty for overall comment or file
overall comment) -->
            <xsl:value-of select='\@line-number'/><xsl:text>
</xsl:text>

      </xsl:for-each>

 </xsl:template>
</xsl:stylesheet>
XSLT
```

## Script Step 4: Initialize Eliza

The Chatbot::Eliza program is initialized by a simple constructor.

```
# initialize Eliza
$eliza = new Chatbot::Eliza;
```

## Script Step 5: Sleep for a few seconds

The rest of the script loops until the script decides to exit (Step 7 939). Every time it starts the loop, the script first sleeps for a few seconds to make sure it does not put too much load on the server.

```
# sleep so we do not hammer the server
sleep($REFRESH_DELAY_SECONDS);
```

## Script Step 6: Query Collaborator server for info

The script invokes the `ccollab admin review-xml` command, passing in the XSL created in Step 3 936 on STDIN. The output is then parsed to get the Review Phase, Defect status, Defect text, comment Author, comment text, and optionally file path and line number.

```
# Query Collaborator server for info
$pid = open2(*CCOLLAB_OUTPUT, *XSL_INPUT, "ccollab $ccollabOptions
admin review-xml $reviewId --xsl-file -");
print XSL_INPUT $xslt;
```

```
close(XSL_INPUT);
chomp(($spacer, $reviewPhase, $defectStatus, $defectText, $author,
$inputChat, $filePath, $lineNumber) = <CCOLLAB_OUTPUT>);
close(CCOLLAB_OUTPUT);
waitpid($pid,0);

#cleanup filePath and lineNumber
chop($filePath);
chop($lineNumber);

#debug
print "reviewPhase = $reviewPhase\n";
print "defectStatus = $defectStatus\n";
print "defectText = $defectText\n";
print "author = $author\n";
print "inputChat = $inputChat\n";
print "filePath = \"$filePath\"\n";
print "lineNumber = \"$lineNumber\"\n";
print "\n";
```

## Script Step 7: Stop script if appropriate

The script runs in an infinite loop, but we do not really want it to go forever! The script exits if any of the following conditions are true:

- The Review finishes (no longer in inspection phase)
- The Defect is fixed or deleted (no longer open)
- The Defect text does not mention Eliza

```
# safety - quit if review is not in Inspection phase
die ("Review is no longer in Inspection phase") if ($reviewPhase !~ /
Inspection/);

# safety - quit if defect is not open
```

```
die ("Defect $defectId is no longer open") if ($defectStatus !~ /
open/);

# safety - quit if defect text does not mention "Eliza"
die ("Defect $defectId text does not mention Eliza") if
($defectText !~ /Eliza/);
```

## Script Step 8: Ignore comment if appropriate

The script checks whether the comment is a "system" comment like "** Marked Read**" or if the last comment in the conversation was made by Eliza herself. In both of these cases the script skips back to the top of the loop.

```
# Eliza should not respond to system messages, like "** Marked Read
**" or "** Accepted **"
next if ($inputChat =~ /^\*\*/);

# Eliza should not talk to herself
next if ($author =~ /$CCOLLAB_USER/);
```

## Script Step 9: Get response from ELIZA

The script passes the user's comment to the ELIZA program and gets back her response.

```
# Get response from Eliza
$outputChat = $eliza->transform($inputChat);

#debug
print "$outputChat\n";
print "\n";
```

## Script Step 10: Upload Eliza's response to Collaborator

The script uploads Eliza's response to the conversation using the ccollab admin review comment create command.

```
# build command to upload Eliza's comment to Collaborator
$uploadCommand = "ccollab $ccollabOptions admin review comment create
$reviewId \"$outputChat\"";
```

```
# file path is optional (no file path for overall review chat)
if ($filePath) {
 $uploadCommand .= " --file \"$filePath\"";
}

# line number is optional (no line number for overall review chat or
overall file chat)
if ($lineNumber) {
 $uploadCommand .= " --line-number $lineNumber";
}

#debug
print "Running $uploadCommand\n";

# upload Eliza's comment to Collaborator
system("$uploadCommand");
```

## Step 11: Test the script from the command-line

That is our whole script! Here it is in finished form: `eliza.pl` (opens in a new window). Do not forget to replace the constants in the script (url, user, password) with the appropriate values for your environment, then copy the script to an accessible place on your server. The script needs to be readable by the system user which is running the Collaborator server.

Before you configure the Collaborator server to invoke the script automatically, test it manually by opening a console on your server machine running it on the command-line. **Be sure to log in to your server machine as the user which is running the Collaborator server.** First create a Review and create a Defect in the Review with the word "Eliza" in it, then run the script.

For example, if you created Review 1234 with Defect D5678, then run the script with this command:

```
/usr/bin/perl /home/rpaterson/eliza.pl 1234 5678
```

You should see output similar to this:

```
reviewPhase = Inspection
defectStatus = open
defectText = Hello Eliza?
```

```
author = rpaterson
inputChat = Created Defect D11: Hello Eliza?
filePath = "SymlinkTest.java"
lineNumber = "73"

How do you do. Please state your problem.

Running ccollab --url http://localhost:8080 --user eliza --password
"eliza"
 --quiet --non-interactive admin review comment create 1234
"How do you do. Please state your problem". --file "SymlinkTest.java"
--line-number 73
```

The script will loop until you mark the Defect fixed or deleted, or finish the Review. When you refresh your browser in Collaborator you should see Eliza's comment.

## Step 12: Invoke Script from a trigger

Make sure you have Step 11 941 working before you move on to this step. The last thing we need to do is tell the Collaborator server to automatically invoke the `eliza.pl` script when a Defect is created. We do this with the Defect Activity trigger 207. We will use these values:

```
Executable: /usr/bin/perl
Parameters: -e "exit unless fork; system('/usr/bin/perl /home/
rpaterson/eliza.pl ${review.id} ${defect.id} ');""
```


*Defect Activity Trigger*

Note that it is important to use the FULL PATH of both the Perl runtime and the `eliza.pl` script. Also note the quotes and spaces - they are important. The Review id 162, and Defect id 164 come from substitution variables 161. The Perl snippet included in the Parameters field causes the script to be launched in a background process.

### Finished

That is it! Now when anyone creates a Defect in Collaborator the script will be invoked. If the Defect has the word "Eliza" in it then the script will monitor that conversation for comments and have Eliza respond to them.

## 9.3 JSON API Web Services

This section describes the JSON API version of the web services. Using web services you can easily integrate your application with Collaborator.

The benefit of JSON API version of the services is that you can exchange data with almost any client application.

The section covers the following questions:

### Using JSON API Web Service 944

Gives general overview of the web service API and demonstrates how to use it from Web Client and from client applications.

### JSON Syntax and Data Formats 946

Describes the syntax of request and response objects.

### Authentication 949

Explains different variants of logging into a server.

### Error Handling 951

Tells about server reaction on errors and describes the format of error list.

### JSON API Reference 952

Gives a link to the complete reference on all objects and commands of JSON API and describes how to work with this reference.

### How To 952

Describes how to perform typical tasks using JSON API.

### 9.3.1 Using JSON API Web Service

JSON API lets you integrate any external tool with Collaborator. To do this, you need to exchange data between your application and your Collaborator server. To use the web service you need to send requests to web service endpoint URL and receive responses from it. The service receives and sends data in JavaScript Object Notation (JSON). For detailed information on JavaScript Object Notation, see JSON Syntax and Data Formats 946.

## Endpoint URL

The web service resides on your Collaborator server at the following endpoint URL:

- **http(s)://*yourServer.com*/services/json/v1**

## Using JSON API Web Client

To get acquainted with service requests and responses, you can open the endpoint URL in a web browser. It will display a simple form, where you can write JSON commands and get service responses within the same page.

Let us try to call some JSON commands manually:

1. Open a web browser and navigate to JSON API web service URL:
   - **http(s)://*yourServer.com*/services/json/v1**
2. Enter the following command into the input field:

```
[
 {"command" : "ServerInfoService.getVersion"},
 {"command" : "Examples.echo","args" : {"echo" : "Some text."}}
]
```

3. Press "Submit Query" button.

After the page is reloaded you will get the response from the service with the results of your commands:

```
[ {
  "result" : {
       "version" : "9.0.9000"
  }
}, {
  "result" : {
```

```
      "echo" : "Some text."
  }
} ]
```

Later on you can use JSON API web interface to try and debug JSON commands manually before sending them programmatically from your client application.

## Using JSON API From Client Applications

The client and server exchange data in the JSON format via HTTP(s) requests and responses. Client application must send POST requests to JSON web service endpoint URL. The requests should contain a JSON object in their body or in the POST variable named "json". For multipart requests the JSON object must be either in a "json" string part or in a "json" query parameter.

A JSON object in a request specifies a list of commands to be executed and arguments for these commands.

Here is an example of a POST request:

```
POST http://yourServer.com/services/json/v1 HTTP/1.1
User-Agent: JSON API
Content-Type: application/json;charset=utf-8
Host: yourServer.com
Content-Length: 138

[
 {"command" : "ServerInfoService.getVersion"},
 {"command" : "Examples.echo","args" : {"echo" : "Some text."}}
]
```

A multipart variant of the same request will be:

```
POST http://yourServer.com/services/json/v1 HTTP/1.1
Content-Type: multipart/form-data; boundary=--------------------------
acebdf13572468
User-Agent: JSON API
Host: yourServer.com
Content-Length: 265
```

```
--------------------------acebdf13572468
Content-Disposition: form-data; name="json"
Content-type: text/plain;charset=utf-8

[
 {"command" : "ServerInfoService.getVersion"},
 {"command" : "Examples.echo","args" : {"echo" : "Some text."}}
]
--------------------------acebdf13572468--
```

On receiving the request, the server processes each command in order and builds up a response. The JSON object in response contains a list of results corresponding to each of the command that you submitted. Each result consists of either the return values for that command, or a list of errors returned by the command. See Error Handling 951.

The response to the any of the requests above will be:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json;charset=utf-8
Content-Length: 68
Date: Thu, 27 Nov 2014 10:55:25 GMT

[{"result":{"version":"9.0.9000"}},{"result":{"echo":"Some text."}}]
```

In further examples of this section, we will omit request and response headers (when their content is not important) and give only the examples of JSON parts.

### 9.3.2 JSON Syntax and Data Formats

All data-interchange with the service is performed in JavaScript Object Notation (JSON).

JSON uses two data structures: a collection of name/value pairs and an ordered list of values. JSON engine ignores whitespace, tabulation, and newline characters, however you can use them to improve readability.

## JSON Objects In Requests

A JSON object that is sent in web service requests (**request object**) specifies a list of commands to be executed and arguments for these commands.

The simplest variant of a JSON request object is:

```
[
 {"command" : "MethodName1"}
]
```

The `"command" : "MethodName1"` pair defines which command a server should run. The method name is specified using dot notation. For example "ServerInfoService.getVersion", or "Examples.checkLoggedIn".

If a method needs some arguments, then you will need to add the "args" : { } pair which specifies a list of arguments:

```
[
 {"command" : "MethodName1",
       "args" :
{"argumentName1":"value1","argumentName2":"value2",..."argumentNameN"
:"valueN"}
 }
]
```

The `"argumentName" : "value"` pairs define argument names and their values. A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. When a method needs more then one argument, the `"argumentName" : "value"` pairs are separated by commas. Argument pairs can be specified in any order.

You can create batch commands, that is, pass several commands in one JSON request object. Different commands should be separated by commas. For example:

```
[
 {"command" : "ServerInfoService.getVersion"
 },
 {"command" : "Examples.echo",
       "args" : {"echo" : "Some text."}
 },
 {"command" : "SessionService.setMetadata",
```

```
      "args" : {
            "clientName" : "HTML JSON API Tester",
            "expectedServerVersion" : "9.0.9000"
      }
 }
]
```

## JSON Objects In Responses

A JSON object that is received in web service responses (**response object**) contains a list of results corresponding to each of the command that was submitted. Each result consists of either the "result" : { } pair that lists the command return values (if any), or the "errors" : [ ] pair that lists the errors returned by the command. (See Error Handling 951 for details).

Results for several commands that were passed in a single JSON request object are returned as a comma separated list. For example the response object for the previous request will be:

```
[
 { "result" : { "version" : "9.0.9000" } },
 { "result" : { "echo" : "Some text." } },
 { "result" : { } }
]
```

## Date and Time Format

To specify date and time values, use strings having the format "YYYY-MM-DDTHH:mm:ssZ". For example: "2008-01-01T22:50:00Z" or "2008-01-01T16:50:00-06:00".

The table below describes the parts of this format.

| | |
|---|---|
| YYYY | Year |
| MM | Month |
| DD | Day of the month |
| T | Specifies the start of a time notation. |
| HH | Hours |

| | |
|---|---|
| mm | Minutes |
| ss | Seconds |
| Z | Time zone.<br><br>The value in this position can be one of the following:<br><br>Z - indicates UTC time<br><br>+hh:mm - indicates that the input time is the specified offset after UTC time.<br><br>-hh:mm - indicates that the input time is the absolute value of the specified offset before UTC time. |

### 9.3.3 Authentication

For the majority of commands, you will need to be logged in to the server. Therefore you may need to pass your authentication information along with the request.

> **Important:** All data (including user credentials) that is sent via HTTP protocol is sent in plain text format. We recommend using HTTPS protocol for secure client/server communication. Read Configuring HTTPS 111 to learn how to enable it.

## General Information

JSON API web service uses session authentication. Authentication data are valid only while executing batch commands were sent in the same request. Authentication takes place inside of the JSON request. No special cookie handling is required.

To execute the commands from another request you will need to pass authentication information again.

**Tip:** To verify if you are logged in for the current batch of commands, call the `Examples.checkLoggedIn` command.

The preferred way to authenticate is to send a user login and a special alpha-numeric identifier called **login ticket.** The benefit of login ticket is that you do not have to specify your password explicitly. Using login tickets works faster if a server uses LDAP authentication 119. Another benefit to login tickets is that a system admin can invalidate login tickets 196 if they want to make sure that Collaborator clients are forced to re-authenticate.

## Generating Login Tickets

To perform first-time authentication and generate a login ticket, use the `SessionService.getLoginTicket` command. It accepts user login and password as input arguments and returns a login ticket which you can use in your subsequent requests.

The following request logs in as "`jsmith" with password "qwerty12345"` and obtains a login ticket for further usage:

```
[
 {"command" : "Examples.checkLoggedIn"},
 {"command" : "SessionService.getLoginTicket",
       "args":{"login":"jsmith","password":"qwerty12345"}},
 {"command" : "Examples.checkLoggedIn"}
]
```

The server will send the following response:

```
[
 { "result" : { "loggedIn" : false } },
 { "result" : { "loginTicket" :
"0123456789abcdef0123456789abcdef" } },
 { "result" : { "loggedIn" : true } }
]
```

## Authenticating With Login and Login Ticket

To authenticate using a login and login ticket use the `SessionService.authenticate` method. It accepts user login and login ticket as input arguments.

The following request logs in using a login ticket:

```
[
 {"command" : "Examples.checkLoggedIn"},
 {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
 {"command" : "Examples.checkLoggedIn"}
]
```

The server response will be:

```
[
 { "result" : { "loggedIn" : false } },
 { "result" : {} },
 { "result" : { "loggedIn" : true } }
]
```

### 9.3.4 Error Handling

Responses form the JSON API web service contain a list of results corresponding to each of the command that was submitted. Each result may contain either a "result" : { } pair that lists the command return values (if any), or the "errors" : [ ] pair that lists the errors returned by the command.

If an error occurs during processing a request having multiple commands, the service stops execution of further commands and sends a response object with the results of preceding commands and the error description for the current command.

## The Format of Error Descriptions

The returned "errors" : [ ] pair holds an array with one or more error descriptions. Each error description has the following fields:

1. "code" - Short error code, which you can use for error handling.
2. "message" - More detailed description of the error. This message can be displayed to the end-users.
3. "data" - Any extra data that is relevant for the particular error. This is an optional field and it may be omitted.

## Sample Error Descriptions

Below are several examples of error descriptions:

- Unneeded parenthesis in command name:

```
[ { "errors" : [
       { "code" : "NoSuchCommand",
       "message" : "The command 'ServerInfoService.getVersion()' does
not exist." } ]
} ]
```

- Incorrect authentication:

```
[ { "errors" : [
        { "code" : "AuthenticationFailed",
       "message" : "Could not authenticate user 'jsmith' using
password provided." } ]
} ]
```

### 9.3.5 JSON API Reference

Complete reference on all API objects and methods is available at the following URL:

- **http(s)://yourServer.com/javadoc/jsonapi/**

The reference lists all the commands that are available to JSON web service and describes the fields of the request and response objects.

API reference documentation uses Javadoc notation to describe the objects, methods and interfaces. In order to use the command, object or field in your JSON requests, you need to convert the Javadoc notation to Java Script Object Notation.

To determine a name of the desired JSON field, remove the "get" prefix from the method name and put the first letter to lower case.

For example, a Javadoc description for the `LoginTicketRequest` interface says that it has the `getLogin` and `getPassword` methods.

A JSON object that corresponds to this interface will look like this:

```
{"login": "jsmith", "password": "js123"}
```

### 9.3.6 How To

This topic demonstrates how to perform some typical tasks via JSON API Web services:

## Manage Reviews and Review Participants 953

1. List reviews assigned to a specified user
2. Create new review
3. Add/change participants of a review

## Upload Review Materials 957

1. Add files through a /content upload servlet
2. Add files with a multipart HTTP requests

## Manage Users and User Groups 965

1. Create new user
2. Add user to group

In order to perform all these tasks you need to be logged in to Collaborator server. Therefore in each sample below we will first call the `SessionService.authenticate` command to provide user credentials. See Authentication 949 topic for details.

#### 9.3.6.1 Manage Reviews and Review Participants

To manage reviews and review participants use the `ReviewService` interface. It provides methods for creating and editing reviews, adding, updating, removing and poking participants, adding review materials, changing review phases, managing comments and conversations and so on.

1. List reviews assigned to a specified user 953
2. Create new review 954
3. Add/change participants of a review 955

### List reviews assigned to a specified user

To get a list of reviews we will use the `UserService.getActionItems` command which returns a list of incoming and outgoing action items 333 a user is included in.

Request:

```
[      {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
       {"command" : "UserService.getActionItems"}
]
```

Response example:

```
[      { "result" : { } },
 { "result" : { "actionItems" : [
  { "nextActionText" : "Waiting for comments", "text" : "(No action
required) Waiting for comments: Review #10254: \"Fix for case 34534
\"" },
  { "nextActionText" : "Finish creating", "text" : "Finish creating:
Review #10435: \"Documentation review\"" },
  { "nextActionText" : "Waiting for Defect rework", "text" : "(No
action required) Waiting for Defect rework: Review #10188: \"Message
Templates\"" }
  ] } }
 ]
```

## Create new review

To create a review use the `ReviewService.createReview` command. It accepts a number of arguments that define the creator (`creator`), review title (`title`), deadline date and time (`deadline`), specify who can access the review (`accessPolicy`) and other parameters. All parameters are optional.

Deadline date and time should be specified in a "YYYY-MM-DDTHH:mm:ssZ" format described in JSON Syntax and Data Formats 946. The `accessPolicy` argument may have one of the following constants: `ANYONE, GROUP, PARTICIPANTS, GROUP_AND_PARTICIPANTS, GROUP_OR_PARTICIPANTS`.

The request below creates a new review:

```
[      {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
{"command" : "ReviewService.createReview",
       "args" :{
 "creator" : "jsmith",
 "title" : "Check JDK version",
 "deadline" : "2015-02-01T09:00:00Z",
 "accessPolicy" : "PARTICIPANTS",
 "customFields": [
```

```
      {"name":"Overview", "value":["Please check the JDK version
that we use."]},
      {"name":"Incident ID", "value":["1321654"]}
  ]
 }
}]
```

On success the command returns an identifier of a newly created review.

Response example:

```
[ { "result" : { } },
 { "result" : { "reviewId" : 10463 } }
]
```

**NOTE:** A newly created review is not yet ready for working on it. To processed, we need to assign review participants 955 and add review materials 957.

## Add/change participants of a review

Web service API offers two commands for assigning participants of the review: `ReviewService.setAssignments` and `ReviewService.updateAssignments`. The `setAssignments` command clears the list of existing review participants (if any) and assigns participants and roles anew. The `updateAssignments` command adds new participants and changes the roles of existing participants.

Both commands have the same set of arguments:

`reviewId` - the ID of the review whose participants we want to add or modify.

`assignments` - an array that defines review participants and their roles.

Each element of the array must be either a `{"user", "role"}` object or a `{"poolGuid", "role"}` object. In the first case the object will denote an individual participant and his role, while in the second it will denote a group of users (review pool 969) that can partake in a review and the role for this group.

The `"role"` argument can be one of the following constants: `AUTHOR, MODERATOR, OBSERVER, READER, REVIEWER, TESTER`.

```
[      {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
 {"command" : "ReviewService.updateAssignments",
```

```
      "args":{"reviewId":"10463",
      "assignments": [
            {"user":"jsmith", "role":"AUTHOR"},
            {"user":"mike", "role":"REVIEWER"},
            {"poolGuid":"1234567890", "role":"OBSERVER"}
      ]}
 }
]
```

## Assign a review pool

To assign a review pool group as the participant of the review use the `ReviewService.assignReviewPool` command. It has similar set of arguments as assigning individual participants:

`reviewId` - the ID of the review whose participants we want to add.

`assignments` - an array that defines review participants and their roles.

Each element of the array must be a `{"poolGuid", "role"}` object, denoting a group of users (review pool[969]) that can partake in a review and the role for this group.

The `"role"` argument can be one of the following constants: `AUTHOR, MODERATOR, OBSERVER, READER, REVIEWER, TESTER`.

```
[     {"command" : "SessionService.authenticate",
      "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
 {"command" : "ReviewService.assignReviewPool",
      "args":{"reviewId":"10396",
      "assignments": [
            {"poolGuid":"1234567890", "role":"REVIEWER"}
      ]}
 }
]
```

#### 9.3.6.2 Upload Review Materials

To upload review materials use the the `ReviewService.addFiles` command. This command is rather complex and allows different variants of adding materials to a review. The files may reside on your local computer or be stored in a source code management system. Moreover, the command can add files to an existing review or to create a new review.

You will describe the following use-cases:

1. Adding local files through a /content upload servlet 958
2. Adding local files with a multipart HTTP requests 960
3. Adding files from source-control system 962

Depending on the particular use-case, the syntax of the `addFiles` command and the number of its arguments will vary. For example, to add materials to an existing review you need to specify the "`reviewID`" argument. If this argument is omitted, the `addFiles` command will create a new review with the attached files.

On success the `addFiles` command returns an ID of the review where the files were added.

## Add files to a review

To add files that are stored on a local computer you need to archive them beforehand. The archived files must follow special conventions described below.

Once the archive is created you can send it to a Collaborator server and attach files to a review. There are two ways to do this:

- Use the /contentupload servlet to upload the archive and the subsequent JSON request to attach files.
- Send a single multipart request containing both "json" commands and the archive.

### Preparing files for upload

Prior to uploading files to the Collaborator server you need to prepare them as follows:

1. Copy the needed files to a temporary folder.
2. Rename each file to its MD5 checksum. In lower-case letters, without any extension. Suppose that we want to upload a file named "listobject.h" having an MD5 checksum of 198575c00a884ae27968e2fcf7d0a26d. Then the file must be renamed to "198575c00a884ae27968e2fcf7d0a26d".
3. Pack the files into a Zip archive. The name of the resulting Zip archive is not important.

## Uploading files via /contentupload servlet

Every Collaborator server has a servlet for uploading data to reviews. It resides at the following URL: `yourServer.com/contentupload`

The servlet accepts multipart HTTP requests that contain Zip archives with files. The file upload request must conform to RFC 1867: Form-based File Upload in HTML.

The servlet requires authentication parameters in the request before processing file uploads. Authentication can be provided with the following methods:

- Header: Authorization header that conforms to the Basic Authentication Scheme from RFC 2617.

```
Authorization: Basic bXl1c2VyOm15cGFzcw==
```

- Header cookies: CodeCollaboratorLogin and CodeCollaboratorTicketId. (To obtain a valid `CodeCollaboratorTicketId` you need to login via web-interface.)

```
Cookie: CodeCollaboratorLogin=jsmith;
CodeCollaboratorTicketId=0123456789abcdef0123456789;
```

If the upload is successful, the servlet will respond with a 200 (OK) status.

Below is an example of multipart HTTP request to the /contentupload servlet:

```
POST http://yourServer.com/contentupload HTTP/1.1
Content-Type: multipart/form-data; boundary=--------------------------
acebdf13572468
User-Agent: JSON API
Host: yourServer.com/
Cookie: CodeCollaboratorLogin=jsmith;
CodeCollaboratorTicketId=0123456789abcdef0123456789;
Content-Length: 1394

--------------------------acebdf13572468
```

```
Content-Disposition: form-data; name="file"; filename="localFiles.
zip"
Content-Type: application/x-zip-compressed

<Binary data of the localFiles.zip file>
---------------------------acebdf13572468--
```

Once the archive is uploaded successfully, the server unpacks it. Later on the you need to send a command to add an already uploaded files to reviews. To add files that were uploaded beforehand, use the following syntax of the `addFiles` command:

```
{"command" : "ReviewService.addFiles",
 "args" : {
 "reviewId": "<Id>", OPTIONAL
 "changelists" : [{ REQUIRED
      "commitInfo" : {<commitInfo>}, OPTIONAL
      "versions" : [ { REQUIRED
            "md5" : "<md5Checksum>", REQUIRED
            "localPath" : "<LocalPathToFile>", OPTIONAL
            "scmPath" : "<PathToFileInSCM>", OPTIONAL
            "action" : "<ActionType>", OPTIONAL
            "source" : "<SourceType>", OPTIONAL
            "baseVersion" : OPTIONAL
            }, {<version2>},..., {<versionN>}]
      }, {<changelist2>},..., {<changelistN>}]
 }
```

When adding already uploaded files, we should specify their MD5 checksums (`md5`) and either their local paths (`localPath`) or their paths in SCM (`scmPath`). All other arguments are optional. Notice that archive name is not specified, as the archive has already been unpacked.

The example below, attaches an uploaded file to the specified review:

```
[{"command" : "SessionService.authenticate",
      "args" : {"login" : "jsmith","ticket" :
"0123456789abcdef0123456789abcdef"}},
```

```
{"command" : "ReviewService.addFiles",
"args" : {
"reviewId" : "10463",
"changelists" : [{
       "versions" : [ { "md5" : "198575c00a884ae27968e2fcf7d0a26d",
       "localPath" : "c:\\work\\collab\\files\\listobject.h",
       "source" : "LOCAL" } ]
       }]
}
}]
```

### Uploading files with a multipart request

In this approach both the JSON commands and one or more archives to be attached are sent in the same multipart request.

The JSON request must be sent in a string part named "json" or as part of the query string as "json" parameter. The attached files must be sent in parts whose names coincides with the archive names. See RFC 1341: The Multipart Content-Type for detailed description of multipart requests.

To add files that are sent within the same HTTP request, use the following syntax of the `addFiles` command:

```
{"command" : "ReviewService.addFiles",
 "args" : {
 "reviewId": "<Id>", OPTIONAL
 "zipName" : "<zipNameForCommand>", OPTIONAL
 "changelists" : [{ REQUIRED
      "commitInfo" : {<commitInfo>}, OPTIONAL
      "versions" : [ { REQUIRED
            "md5" : "<md5Checksum>", REQUIRED
            "zipName" : "<zipNameForChangelist>", OPTIONAL
            "localPath" : "<LocalPathToFile>", OPTIONAL
            "scmPath" : "<PathToFileInSCM>", OPTIONAL
            "action" : "<ActionType>", OPTIONAL
            "source" : "<SourceType>", OPTIONAL
```

```
            "baseVersion" : OPTIONAL
            }, {<version2>},..., {<versionN>}]
      }, {<changelist2>},..., {<changelistN>}]
 }
```

When adding files with a multipart request, we should specify the names of archive (`zipName`), MD5 checksums of individual files (`md5`) and either their local paths (`localPath`) or their paths in SCM (`scmPath`). The archive name (`zipName`) can be given at the command level or per changelist. The `zipName` arguments are interchangeable, however at least one of them must be specified. All other arguments are optional.

As mentioned above, the attached files must be sent in parts whose names matches the values of `zipName` arguments of the `AddFiles` command:

```
--boundary separator
Content-Disposition: form-data; name="<zipName>";
filename="<zipName>"
Content-Type: application/x-zip-compressed

<Binary data of the archive file>
```

The overall example of a multipart request may look like this:

```
POST http://yourServer.com/services/json/v1 HTTP/1.1
Content-Type: multipart/form-data; boundary=--------------------------
acebdf13572468
User-Agent: JSON API
Host: yourServer.com
Content-Length: 2041

--------------------------acebdf13572468
Content-Disposition: form-data; name="json"
Content-Type: application/json;charset=utf-8

[{"command" : "SessionService.authenticate",
```

```
      "args" : {"login" : "jsmith","ticket" :
"0123456789abcdef0123456789abcdef"}},
 {"command" : "ReviewService.addFiles",
 "args" : {
 "reviewId" : "10463",
 "changelists" : [{
 "zipName" : "localFiles.zip",
 "versions" : [ { "md5" : "198575c00a884ae27968e2fcf7d0a26d",
 "localPath" : "c:\\work\\collab\\files\\listobject.h",
 "source" : "LOCAL" } ]
 }]
 }
}]
---------------------------acebdf13572468
Content-Disposition: form-data; name="localFiles.zip"; filename="
localFiles.zip"
Content-Type: application/x-zip-compressed

<Binary data of the localFiles.zip file>
---------------------------acebdf13572468--
```

## Add files from source-control system

To add files from source-control system, use the following syntax of the `addFiles` command:

```
 {"command" : "ReviewService.addFiles",
  "args" : {
  "reviewId": "<Id>", OPTIONAL
  "changelists" : [{ REQUIRED
       "commitInfo" : {<commitInfo>}, OPTIONAL
       "changelistLog" : ["<changelistID1>", ..., "<changelistIDN>"],
OPTIONAL
```

```
      "scmConnectionParameters" : ["<ConnectionParam1>", ...,
"<ConnectionParamN>"], OPTIONAL
      "scmToken" : "<scmToken>", OPTIONAL
      "versions" : [ { REQUIRED
           "md5" : "<md5Checksum>", REQUIRED
           "scmVersionName" : "<FileVersionNameInSCM>", OPTIONAL
           "scmPath" : "<PathToFileInSCM>", OPTIONAL
           "action" : "<ActionType>", OPTIONAL
           "source" : "<SourceType>", OPTIONAL
           "baseVersion" : OPTIONAL
           }, {<version2>},..., {<versionN>}]
      }, {<changelist2>},..., {<changelistN>}]
  }
```

When adding files from source-control system, we should specify their MD5 checksums (`md5`) and their paths in SCM (`scmPath`), and connection parameters to SCM ( `scmConnectionParameters).` All other arguments are optional.

The example below attaches files from Git commit with ID `85994af65fb900f6e55606cb5ca498cc54463dbc` to review 12:

```
[
    {
        "command": "SessionService.authenticate",
        "args": {
            "login": "jsmith",
            "ticket": "0123456789abcdef0123456789abcdef"
        }
    },
    {
        "command": "ReviewService.addFiles",
        "args": {
            "changelists": [
                {
                    "versions": [
```

```
                        {
                            "baseVersion": {
                                "source": "CHECKEDIN",
                                "scmVersionName":
"40bf3c9346272c4e9eb77e8e379732c922c4f93a",
                                "md5":
"f5212826f77081a8f7cfb9c0fed8967b",
                                "scmPath": "foobar/src/main/
resources/js/view.js",
                                "action": "MODIFIED",
                                "commitInfo": {
                                    "comment": "Merge pull request
#5",
                                    "date": "2020-04-09T18: 57: 51
+03: 00",
                                    "author": "GitHub",
                                    "local": false,
                                    "hostGuid":
"266a5c8637da6b0c568803161fcb54a3",
                                    "scmId":
"28a221d213b28ad1e38e4805b45b13f995bc5d94"
                                }
                            },
                            "source": "CHECKEDIN",
                            "scmVersionName":
"18d9ace010a575468b33b231040d4776196606d8",
                            "md5":
"cb75d9547200ccc50c4d4142c356d9b3",
                            "scmPath": "foobar/src/main/resources/js/
view.js",
                            "action": "MODIFIED"
```

```
                        }
                    ],
                    "changelistLog": [
                        "85994af65fb900f6e55606cb5ca498cc54463dbc",
                        "28a221d213b28ad1e38e4805b45b13f995bc5d94"
                    ],
                    "scmConnectionParameters": [
                        "https://github.com/acme/foobar.git"
                    ],
                    "scmToken": "GIT",
                    "commitInfo": {
                        "comment": "Merge pull request #6",
                        "date": "2020-05-05T12:22:34+03:00",
                        "author": "GitHub",
                        "local": false,
                        "hostGuid":
"266a5c8637da6b0c568803161fcb54a3",
                        "scmId":
"85994af65fb900f6e55606cb5ca498cc54463dbc"
                    }
                }
            ],
            "reviewId": 12
        }
    }
]
```

#### 9.3.6.3 Manage Users and User Groups

To perform administrative operations over the users of Collaborator server use the `UserService` interface. It has methods for creating, editing and deleting users and user subscriptions.

To perform administrative operations over user groups use the GroupService interface. It has methods for creating, editing and deleting groups, adding subgroups, add users to groups, excluding users from groups and so forth.

To call any of the UserService or GroupService commands the current user (that is a user who requests to run the command) **must have administrator privileges** on the Collaborator server.

## Create new user

To create new user, we will call the UserService.create command. It has two obligatory parameters: login and password. Additionally you can enter the users full name (fullName), phone number (phone), email (email) and specify whether it will be an active user (enabled) and whether the user will have administrative privileges (admin).

```
[      {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
 {"command" : "UserService.create", "args" : {
       "login" : "alice",
       "password" : "alice",
       "fullName" : "alice alice",
       "phone" : "345345345",
       "email" : "alice@alice.com",
       "enabled" : "true",
       "admin" : "false"}
 }
]
```

## Add user to a group

To add a user to some existing group, we will call the GroupService.addUser command. The command has two parameters: a login of a user to be added (memberLogin) and group identifier 230 (guid).

```
[      {"command" : "SessionService.authenticate",
       "args":
{"login":"jsmith","ticket":"0123456789abcdef0123456789abcdef"}},
 { "command" : "GroupService.addUser",
```

```
      "args" : {
      "memberLogin" : "alice",
      "guid" : "549ce60e-ea35-46fb-9e39-54529a049abf" }
 }
]
```

# 10 Techniques & Best Practices

At SmartBear, we are experts in all kinds of peer review.

In this part of the manual, we cover techniques and best practices in code review and in Collaborator.

Topics covered include:

- Invite A Colleague 968
  Describes how to invite a colleague who currently do not have a Collaborator user account.
- Review Pools 969
  Explains how to assign a group of users as participant of the review.
- Multiple Change Changelists 970
  Describes the specifics of working with changelists that accumulate changes to file versions over time.
- Optimal Review Size 971
  Describes which review sizes are optimal for peer review process.
- Metrics: Definitions 972
  Which metrics should you collect during reviews? Which metrics are collected automatically by Collaborator? How are they calculated and what exactly do they mean?
- Metrics: Analysis 974
  What do you do with raw metrics numbers? How do you collect those which are not collected automatically? What can metrics really tell us? Where might they lead us astray?
- Tips and Tricks 976
  Are there any short-cuts to help me get the results I want?
- Improving Performance 977
  Describes several ways to improve the speed of Collaborator.

# 10.1 Invite A Colleague

## Overview

A person currently cannot be added as a participant in a Review until they have a Collaborator user account. This is created the first time they login (even with LDAP integration enabled). The "Invite a colleague" feature makes it possible to invite someone to a Review even if they do not currently have a Collaborator user account, though they will have to create one.

## Admin Setting

A setting under Admin -> General -> Access Restrictions[188] allows administrators to specify a regular expression that must match against email addresses being invited to Review. The regular expression must match the entire email address, by default it is ".*" which should match all email addresses entered. The administrator can narrow the available addresses by modifying the regular expression.

For example: ".*@mycompany.com" or ".*@mycompany.com|.*@contractor.com". If an administrator want to disable this feature entirely they can simply enter "This feature is disabled"., which will not match any email address.

## Using the Feature

When the feature is enabled, each comment added in the Web UI (in any phase other than the Planning phase) will be evaluated to see if it includes e-mail addresses that matches the regular expression. If so, the user is presented with a dialog stating:

"Your comment mentioned an email address. Would you like to invite [list of email addresses in comment] to this Review?"

If the user selects "Yes", their default e-mail client will be opened with a standard invitation that they can edit and send.

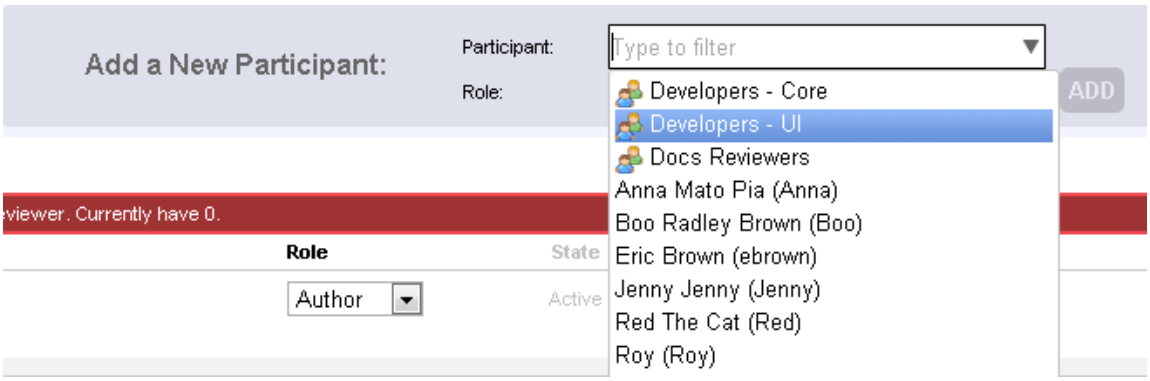
If the user selects "No", the email address is ignored.

**NOTE**: In addition to the regular expression mechanism for configuring and disabling this feature, the user may be presented with different dialogues depending on other access control specifications. For example, if participant-based access restrictions are in place, the user will be notified that the invited users will not be able to access this review. If group-based access restrictions are in place, the system cannot know whether the invited individual will be able to access the review, or not. In this case, the user is notified that the invited person may not be able to access the review.

# 10.2 Review Pools

## Using Review Pools

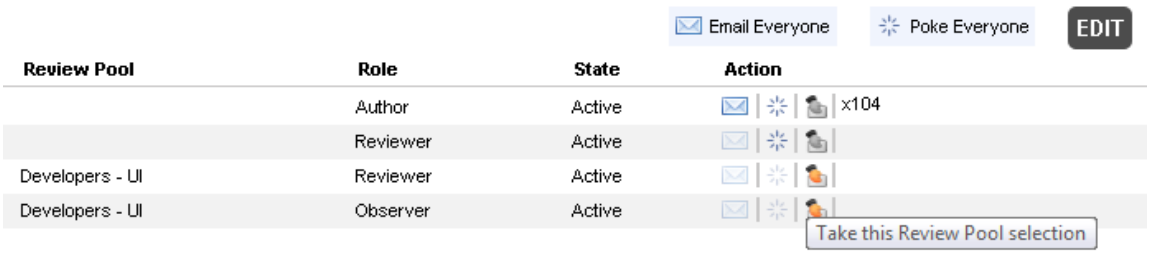
Review Pools allow an Author to select a Group as a participant, for a given Role, in a review. Groups which have been configured for use as Review Pools[245] will be displayed in the New Participant dropdown on the Review Summary page.




Any number of Review Pool selections can be made, each designating a Role of the participants from each group. Selections can be made only during the Planning and Annotating phases. When Planning is complete, the Author moves the review to the Annotating phase to notify Review Pool participants there is a Review for which they may designate themselves as the participant. All members of the selected Review Pool group will receive notifications that they are invited to participant. Email notifications are sent, if configured. Note that Begin Review will also start the Annotating phase if there are pending Review Pool assignments.

## Taking a Review Pool Selection

For each Review Pool assignment, the Participants section of the Review Summary page displays a Take button, which is enabled for eligible participants (members of the Review Pool group). The first eligible participant to go the the Review Summary page and select the Take button becomes a participant with the given Role in the Review.



The Review remains in Annotating until all pending Review Pool assignments are taken - the Inspection is not allowed to begin until then.

A review in the Annotating phase with pending Review Pool assignments will automatically begin the Inspection phase once the last review pool selection is made, as long as other participant constraints have been met.

### Configuring For Review Pools

In order to enable Review Pools, a Collaborator Administrator will need to configure Groups to be used for Review Pools 245.

### Review Pool Subscriptions

Administrators can also set-up review pool subscriptions. Once a review of particular author is created, or a review contains some particular files, or a review uses some particular template, Collaborator will automatically add a group to this review. At that the number of review participants to be added and their roles are fully configurable 232 for each group.

## 10.3 Multiple Change Changelists

A changelist is a generic SCM concept representing a set of changes in version control. Changelists for some SCM systems like Perforce and Subversion are atomic entities, representing a single unit of changes that occur at the same time. Other SCM systems have changelists that accumulate changes to file versions over time, allowing multiple changes (versions) of any given file within the same changelist. Some examples of this latter type of changelist are ClearCase Activities and PTC Change Packages.

For these multiple-version changelists, the changelist represents an accumulation of changes to each of the source files in it. In most cases, in the context of a review or in thinking about what the changelist represents, users are interested in the difference represented by the accumulation of changes in the changelist, that is, for each source file the difference between the latest version occurring in the changelist and the version content that existed before the first change in the changelist. Collaborator calculates differences based on this, when a multiple-version changelist is uploaded to a review.

This can be confusing in some circumstances. If an added file is part of the changelist, and there are subsequent changes to that file in the changelist, then uploading this changelist for review will result in the latest change to that file appearing as an add in the changelist. In other words, it will have no predecessor. In an SCM system where added files are always version 1.1, a review of a changelist having version 1.5 of a file with no predecessor is incongruent with the versioning of the SCM system. Yet this is exactly what the accumulation of changes to that file in the changelist represent - a sum total of changes that did not exist before the changelist.

While it might be possible to find and upload all versions of each file that occur in a multiple-version changelist and make them available for comparison in Collaborator, in our experience this makes for an unnecessarily complicated review - this is less of a peer review feature and more of a version history browser feature. If a review is to be conducted on successive revisions to a file in this way, the review should be conducted at each iteration of the changelist. Uploading the same changelist to the same review as each successive set of changes is made as part of the rework step of the review will result in all of the versions of each file being available in the it, and each version being available for inspection by the other review participants.

## 10.4 Optimal Review Size

We sometimes get asked by customers how large their reviews should be. It is a difficult question to answer because it depends a lot on the review culture of the team and the nature of the content being reviewed. Opinions vary widely on what the optimal size should be.

Creating many tiny reviews is not optimal because of there is some static overhead for each review (setting up participants and metadata, and so forth). There is also a mental penalty for "context switching" when a reviewer changes their attention between two different reviews.

Creating a few giant reviews is not optimal because there is a limited amount of time a reviewer can truly concentrate on a review. Our studies have shown the maximum time to be about 90 minutes. If the review is too large to process in 90 minutes then the review is less effective - defects will be missed and time will be wasted as the reviewer's attention wanders.

Based on our research we recommend the following Optimal Size for reviews, with the understanding that one size does not fit all:

- 3 participants
- 15 files, 3 versions each (3 changelists)
- 2 overall defects, 5 file defects
- 30 file conversations
- 8 comments per conversation

Collaborator is designed to give the best experience at this size. However we do try to support a wide range of sizes so that customers can do what works best for them. Note though that reviews that are more than an order of magnitude larger than this Optimal Size will start to perform sluggishly.

### See also

- Best practices for peer code review - a web site article that describes a successful peer review strategy.

# 10.5 Metrics: Definitions

Collaborator collects a variety of raw metrics automatically. This section defines these metrics; a later section[974] discusses what these metrics can tell us.

## Lines of Code

The most obvious raw metric is "number of lines of source code". This is "lines" in a text-file context. Often this is abbreviated "LOC".

Collaborator does not distinguish between different kinds of lines. For example, it does not separately track source lines versus comment lines versus whitespace lines.

For code review metrics, often you usually want to use general lines of code and not break it down by type. Often the code comments are just as much a part of the review as the code itself -- check for consistency and ensuring that other developers will be able to understand what is happening and why.

The lines of code metrics (LOC metrics) are calculated only for source code and other text-based files. For other types of review materials (Word, Excel, PDF or Image files) the metrics are not calculated and return 0.

The LOC metrics displayed on the Review Summary[362] page include added lines (), changed lines () and removed lines (). If the Overlay view is selected (default), the LOC metrics are calculated comparing the latest uploaded revision of file against the baseline revision of that file. Here, the baseline revision stands for the revision at the moment the review was created. If the Separate view is selected, the LOC metrics are calculated comparing each individual file revision against its previous revision.

For reviews created by pull requests[826], file changes made by merge commits (if any) are only displayed in Separate view and they are not taken into account when calculating overall LOC metrics. Besides, file changes made by merge commits do not affect the overall rework count of a file.

The Customizable Review Reports[468] may provide you with more line-related metrics. Additionally to added, changed and removed lines, they can display total number of all lines of all uploaded files (LOC Uploaded), number of reworked lines (sum of added, changed and removed lines) (LOC Reworked) and the difference between number of added and removed lines (LOC Delta).

Keep in mind, that "Ignore Whitespace", "Ignore Sequence Number" and other Diff Viewer settings[372] do not affect on how line metrics are calculated. They only affect on how line differences are displayed.

## Time in Review

How much time (person-hours) did each person spend doing the review? Collaborator computes this automatically. This raw metric is useful in several other contexts, usually when compared to the amount of file content reviewed.

Developers (rightly) hate using stopwatches to track their activity, but how can Collaborator -- a web server -- automatically compute this number properly?

Our technique for accurately computing person-hours came from an empirical study we did at a mid-sized customer site. The goal was to create a heuristic for predicting on-task person-hours from detailed web logs alone.

We gave all review authors and reviewers physical stop-watches and had them carefully time their use of the tool. Start the stopwatch when they began a review, pause if they break for any reason -- email, bathroom, instant messenger. The times were recorded with each review and brought together in a spreadsheet.

At the same time, we collected detailed logs of web server activity. Who accessed which pages, when, and so forth. Log data could easily be correlated with reviews and people so we could "line up" this amalgamation of server data with the empirical stopwatch times.

Then we sat down to see if we could make a heuristic. We determined two interesting things:

First, a formula did appear. It goes along these lines: If a person hits a web page, then 7 seconds later hits another page, it is clear that the person was on-task on the review for the whole 7 seconds. If a person hits a web page, then 4 hours later hits another page, it is clear that the person was not doing the review for the vast majority of that time. By playing with various threshold values for timings, we created a formula that worked very well -- error on the order of 15%. Technically, Collaborator server queries for review activity every 15 seconds and updates the time counter if any of these requests were successful during specified time interval (60 seconds, by default). Also if user does not perform any  actions after 5 minutes, time tracking is stopped.

Second, it turns out that humans are awful at collecting timing metrics. The stopwatch numbers were all over the map. People constantly forgot to start them and to stop them. Then they would make up numbers that "felt right," but it was clear upon close inspection that their guesses were wrong. Some people intentionally submitted different numbers, thinking this would make them look good (that is, "Look how fast I am at reviewing!").

So the bottom line is: Our automated technique is not only accurate, it is more accurate than actually having reviewers use stopwatches. The intrinsic error of the prediction heuristic is less than the error humans introduce when asked to do this themselves.

## Total Person-Time

The total of all recorded time that all the users were looking at review (includes time spent in annotation, planing, inspection and rework phases). **Total Person-Time** is an aggregate value for all users taking part in a review, while **Time in Review** is counted for each separate user.

**Reviewer Time** and **Author Time** are subsets of Total Person-Time, limited to the time that was spent in the reviewer and author roles, respectively.

### Defect Count

How many defects did we find during this review? Because reviewers explicitly create defects during reviews, it is easy for the server to maintain a count of how many defects were found.

Furthermore, the system administrator can establish any number of custom fields 256 for each defect, usually in the form of a drop-down list. This can be used to subdivide defects by severity, type, phase-injected, and so on.

### File Count

How many files did we review? Usually the LOC metric 972 is a better measure of "how much did we review," but sometimes having both LOC and number of files is helpful together.

For example, a review of 100 files, each with a one-line change, is quite different from a review of one file with 100 lines changed. In the former case, this might be a relatively simple refactoring; with tool support, this might require only a brief scan by a human. In the latter case, several methods might have been added or rewritten; this would require much more attention from a reviewer.

### Wall-Clock Time, Review Wall-Clock Duration

How much time has passed since the review was created and till the review was completed (or now, if the review is still in progress). This is a useful metric if you want to make sure all reviews are completed in a timely manner.

## 10.6 Metrics: Analysis

It is fine to collect metrics, but what do they tell us? It is tempting to apply them in many different contexts, but when are metrics telling us something and when are we reading too much into the numbers?

### Defect Density

Defect Density is computed by: ( number of defects ) / ( 1000 lines of code ).

This is the number of defects found, normalized to a unit amount of code. 1000 lines of code, or "kLOC" is often used as a standard base measure. The higher the defect density, the more defects you are uncovering.

It is impossible to give an "expected" value for defect density. Mature, stable code might have defect densities as low as 5 defects/kLOC; new code written by junior developers may have 100-200.

What can defect density tell us?

Let's make an experiment. We take some reviewers and have them inspect many different source files. Source files vary in size from 50 lines to 2000 lines. Reviewers inspect about 200 lines at a time so as not to get tired. We will record the number of defects found for each file.

What would we expect to find? First, longer files ought to have more defects than shorter ones, simply because there is more code. More code means more that could go wrong. Second, some files should contain more defects than others because they are "risky" -- maybe because they are complex, or because their routines are difficult to unit-test, or because their routines are reused by most of the system and therefore must be very accurately specified and implemented.

If we measure defect density here, we handle the first effect by normalizing "number of defects" to the amount of code under review, so now we can sensibly compare small and large files. So the remaining variation in defect density might have a lot to do with the file's "risk" in the system. This is, in fact, the effect we find from experiments in the field.

So defect density can, among other things, determine which files are risky, which in turn might help you plan how much code review, design work, testing, and time to allocate when modifying one of those files.

Now let's make another experiment. We will take a chunk of code with 5 known algorithm bugs and give it to various reviewers. We will see how many of the defects each review can find in 20 minutes. The more defects a reviewer finds, the more effective that reviewer was at finding the defects. This is a simple way to see how effective each reviewer is at reviewing that kind of code.

Of course in real life the nature of the code and the amount of code under review varies greatly, so you cannot just look at the number of defects found in each review -- you naturally expect more defects from a 200-line change than from a 2-line change. Defect density provides this normalization so you can compare reviewers across many reviews.

If you are comparing defect density across many reviews done by a single person, you are measuring the relative "risk" of various files and modules.

## Inspection Rate

Inspection Rate is computed by: ( Lines of Code Reviewed ) / ( Total Person-Time ).

This is a measure of how fast we review code. A sensible rate for complex code might be 100 LOC/hour; generally good reviews will be in the range of 200-500 LOC/hour. Anything 800 LOC/hour or higher indicates the reviewer has not really looked at the code -- we have found by experiment that this is too fast to actually read and critique source code.

Some managers insist that their developers try to increase their inspection rate. After all this means "review efficiency" is improving. *This is a fallacy.* In fact, the *slower* the review is, the *better* job the reviewers are doing. Careful work means taking your time.

Instead, use inspection rate to help you predict the amount of time needed to complete some code change. If you know this is roughly a "1000-line change" and your typical inspection rate is 200 LOC/hour, you can budget 5 hours for the code review step in your development.

If anything, a manager might insist on a slower inspection rate, especially on a stable branch, core module, or close to product release when everyone wants to be more careful about what changes in the code.

**Inspection Rate (Changed)** metric counts only lines of code that were changed (added, removed, or modified).

**Inspection Rate (Uploaded)** metric counts only lines of code that were uploaded in the review.

### Defect Rate

Defect Rate is computed by: ( Number of defects ) / ( Total Person-Time ).

This is the speed at which reviewers uncover defects in code. Typical values range between 5 and 20 defects/hour, possibly less for mature code, but not usually much greater.

The same caveats about encouraging faster or slower inspection rates apply also to defect rates. Read the Inspection Rate section for details.

### Metrics Applied

If we have learned one thing about metrics and code review it is: *Every group is different, but most groups are self-consistent.* This means that metrics and trends that apply to one group do not necessarily apply to another, but within a single group metrics are usually fairly consistent.

This between-group difference can be attributed to the myriad of variables that enter into software development: The background, experience, and domain knowledge of the authors and reviewers, programming languages and libraries, development patterns at different stages of a product's life-cycle, project management techniques, local culture, the number of developers on the team, whether the team members are physically together or separate, and so forth.

## 10.7 Tips and Tricks

This section will describe workarounds and tricks we use for Collaborator.

### Picking reviews through the Command-Line Client

When sending files to a review through the Command-Line Client, use "last" instead of the review ID to pick the last review or "ask" to be prompted with choices to pick.

### Custom Fields with Date and Times

Collaborator does not have date or time custom fields, but you can closely approximate a date or time by using regular expression validation of single line text fields. For dates, the following regular expression requires a date in the 20th or 21st century that is approximately valid (yes, it accepts 31 days each month):

```
(?:19|20)\d\d-(?:0[1-9]|1[012])-(?:0[1-9]|[12][0-9]|3[01])
```

The following regular expression validates a time on a 24 hour clock:

```
(?:[01][0-9]|2[0-3]):[0-5][0-9]
```

The two could be combined to accept a date and time field. Be sure you set the description of the field to describe exactly the format you are looking for so that users do not have to parse the regular expression to know what to enter.

## 10.8 Improving Performance

If your Collaborator server is not working as fast as you would like there are a couple of things you can try:

1) Use a modern browser. The latest versions of Internet Explorer, Firefox, Chrome, and Safari all render much faster.

2) Make your reviews a reasonable size. Collaborator supports a wide range of review sizes, but reviews that are larger than 10x our recommended optimal size[971] can become uncomfortably slow.

3) Make sure your database is fast. Collaborator spends much of its time accessing the database, so this needs to be as fast as possible. If the database is not located on the same physical server as Collaborator, make the network connection to the database as fast as possible.

4) Try a different database. Embedded database is convenient for small environments, but with larger databases (~100k reviews or more) you should use external database: MySQL[60], SQL Server[64] or Oracle[67].

5) Tune the server parameters. Try increasing the maximum heap size[1232] and/or tweaking the sizes of the various server caches.

# 11 Appendices

Appendix A: Known Issues & Errata[978]

Current known issues in the server and various client components, including integrations with other systems.

Appendix B: Version History[982]

Complete version history for each public release of the various components.

Appendix C: Java VM Options[1232]

Overview of Java options useful for Collaborator configuration.

# 11.1 Appendix A: Known Issues

These are the major known issues currently known for all Collaborator components.

Many of the issues have workarounds; those are given here as well.

## Get notified automatically when a new version is available!

SmartBear announces new publicly-available versions of Collaborator at our forum located at:

http://community.smartbear.com/t5/Collaborator/Collaborator-Release-Notifications/td-p/97024

## Known Issues with the Collaborator

- On Unix systems (Linux, macOS) Collaborator clients could fail to upload review materials if the system's region-language pair is not supported by Java.
- Dump for a single review could include remote system data from the interconnected reviews as well.
- The built-in PDF viewer of Windows 8, 8.1 and 10 do not support overlay layers in PDF files. Collaborator uses such layers to render coordinate comments (pushpins) 425 in PDF versions of review materials for archived reviews 165. Because of this, pushpins will not be displayed if the PDF file from the archive is opened in the built-in PDF viewer of Windows 8, 8.1 and 10. As a workaround, please install the Adobe Acrobat Reader or any other full-featured PDF viewer.
- Icon could be missing for comments or defects made in Word documents having several revisions that were staged to the same Git commit.
- Excel files with high column counts may cause severe performance degradation when viewed in the Collaborator web application (for example, for review of differences). Depending on the processor and memory resources of the client machine, this may be noticed at varying numbers of column counts. For a typical desktop, experimentally we have noticed that approximately 500 columns is where performance begins to suffer noticeably and 3000 columns is the point at which it becomes infeasible to effectively work with the Excel file in question.

- Double quote characters (") in custom field names may break Oracle reporting views. Custom field names become column headers in the views, and Oracle does not allow double quotes in column names. Because of this, Collaborator removes double quote characters from the names of custom fields when it creates reporting views for Oracle databases.
  If some custom field names differ only by double quotes this would result in an "ORA-00957: duplicate column name" error in server logs. To resolve the issue, you may either remove one of duplicate column names from the reporting view, or rename the custom fields to avoid coincidence.

- Loading large dump files to Oracle may take up to a full day to complete. This appears to be an issue with Oracle driver and is under investigation.

- On Oracle databases, Collaborator does not search the contents of custom fields by default (since this significantly reduces search performance). Instead, the search results page display additional fields that define in what areas to perform new search. In this panel you can enable searching in custom fields.

- When uploading .doc and .docx files, the 'Track Changes' feature is not supported. If you do not want to see all of the content from all versions, you will need to accept all of the changes in the document before adding it to the review.

- If the installer cannot find a JRE on your system, it will prompt you for the location of a suitable JRE. On systems with multiple JREs installed, it may be necessary to specify to the installer which JRE should be used for Collaborator. On Windows platforms, running the installer with the `-manual` argument will suppress the JRE search and cause the installer to prompt for the JRE location (specifically, java.exe). On *nix platforms, you can specify the JRE location by setting the `INSTALL4J_JAVA_HOME_OVERRIDE` environment variable to the `JAVA_HOME` value.

- In Collaborator 8 and earlier, changes uploaded via the "Add Diffs" command were listed in review materials under the “Changes from Uploaded Files” group. Whereas in Collaborator 10 and later these changes are listed under the "Changes from <SCM_Name>: diffs" group.

## Known Issues with the Server Component 56

- When the server is running on Linux systems, there can be issues with unavailable fonts or available fonts with sufficiently different font metrics that can cause the reporting engine to not render content. Workaround: Install the Microsoft True Type fonts (msttcorefonts on Debian-based systems).

- Collaborator server 12.0 and later does not install on Windows Server 2008 unless Oracle JRE is installed beforehand. If no Java is pre-installed, server installer tries to install OpenJDK which does not support Windows Server 2008.

- If your charts are not being displayed, verify that <collab server install dir>/tomcat/temp exists and is writable by the user that runs the server.

## Known Issues with the Web Client 313

- Search results may not return all results without noting the truncation. This can happen if many results match the search but are inaccessible to the user.
- Home screen cannot display more the 999 open reviews.
- Custom reports with a large number active columns or filters; or with long filter text can break the bookmark, SQL, Printable, and CSV links in Internet Explorer. Because the links contain all the filter information, the URL's can exceed the IE's maximum URL length (2083 characters).
  Workaround: Use Firefox or Chrome for complex reports
- Character set differences that change the location or number of line breaks can change the way comments get promoted.
  Workaround: If necessary, use the auto-detected character set to make comments and mark conversations read.
- Character set differences that change the location or number of line breaks result in metrics that may not be perfectly correct. The metrics for files are computed using the auto-detected character set.
- If there are multiple files with the same filename (but different paths) in a review the automatically linked filenames in review text (custom fields, chat, and so forth) may not link to the intended file.
- If the column widths of spreadsheets are not sized to show the entire text or set to wrap text, the numbering on the rows in Diff Viewer may not line up perfectly.

## Known Issues with Anti-Virus Software

- Anti-virus software is known to interfere with launching sub-processes and communicating with SCM clients. If you experience any problems with a Collaborator client hanging or getting unexpected results, and you have anti-virus software running, disable the anti-virus software and try reproducing the problem. If you subsequently contact technical support, let them know that you have anti-virus software running.

## Known Issues with the Command-Line Client 506

- Control Characters in text are replaced with the Unicode Replacement Character. This is most often encountered with smart quotes pasted into changelist descriptions.

## Known Issues with the GUI Client 496

- Because of technical issue with GTK3, GUI Client on Unix/Linux operating systems may fail to display table data on Add Changelists, Add Commits, Add Transactions and similar pages.

- Git users on Windows systems may need to modify their path to allow the client to detect their SCM. The default Windows system PATH entry defaults to "...\git\cmd". Changing this to "...git\bin" should allow the client to properly detect the SCM when the client is launched from the tray notifier.

- If your PTC Integrity server uses IPv6 connection, GUI Client cannot map the server host name. To workaround the issue, you will need to add the IPv6 address of your server to the hosts file on your machine. The hosts file resides at "/etc/environment/hosts" on Unix and at "%SystemRoot%\System32\drivers\etc\hosts" on Windows.

## Known Issues with the Eclipse Plug-in 525

- Conversations View in Eclipse Oxygen does not split long line of text into multiple lines, because of this a horizontal scroll bar can appear if your conversation of checklist item contains long text.

- Collaborator Eclipse Plug-in is shipped with the Google Guava library version 20. Other Eclipse plug-ins may use another versions of this library and in some rare cases this may cause dependency conflicts.

## Known Issues with Subversion Integration 799

- Directory entries are ignored - so adding an entire directory does not show up in a review.

- Symlinks are not supported pre-commit - a Symlink uploaded pre-commit will show invalid content in a review.

## Known Issues with Team Foundation Server Integration 738

- "Unable to determine the source control server" error can occur when adding modified files. There must be a corresponding Team Foundation Server working folder for the directory configured in the GUI client, or the directory specified for the command line client to avoid this error.

- Team Foundation Server integration will not work with non-English installations of Visual Studio.NET. Regional settings for other locales are supported, but installing a non-English Visual Studio prevents correct parsing of `tf.exe` output.

## Known Issues with JIRA Integration 887

- Currently, Collaborator cannot populate values of arbitrary custom fields. If your JIRA server requires certain custom fields to be set during the creation of tickets/items, then Collaborator will be unable to create new tickets/items.

- Due to technical issues, JIRA 7 cannot display the "Review Information" block automatically. As a workaround you can manually make the Review Id, Review Participants, Review Phase and Review Link fields be visible on your JIRA screens.

## 11.2 Appendix B: Version History

### Get notified automatically when a new version is available!

SmartBear announces new publicly-available versions of Collaborator using a forum thread.

If you want to subscribe to the Release Notification thread to get all release notifications, you can do so here:

http://community.smartbear.com/t5/Collaborator/Collaborator-Release-Notifications/td-p/97024

Links to individual versions:



### 11.2.1 Version 13

**13.6.13601 -May 25, 2021**

**Improvements:**

- **Compute Simulink model diffs on the server side.** Now Collaborator calculates differences in Simulink model revisions on the server. This improves the overall performance: Collaborator computes differences only once and doesn't need to send files to every client. (COLLAB-8360)

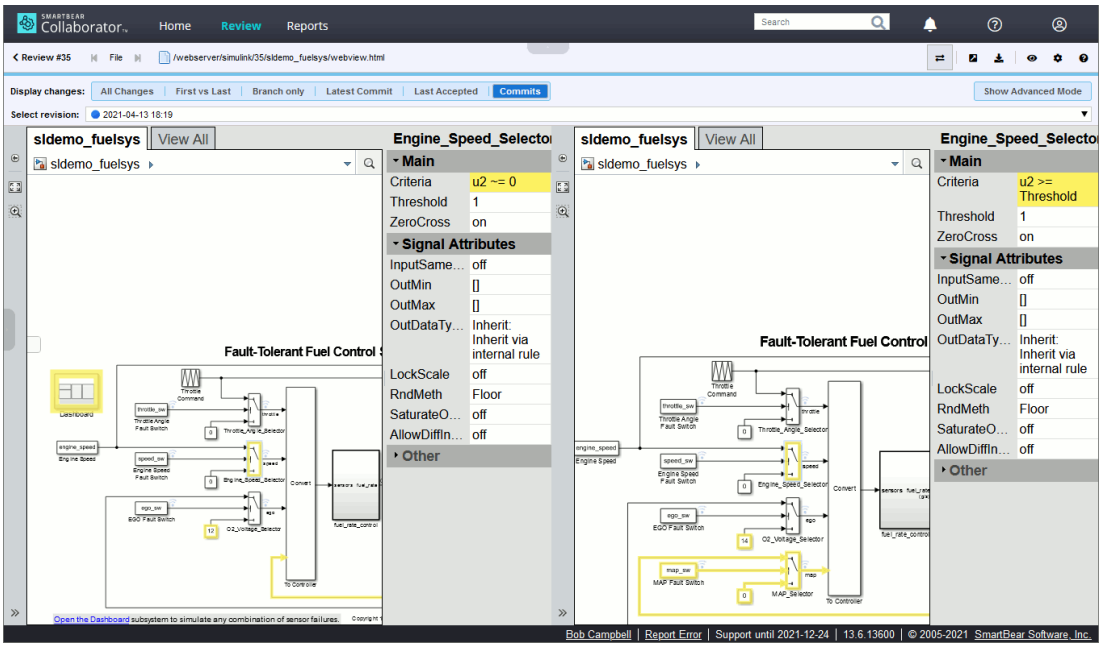
**Bug Fixes:**

- fixed - Collaborator desktop clients didn't work on some versions of MacOS Big Sur. (COLLAB-8004)
- fixed - After upgrading Collaborator Team edition to 13.6.13600, the server stopped recognizing valid license codes. (COLLAB-8441)
- fixed - An empty or corrupted copy of a PDF file uploaded to a review could cause endless page loading and block the review process. (COLLAB-8081)
- fixed - Incorrect link was used in the 'User Mentioned' email notifications. (COLLAB-8176)
- fixed - Changes were not calculated correctly for the Perforce revert mode commits. (COLLAB-8431)
- fixed - Collaborator could fail to create review from GitLab when merge request contains submodule comment. (COLLAB-8451)

**13.6.13600 - April 22, 2021**

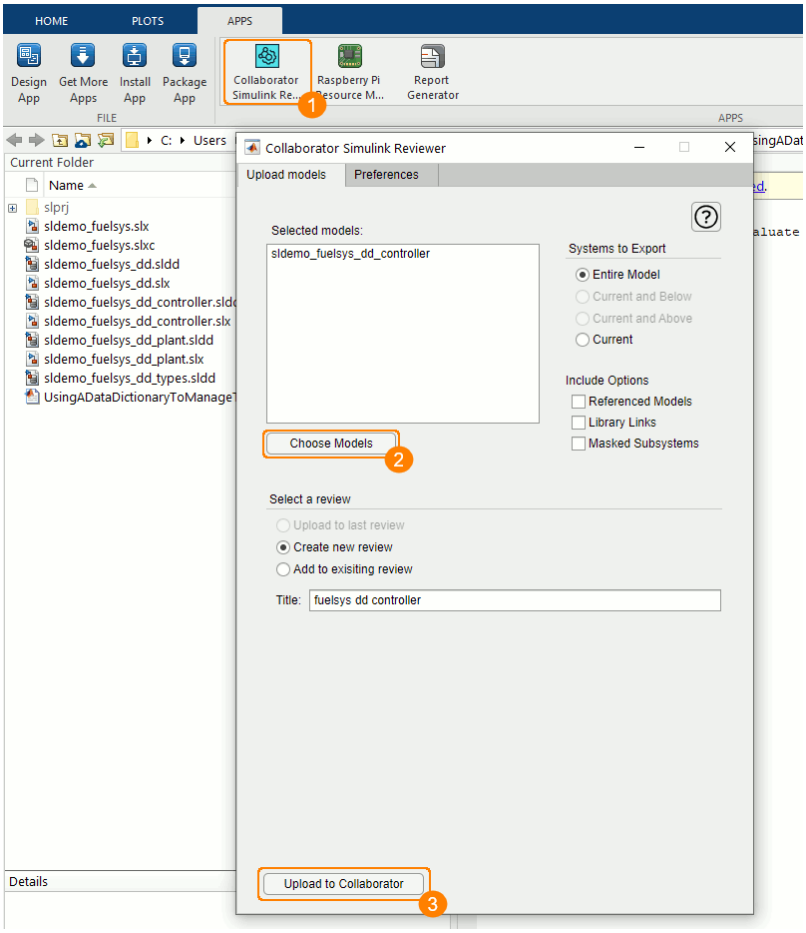
**New Features**

**Review Simulink models.** With Collaborator version 13.6 it became possible to review MathWorks Simulink models 404 that are exported as web view archives.




To use the feature, you should have Collaborator Enterprise edition and also have to obtain additional Simulink integration licenses from SmartBear and assign them to Collaborator users. (COLLAB-1489)

**Collaborator Simulink Reviewer App.** This application facilitates performing Simulink model reviews. It is installed as plug-in into MathWorks Simulink and allows uploading models to Collaborator server directly form Simulink. (COLLAB-8078)




**More improvements:**

- **Cross-page scrolling in document reviews.** In previous versions, when reviewing documents, the Diff Viewer displayed only one page per revision. Now it can show multiple pages simultaneously (for example, the end of the current page and the beginning of the next page). (COLLAB-4855)

- **Cache for document reviews.** Now Collaborator can cache previously opened documents in order to display them quickly when you reopen them. Caching is on by default and is controlled by the `com.smartbear.diff.image.cache` VM option. (COLLAB-8039)

- **History of file renames.** Now, when uploading file revisions under a different name[455], the history of file renames is listed in the review's Chat[356] section. (COLLAB-8210)


- **Database performance improvements.** We have refactored and improved our data access mechanisms to reduce number of calls to database and level up data cashing. (COLLAB-8162, COLLAB-8266, COLLAB-8267)

- Improved how Diff Viewer highlights additions and removal of entire columns when reviewing spreadsheets. Now inserting and removing of entire columns with content will be interpreted as inserting/removing a series of cells. Whereas inserting and removing of entire columns and modifying nearby cells will still be interpreted as modifying a series of cells. (COLLAB-8278)

**Bug Fixes:**

- fixed - If the Allow users to move comments server setting was enabled, comments could move themselves sporadically in some cases. (COLLAB-8323)

- fixed - Excel file revision where cells of entire row have been merged could not be displayed after upgrade to 13.5.13500. (COLLAB-8306)

- fixed - Non-participant could change list of review participants. (COLLAB-8256)

- fixed - Reviews could not be loaded when file revisions were removed from content storage. (COLLAB-8255)

- fixed - TFS Work Item integration failed to start when the server contained "tfs" in its host name. (COLLAB-8251)

- fixed - The "addgitdiffs" command showed the reference record as a standalone branch. (COLLAB-8241)

- fixed - Azure DevOps webhook was not created if the project name contained spaces. (COLLAB-8239)

- fixed - Cell comments and defects could be misaligned across revisions when content was added or removed. (COLLAB-8208)

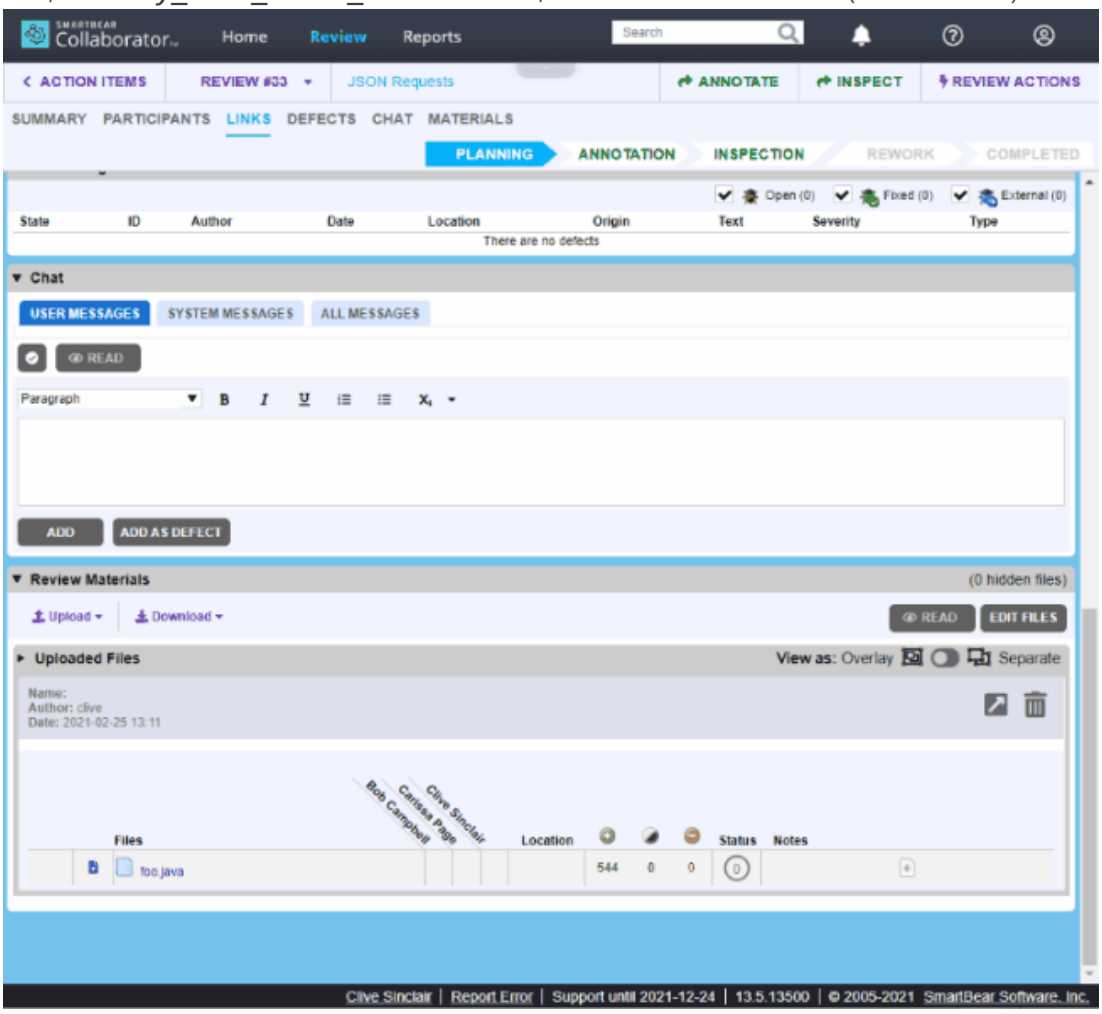
- fixed - The C# syntax highlighting was missing a few keywords (COLLAB-8179)

- fixed - The Microsoft Office plug-ins could not establish TLS/SSL connection with Collaborator server unless the TLSv1 protocol was not enabled on the server. (COLLAB-8175)
- fixed - Collaborator can be stuck on endless loading when an empty or corrupted version of PDF file was uploaded to a review. (COLLAB-8081)
- fixed - Security issue: HTML markup in the review title was not escaped. (COLLAB-7977)
- fixed - The "search" icon button in Web UI was not working. (COLLAB-7805)

### 13.5.13500 - February 26, 2021

#### New Features

**Different file names for the same document.** For manually uploaded files it became possible to upload their new revisions under a different file name 455. For example, you can upload a file named - `my_file1.doc` and then upload its revised version as `my_file2.doc`, or as `my_file_Johns_version.doc`, or whatever other name. (COLLAB-1846)



**More improvements:**

- **Support 'Mark All Read' on the review materials level.** The Review Materials 357 section of Web Client got a new button that allows to mark all comments in all review materials as read. Since this operation makes it extremely easy for someone to mark all comments as read without actually reading them, this action should be enabled by Collaborator administrators via the respective option 191. (COLLAB-7551)
- **Support merge request pipelines for GitLab.** If your GitLab repository is configured to use merge request pipelines, Collaborator integration will use them as well. (COLLAB-8154)
- Now Collaborator updates user email address, first name and last name from the SAML server, when synching user membership with a SAML SSO server 142. (COLLAB-7897)
- Updated the JavaMail library to 1.6.2 in order to use the TLSv1.2 protocol by default. (COLLAB-7891)
- When changing a review group, Collaborator will now retain existing template and filled-in checklist items if the template/checklist is available to the newly selected group. (COLLAB-7445)

**Bug Fixes:**

- fixed - Reports returned blank values for the LOC metrics. (COLLAB-8151)
- fixed - The Azure DevOps integration was trying to search for workitems only in the "DefaultCollection" collection. (COLLAB-8126)
- fixed - DiffViewer did not open a review if it contained invalid checklist item identifier. (COLLAB-8124)
- fixed - Running the `ccollab admin review-xml` command caused "Error accessing URL". (COLLAB-8109)
- fixed - The Comments panel was displayed incorrectly for the files uploaded from Subversion or linked as live URLs. (COLLAB-8106)
- fixed - An error appeared in the timeline report if the user deleted the review's author from the list of subscriptions. (COLLAB-8105)
- fixed - Remote system cache statistics could be absent in logs on certain JDK's. (COLLAB-8072)
- fixed - If the TFS workspace location was set to "Server" in the Visual Studio settings, the "Condition must not be false" error appeared when trying to add changes. (COLLAB-8060)
- fixed - The Azure DevOps integration failed to map VisualStudio.com repository if it contains the DefaultCollections portion in URL. (COLLAB-8049)
- fixed - A new incoming comment could destroy not-yet-submitted comment. (COLLAB-8028)

- fixed - An exception could occur in the Eclipse plug-in while opening reviews that were created by copying previous reviews. (COLLAB-8023)
- fixed - When entering a comment of a certain length in a conversation, the top line of text could become obscured. (COLLAB-8006)
- fixed - Merged cells caused incorrect cell identification. (COLLAB-7983)
- fixed - The top-most row could sometimes become misaligned with the rest of the rows when reviewing spreadsheets. (COLLAB-7982)
- fixed - The "Email All" feature did not launch an email client if the list of email addresses exceeded 1752 characters. (COLLAB-5455)
- fixed - Long review title could affect the logic of sending the approaching deadline notifications. (COLLAB-4208)

**13.4.13402 - January 26, 2021**

- fixed - The Azure DevOps integration was trying to search for workitems only in the "DefaultCollection" collection. (COLLAB-8126)
- fixed - DiffViewer did not open a review if it contained invalid checklist item identifier. (COLLAB-8124)
- fixed - An exception could occur in the Eclipse plug-in while opening reviews that were created by copying previous reviews. (COLLAB-8023)
- fixed - The Azure DevOps integration failed to map VisualStudio.com repository if it contained the DefaultCollections portion in URL. (COLLAB-8049)

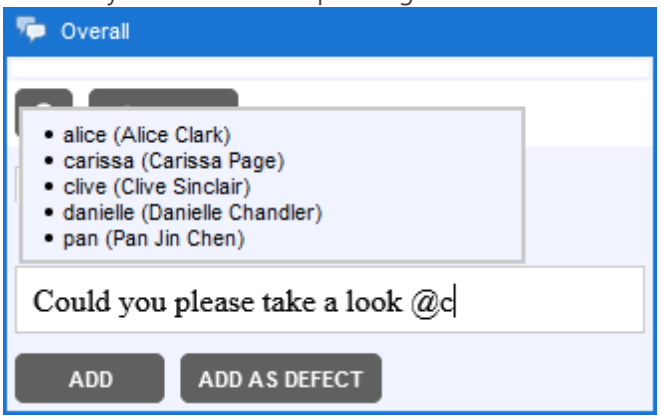
**13.4.13401 - January 15, 2021**

- fixed - The Comments panel was displayed incorrectly for the files uploaded from Subversion or linked as live URLs. (COLLAB-8106)

**13.4.13400 - December 30, 2020**

**New Features**

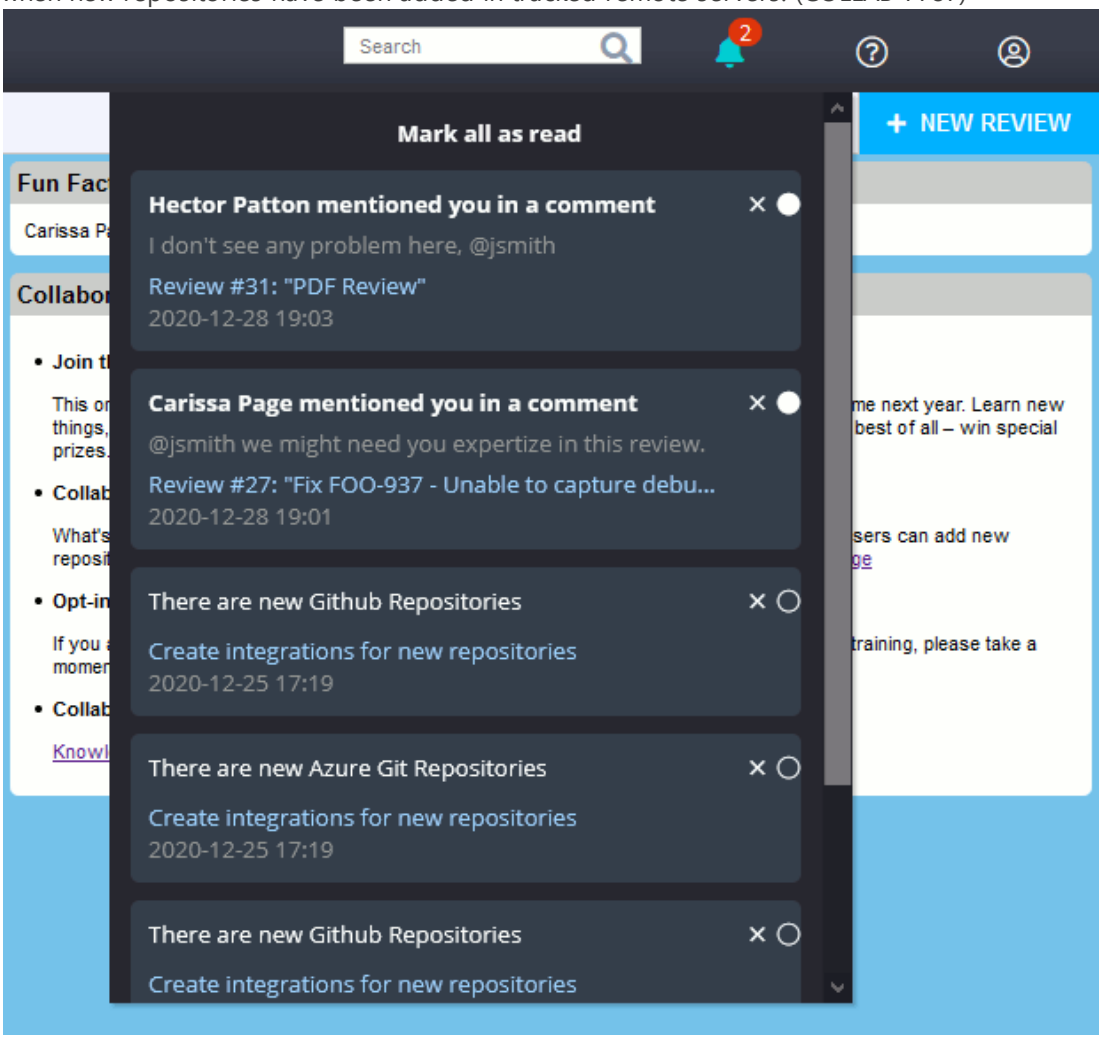
**Mention other users in comments and defects.** Now you can type the @ character followed by user login or display name to mention a user in comment or defect. When you submit your comment or defect, Collaborator sends an e-mail and in-app notifications, so that mentioned users may check the corresponding comment or defect. (COLLAB-3321)




*Typing-in a user mention*

**Auto-polling for new repositories.** Previous versions of Collaborator are able to create integrations for various existing remote server repositories 826. Yet, when new repositories were created on a remote server, Collaborator administrators were not aware of them. Now administrators may configure Collaborator server to automatically check if any new repositories have been added 218 in tracked remote servers and be notified of these new repositories. So they can integrate newly created repositories with little or no delay. (COLLAB-7719)

**In-app notifications.** The top toolbar of Web Client is now able to display some of user-related notifications. Namely, it displays when the user was mentioned in some review, or when new repositories have been added in tracked remote servers. (COLLAB-7787)

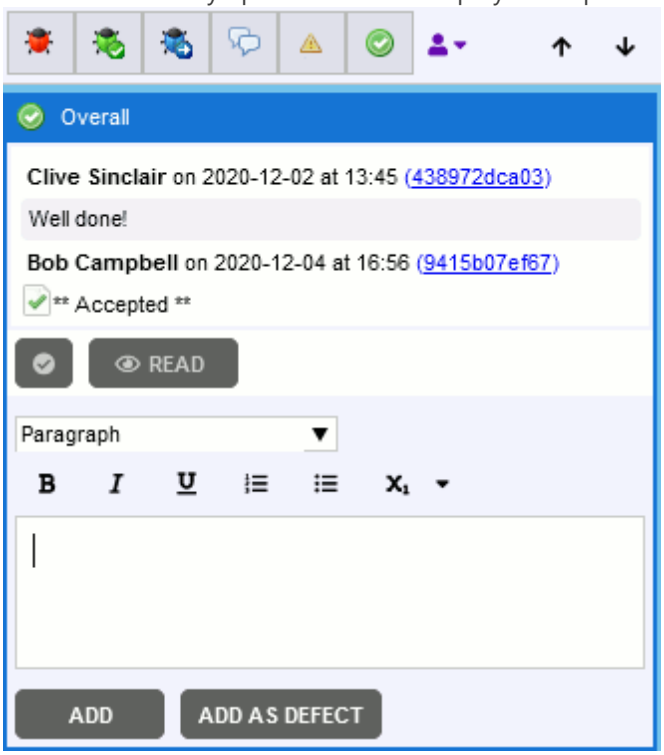



**Generating login tickets.** Now Collaborator users can generate login tickets[318] themselves. Login tickets are special alpha-numeric identifiers that act as user credentials for a limited time period. By default, tickets are valid for 30 days. Collaborator administrators can decrease or increase ticket's time-to-live[196].

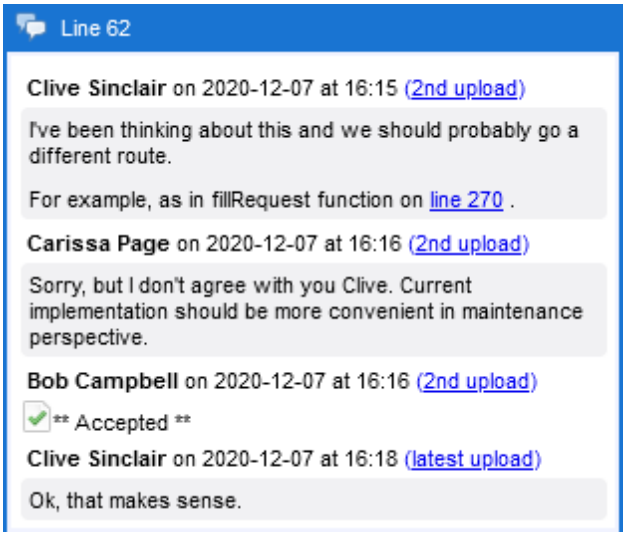
To obtain a ticket, go to the Account information tab of User Preferences, press Generate Login Ticket and copy the resulting alpha-numeric identifier. Now you can specify it instead of password in Collaborator desktop clients. (COLLAB-7919)

**Single sign-on support for desktop clients.** To use desktop clients[484] (GUI Client, Command-Line Client), Office plug-ins[605] and IDE plug-ins[524] (Visual Studio Extension, Eclipse Plug-in) when single sign-on authentication is enabled, users should generate login tickets[318] and specify them in client connection settings instead of password. (COLLAB-7916)

**Version info in comments and defects.** Conversation panes for file comments and defects now display information about the specific revision where that comment or defect was made. For files from source-control systems, the version information link displays the commit ID, whereas for locally uploaded files it displays the upload number. (COLLAB-5422)




*Version info for source-control files*

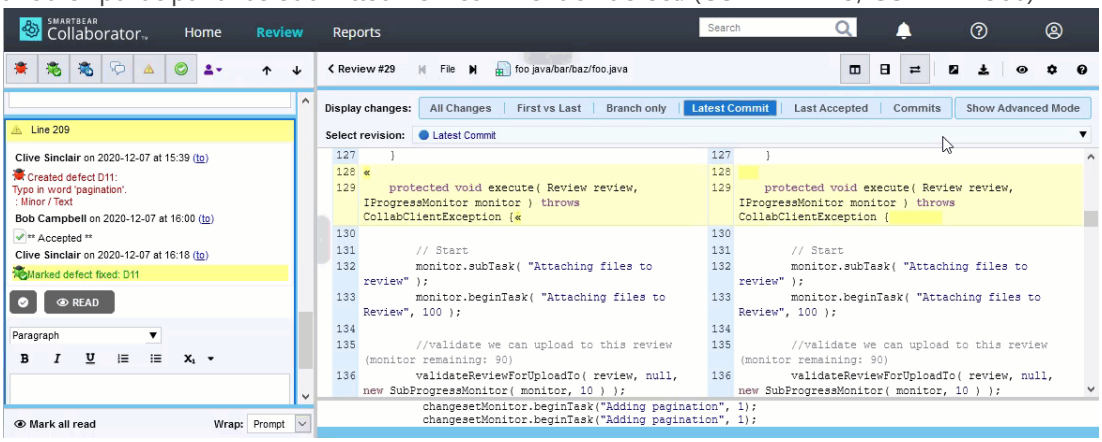



*Version info for local files*

**Support for GitHub teams in pull request reviewers and Code owners.** If some team was added as reviewer when creating pull request on the GitHub side, or some team was specified in CODEOWNERS file, Collaborator's GitHub integration tries to match members of that team with Collaborator users and can automatically add those users as reviewers on Collaborator side. (COLLAB-7838)

**More improvements:**

- **Expandable chat panel and Focus mode of Diff Viewer.** Chat panel of Diff Viewer now could be collapsed and expanded. Additionally a new Focus mode button was added to the Diff Viewer toolbar. When you want to focus on reviewed code or document, just click Focus mode to hide both the main toolbar and the Chat section. When Chat is hidden and you click anywhere in the Diff Viewer content, the Chat pane automatically expands so that you can input your comment or see existing comment or defect. It also automatically expands when another participant has submitted new comment or defect. (COLLAB-7729, COLLAB-7986)




*See how focus mode works*

- **Restrict moving pushpins to creator only.** The Allow users to move comments 185 option got another possible value which restricts moving coordinate comments to review creators only. (COLLAB-7879)

- **Support for TLS 1.3.** Collaborator now supports the `TLS 1.3` protocol for HTTPS connections. If your Collaborator server uses HTTPS, the `TLS 1.3` protocol will become enabled during server upgrade. To use the TLS1.3 protocol your Collaborator server should be running on OpenJDK 11 or higher, since support for this protocol was added in OpenJDK 11. (COLLAB-7841)

- **User list improvements.** Now the *Show active users* filter displays all users that consume licenses. (COLLAB-7962)

- The "Fix only" value was added to the *Can change other users' defects* 282 role setting. (COLLAB-7703)

- The `admin user edit` command got new options to modify the users first and last name. (COLLAB-7662)

- fixed - Some of the reviews cannot be loaded because the max integer value was reached in `MetaDataValueInteger` database table. (COLLAB-8061)
- fixed - Current cell and cells with conversation were not highlighted in Diff Viewer since version 13.1. (COLLAB-7980)
- fixed - Pushpin comments were cloned across all document reviews. (COLLAB-7979, COLLAB-7975)
- fixed - Unable to upload .docx, . doc, or .pdf file for a second time if a review with this file was been deleted previously. (COLLAB-7963)
- fixed - Collaborator allowed creating several remote hosting service configurations for the same repository. (COLLAB-7945)
- fixed - In some cases, Collaborator could display incorrect move to phase button when navigating between reviews. (COLLAB-7936)
- fixed - Permanently remember the "Continue from the beginning of the file" warp option. (COLLAB-7933)
- fixed - When upgrading from 11.0 and below servers, the maximal length of the `RemoteSystemReviewRelation.title` column became 256 symbols instead of 2000. (COLLAB-7902)
- fixed - Defects were not highlighted with red frame when reviewing .xls/.xlsx files. (COLLAB-7901)
- fixed - GitHub review creation fails when processing a pull request containing a commit with a file move. (COLLAB-7898)
- fixed - MySQL: Problem with importing dumps with string records larger than 65535 characters. (COLLAB-7896)
- fixed - GitLab pipeline was not updated on review completion. (COLLAB-7828)
- fixed - Collaborator calculates file metrics for non-modified files controlled by TFS. (COLLAB-7825)
- fixed - Participant Custom Fields using a Dropdown Series do not display properly. Labels do not associate with their designated cells. (COLLAB-7790)
- fixed - Check string input before deserialization of `UpdateMemento` class. (COLLAB-7718, CVE-2020-26118)
- fixed - Collaborator sends the "Review deadline has changed" notification in the Planning phase. (COLLAB-6166)
- fixed - Collaborator calculates file metrics for non-modified files controlled by TFS. (COLLAB-7825)

- fixed - MySQL: Problem with importing dumps with string records larger than 65535 characters. (COLLAB-7896)
- fixed - Defects were not highlighted with red frame when reviewing .xls/.xlsx files. (COLLAB-7901)
- fixed - Unable to upload .docx, . doc, or .pdf file for a second time if a review with this file was been deleted previously. (COLLAB-7963)
- fixed - Pushpin comments were cloned across all document reviews. (COLLAB-7979)
- **Retain accepted conversation  status for code and text.** If some changes in source code and text files were accepted earlier and have not changed in newly uploaded revision, the respective line comments will keep their accepted status 433. (COLLAB-4829)
- **Display name improvements.** Now the Display name 317 user setting can be up to 128 characters long. Besides, in Collaborator 13.3 we have added an option to restrict editing by end-users, so that only administrators could modify all user display names 189. (COLLAB-7730)
- Azure DevOps integration, now can automatically assign Collaborator reviewers when some specific users were added as pull request reviewers on the Azure DevOps side and integration can match those Azure DevOps users with Collaborator users. (COLLAB-7720)
- Verified support for Rational Team Concert version 7.0.1. (COLLAB-7676)
- Bitbucket Server 7.0 and later no longer support the upstream changes functionality. Collaborator's integration with Bitbucket Server will not support it either.
- Single sign-on integration using Java servlet for Tomcat is deprecated and will be removed in future releases. Please consider implementing single sign-on using SAML 132 or Atlassian Crowd OpenID 145 instead.
- fixed - The column with the row numbers could become hidden during the horizontal scrolling. (COLLAB-7884)
- fixed - Improve Group Sync option to affect GitLab integrations only. (COLLAB-7861)
- fixed - Review cannot be created or updated if pull request contains submodule commit. (COLLAB-7844)
- fixed - The "Phase Change" notification was not always sent. (COLLAB-7811)
- fixed - Checklist panel doesn't display properly after expanding. (COLLAB-7788)
- fixed - An exception could occur in `getImageWriterCallback()` method during locator promotion. (COLLAB-7766)
- fixed - Participants added by subscriptions were not always removed if the review template was changed. (COLLAB-7687)

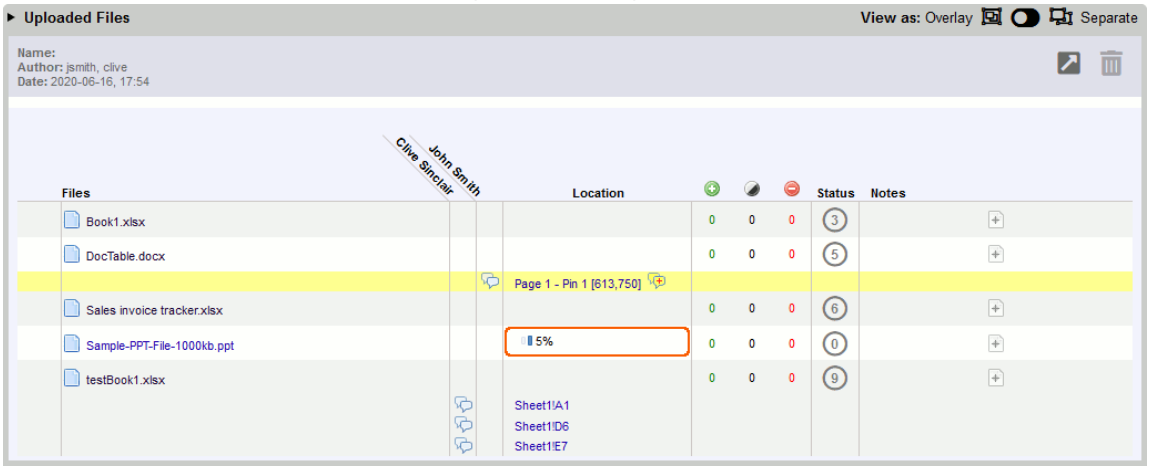
- fixed - Misplaced sections in Review Detail report. (COLLAB-7656)
- fixed - Do not call "p4 diff -sr" in Perforce triggers. (COLLAB-7614)
- fixed - Collaborator does not land on a target URL if access was interrupted by single sign-on processing. (COLLAB-6135)
- fixed - The content of the "List Review Stalled Author Not Reworking" template contained the username variable instead of the author. (COLLAB-6031)
- fixed - The value of Wrap option did not change when selecting "Remember my decision" in "Navigate to next location" dialog. (COLLAB-2231)
- fixed - Collaborator calculates file metrics for non-modified files controlled by TFS. (COLLAB-7825)
- fixed - Review cannot be created or updated if pull request contains submodule commit. (COLLAB-7844)
- fixed - The column with the row numbers could become hidden during the horizontal scrolling. (COLLAB-7884)
- fixed - MySQL: Problem with importing dumps with string records larger than 65535 characters. (COLLAB-7896)
- fixed - Defects were not highlighted with red frame when reviewing .xls/.xlsx files. (COLLAB-7901)
- fixed - Unable to upload .docx, . doc, or .pdf file for a second time if a review with this file was been deleted previously. (COLLAB-7963)
- **Caching of most frequently used API calls to remote repositories.** In order to decrease number of calls to repository hosting servers, Collaborator caches some of the most often retrieved entities from APIs (commits, pull request diffs, commit diffs). By default cache contains up to 20000 entities for each active remote system integration. To change the default cache size or disable caching, use the server's `-Dcom.smartbear.collab.datamodel.remotesystem.cache.size` Java VM Option 1232. (COLLAB-5913, COLLAB-6373, COLLAB-7436, COLLAB-7437, COLLAB-7438)
- **Re-connecting when remote repository server is temporary unavailable.** If the remote repository server is not available, Collaborator will wait for some time and will make several more attempts to reach the server. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on. Default wait time is 1 second and Collaborator will perform 3 retries. Wait time and maximal number of attempts could be configured via Java VM Options 1234. If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to the remoteSystem.log 169. This behaviour currently applies to GitHub, Bitbucket, Bitbucket Server, and Azure DevOps integrations. (COLLAB-6450)

- Added a progress indicator on Review screen during upload of pull request files. (COLLAB-6449)
- Built-in syntax highlighting schemes were added for Swift, Kotlin and Typescript programming languages. (COLLAB-7616, COLLAB-7617, COLLAB-7618)
- fixed - There could be significant differences between license usage values: NumUsersCurrentlyLoggedIn and LicenseSeatsInUse. (COLLAB-7725)
- fixed - Collaborator was unable to load a review (the review screen hung) if that review was based on a template that had no Moderator role specified. (COLLAB-7723)
- fixed - There were too many unexpected "Metrics are missing ..." messages in the log file. (COLLAB-7705)
- fixed - Too many errors with "com.google.common.cache.LocalCache" in the log made the log hardly readable. (COLAB-7702)
- fixed - Sometimes, the pins' coordinates got broken after you uploaded a newer version of the file being reviewed. (COLLAB-7677)
- fixed - Collaborator SAML metadata file contained a wrong certificate, which caused errors when attempting single logout. (COLLAB-7639)
- fixed - A review created from a pull request did not contain the review materials if the PR included file permission changes for files larger than 1 MB. (COLLAB-7628)
- fixed - The Defect Log did not display the defect text if it was long and contained HTML tags. (COLLAB-7559)
- fixed - The "Show conversation with open defects" filter does not work when there are unread comments. (COLLAB-6949)
- fixed - The Diff Viewer did not show some defects and comments until a user started scrolling the list. (COLLAB-6901)
- When deleting a review[346], Collaborator now deletes files that were uploaded for this review from the content storage[97] as well (unless they are used in some other reviews). (COLLAB-5300)
- New option to enable or disable user role synching[140] between SAML and Collaborator servers, when synching user membership with SAML SSO server[142]. (COLLAB-7440)
- New option to specify type of SAML SLO signature[139]. It allows selecting whether to use embedded attribute signature (HTTP-POST binding), or send signature as url parameter (HTTP-Redirect binding). (COLLAB-7430)
- The password of built-in administrator could be changed via command-line, just like any other user's password. (COLLAB-7275)

- Improved logic for uploading files and changelists to the review. (COLLAB-5561, COLLAB-4680)
- The Report Error feature[483] of Web client now uses SmartBear Support Portal to report issues. (COLLAB-7328)
- Messages about new version uploads are now displayed only in the Review Chat and in file's Overall comment section. (COLLAB-7345)
- Improved the performance of the "List Complete" notifications on huge databases. (COLLAB-7429)
- Calculate time tracking activity on active tab only; make tracking mechanism more accurate. (COLLAB-6140)
- The review timeline now logs who and when has changed review template, added or removed participants, changed participant roles, added or removed checklists.(COLLAB-5942)
- New option to enable or disable user role synching[140] between SAML and Collaborator servers, when synching user membership with SAML SSO server[142]. (COLLAB-7440)
- New option to specify type of SAML SLO signature[139]. It allows selecting whether to use embedded attribute signature (HTTP-POST binding), or send signature as url parameter (HTTP-Redirect binding). (COLLAB-7430)
- The password of built-in administrator could be changed via command-line, just like any other user's password. (COLLAB-7275)
- Improved logic for uploading files and changelists to the review. (COLLAB-5561, COLLAB-4680)
- **Multiple checklists per review.** Starting from version 13.0 it will become possible to have several checklists[272] in a template. When creating review author could select which of the checklists to use. (COLLAB-3695, COLLAB-2005)


- **Select All Items for checklists.** It became possible to check or un-check all items in a checklist at once. To do this, click the check mark in the Status column header (not available in edit-mode). (COLLAB-6737)
- **New Approved by workflow state.** In previous versions, some of non-approving roles (in standard roles scheme they are Observer and Moderator) could remain in "*Active*" or "*Waiting* " state when the review was completed without their interaction. Now such participants would get the "*Approved by workflow*" state. (COLLAB-7257)

- **Parallel conversion of files** - Previously all uploaded materials were converted in one chunk, so you had to wait until all uploaded materials are converted. Now each document is uploaded and converted separately, so that if there is a delay or conversion issue with one of the files, it would not lock all review materials and you can work with other files.
When the respective document is being converted, its progress is displayed in the **Location** column of the Review Materials section. (COLLAB-6865)




*Document conversion progress*

- **Asynchronous conversion upon initial upload** - When a document is uploaded for the first time, it is now converted in asynchronous threads and could be displayed in Diff Viewer as soon as its top pages are ready. (COLLAB-6777)

- **Conversion on demand** - When user navigates to a specific page that is not yet converted, that page is converted primarily. (COLLAB-6783)

- **Performance speedup** - We have optimized some of conversion and comparison routines to improve their performance. (COLLAB-7427)

- Clarified and improved text of error messages that may occur during document conversion. (COLLAB-6155)

- **Retain document zoom when changing pages.** Diff Viewer now keeps document zoom value in browser cache (per each document). Initial zoom value is specified by the Default Scale for Documents in Diff Viewer 326 setting. When you change scale of current document in the Diff Viewer it will be cached so that all further pages of that document would be displayed in the same scale. (COLLAB-6906, COLLAB-6133, COLLAB-5854)

- Information about pull request branches is now displayed in the Chat section of the Review Summary screen. (COLLAB-4138)

- Now you can remove files with a warning about missing fonts unless they have comments from other participants. (COLLAB-6792)

- All Web Client theme-related setting are now cached to decrease amount of queries to database. (COLLAB-7244)

## 11.2.2 Version 12

### 12.5.12504 - February 3, 2021

fixed - Refactor the `GetPageCount` method to reduce the number of calls to the `FileMetrics` table and avoid blocking queries from multiple concurrent users. (COLLAB-7786)

### 12.5.12503 - July 24, 2020

**Improvements**

- New option to enable or disable user role synching[140] between SAML and Collaborator servers, when synching user membership with SAML SSO server[142]. (COLLAB-7440)
- New option to specify type of SAML SLO signature[139]. It allows selecting whether to use embedded attribute signature (HTTP-POST binding), or send signature as url parameter (HTTP-Redirect binding). (COLLAB-7430)
- Improved FileMetricCache to use version id instead of version. (COLLAB-7405)
- Improved logic for uploading files and changelists to the review. (COLLAB-5561, COLLAB-4680)

**Bug Fixes:**

fixed - Decrease amount of queries while opening review summary. (COLLAB-7490)

fixed - After calling version.getContentStream() input stream was not closed which could lead to significant number of opened streams. (COLLAB-7484)

fixed - In certain cases Diff Viewer did not show navigation panel when opening spreadsheet file. (COLLAB-7434)

fixed - Collaborator server could not start after upgrade if single sign on was enabled before the upgrade. (COLLAB-7393)

fixed - Failed to generate review detailed report if there are any defects in overall review chat. (COLLAB-7283)

### 12.5.12502 - June 29, 2020

fixed - Database upgrade failure when multiple themes are present but dark theme is missing. (COLLAB-7379)

fixed - On SQL Server databases with indexes enabled, for example on Azure SQL database with automatic tuning mode, upgrading Collaborator may fail with the "The index 'NNN' is dependent on column 'MMM'." exception. (COLLAB-6267, COLLAB-7473)

### 12.5.12501 - June 15, 2020

fixed - Remove redundant calls to database while getting user unread comments and info about remote system relations. (COLLAB-7412)

fixed - Improve "unread conversations" SQL query that caused performance degradation. (COLLAB-7410)

**12.5.12500 - May 18, 2020**

**New Features**

- **Automatically start pull request reviews.** New Move pull request reviews to Inspection 192 setting allows to start reviews created by repository hosting integrations 826 automatically. Once enabled, Collaborator will verify if the review meets the workflow requirements (for example, has minimal number of participants) and will move it to the Inspection phase on success. (COLLAB-6781)

- **New Text file types 198 setting to treat arbitrary files as text.**
  Diff Viewer detects type of review materials 363 based on the extension of the uploaded files. For example, `.java` files typically stand for Java source code, `.rtf` and `.doc` are typically word-processing documents and so on. However, some extensions could stand for multiple types of data and this could mislead the Diff Viewer. For instance, the `.pot` extension could be either a PowerPoint template file or a portable object file. In this case, you can use this setting to specify explicitly which file types should be treated as text-based files. (COLLAB-5538, COLLAB-6485)

- **Export reports in DOCX format.** Now all reports 467 could be exported to editable DOCX format. This could be useful, for example, to remove some sensitive data from the reports before sharing them with contractors or third party auditors. (COLLAB-6915)

**More Improvements**

- GitHub pull request reviewers are automatically assigned as reviewers in Collaborator. You can specify GitHub users as reviewers when you are creating a pull request. If your teammates link 885 their GitHub accounts with their Collaborator accounts correctly, then Collaborator will automatically add those users as reviewers to the created review on the Collaborator side. (COLLAB-5493)

- Previously reviews created by repository hosting integrations had pull request description displayed in the Overview field. Yet this field is custom field and could be disabled in template settings. Now pull request description and all comments that users made to pull request/ commit on the remote repository side are copied to the Chat section of the Review screen. (COLLAB-6858)

- Now a warning about missing fonts is less annoying. In previous versions, each missing font and its replacement was reported separately, now a warning has a list of missing fonts and a list of their replacements. Also a new option 185 was added to hide this warning in the DiffViewer. (COLLAB-6792)

- Optimized processing of spreadsheets with big number of defects and comments. (COLLAB-6927)

- Web UI now disables controls while processing requests on review state changes. This helps to avoid sending duplicated requests and thus reduces server load. (COLLAB-7189)
- Limited the number of attempts to reconvert a document. If document conversion has failed during upload, opening it in DiffViewer makes another attempt to reconvert that document. In previous versions Collaborator continuously tried to reconvert the document while the DiffViewer window was opened. Since version 12.5.12500 DiffViewer will make only 3 attempts to reconvert the document upon opening. (COLLAB-6942)
- Updated Apache Tomcat server to version 9.0.34 to resolve a number of performance and security issues. (COLLAB-7153)
- Updated the Aspose library used for document convertion. (COLLAB-6684)
- The `SignatureValue` and `DigestValue` elements were added to SAML logout request. (COLLAB-7006)

**Bug Fixes**

fixed - Remote Accounts user preference page could display only 100 remote servers. (COLLAB-7276)

fixed - JIRA plug-in caused re-indexing failure on an on-premise JIRA 8.5 Server. (COLLAB-7237)

fixed - Improve file metrics cache to purge invalid data when DB connection fails. (COLLAB-7228)

fixed - GitLab integration failed to attach files to a review when a development branch was updated with changes from main branch before a merge request. (COLLAB-7147)

fixed - Collaborator could not properly handle diffs generated for two mercurial revisions. (COLLAB-7085)

fixed - Document with a table failed to convert when column size was not specified. (COLLAB-6943)

fixed - Some defects were duplicated after upgrading Collaborator from v12.3.12304 to v12.4.12400. (COLLAB-6916)

fixed - Null pointer exception could occur when uploading subsequent revisions of documents with comments or defects. (COLLAB-6899)

fixed - Bitbucket Server integration could not match pull requests to a registered repository configuration if repository name contained lowercase and uppercase letters. (COLLAB-6892)

fixed - Unable to mark the defect as Read when comment have no related conversation (corresponding conversation has been deleted). (COLLAB-6862)

fixed - Author is not getting moved into the "Waiting" state after the review is moved into the "Inspection" phase when the "Phase-change waits for finished" setting is enabled for this role. (COLLAB-6847)

fixed - On some Ubuntu environments generating reports in PDF-format could produce zero-byte files. (COLLAB-6844)

fixed - Repository hosting integrations could create empty review while processing merge commit with modified file which was removed in one of the branches. (COLLAB-6832, COLLAB-5948)

fixed - Remove the stpwvcm-1.0-custom.jar from the Client installer. (COLLAB-6820)

fixed - When an already-defined locator is selected in the Diff Viewer, the keyboard focus was not set to the associated comment textbox in Chrome browser. (COLLAB-6819)

fixed - Keyboard shortcuts in the Diff Viewer don't work correctly. (COLLAB-6810)

fixed - Make Missed fonts warning less annoying. (COLLAB-6792)

fixed - Hyperlinks to files with underscore characters in name were rendered without the underscore characters. (COLLAB-6787)

fixed - Endless Loading indicator when attaching links to PDF files as Live URLs. (COLLAB-6754)

fixed - Space character was URL-encoded twice for GitHub file names. (COLLAB-6701)

fixed - No Loading page status when opening subsequent revisions of documents with comments or defects. (COLLAB-6690)

fixed - When a reviewer made comments while in Waiting state his state was changed to Active. (COLLAB-6663)

fixed - Issue when converting Visio image having a solid-fill box grouped with a text box. (COLLAB-6647)

fixed - Too many "Limiting database result to NNN records" warnings. (COLLAB-6608)

fixed - Clicking the "Add as defect" button removed the last character specified in the text entry box. (COLLAB-6525)

fixed - Template based e-signatures affects other reviews status information that are not using templates that required signatures. (COLLAB-6483)

fixed - Author's state didn't change from Waiting to Completed upon completing the review while the electronic signature is enabled. (COLLAB-6316)

fixed - Deleted file and merge commits in some cases could cause content loading errors. (COLLAB-5948)

fixed - When the "Report Access" option was set to "Respect permissions", Collaborator ignored the value of the "Max # Rows" option and returned all the reviews to which the user had access. (COLLAB-5845)

fixed - If there are differences in code, fixed sizing fonts look misaligned. (COLLAB-4190)

### 12.4.12403 - June 15, 2020

fixed - Remove redundant calls to database while getting user unread comments and info about remote system relations. (COLLAB-7412)

fixed - Improve "unread conversations" SQL query that caused performance degradation. (COLLAB-7410)

### 12.4.12402 - April 15, 2020

added - **Upload file revisions via ClearCase Remote Client.** Previously it was possible to upload revisions of files controlled by ClearCase only by using regular ClearCase Client. Now the same action could be performed when using ClearCase Remote Client (COLLAB-4962).

fixed - Null pointer exception could occur when uploading subsequent revisions of documents with comments or defects. (COLLAB-6899)

### 12.4.12401 - March 31, 2020

**New Features**

- **Mapping users by emails.** Previously, Collaborator could automatically match Collaborator user account with a remote system account when both of them used identical login names. Now it also checks email address (if available) and matches accounts when they coincide. This reduces the need to link the accounts manually 885. (COLLAB-6047)
- **GitHub build statuses.** If your GitHub repositories 836 use continuous integration, the Remote System Links 353 section now display build statuses from those CI systems (Jenkins, Travis and so forth). (COLLAB-5500)
- **Text alignment in spreadsheets.** Diff Viewer now supports aligned/centered text in spreadsheet cells. (COLLAB-4669)

**SAML Single Sign-On Improvements**

- **Syncing groups with SAML identity provider.** You may now establish group synchronization 241 between Collaborator and your SAML single sign-on server. In this case, Collaborator will retrieve user membership in groups from the single sign-on server when the users login. (COLLAB-4242)

- **Syncing user privileges with SAML identity provider.** Previously all accounts that were created via SAML integration had regular user privileges in Collaborator. Now users with admin privileges on the SSO side (having `Role` parameter equal to admin) will automatically become administrators in Collaborator. And wise versa: accounts having admin privileges on Collaborator side, but having regular privileges on the SSO side would become regular users in Collaborator. (COLLAB-3145)

- **SAML metadata.** Collaborator server now provides its SAML metadata[140] at specific endpoint. (COLLAB-5951)

**More Improvements**

- Now support for Markdown formatting in comments and defects could be enabled or disabled via admin option[185]. Existing comments and defects will retain their current formatting. (COLLAB-6672)

- Better progress indication when loading, converting or comparing review documents. (COLLAB-6564)

- Support for Microsoft SQL server 2019 and Oracle 18c and 19c (COLLAB-6128, COLLAB-6537)

**Bug Fixes**

fixed - Collaborator tried to connect to an HTTP Azure DevOps server over HTTPS (COLLAB-6869)

fixed - JSON API Error: User and Review Pool Group must not both be null (COLLAB-6764)

fixed - Error when opening spreadsheet file that uses unavailable font. (COLLAB-6747)

fixed - Upgrading to Collaborator 11.3.11301 could fail on Oracle DB because of incorrect handling of Oracle CTX indices. (COLLAB-6730)

fixed - The Overview field stopped rendering of newline characters. (COLLAB-6696)

fixed - Collaborator client did not pick up config from p4config (COLLAB-6695)

fixed - Cannot change defect type or defect custom field without changing its description field. (COLLAB-6668)

fixed - Azure DevOps integration endpoints did not consider that Azure DevOps Server (on-premises) could be installed to a non-default location. (COLLAB-6665)

fixed - Incorrect LOC calculation in Customizable review report. (COLLAB-6627)

fixed - Users had to login twice when they had Reports page open or minimized for more than 1 hour. (COLLAB-6610, COLLAB-6289)

fixed - Customizable Review Reports don't provide the External URL in links of PDF and XLS files. (COLLAB-6476)

fixed - Jump to next change in code review does not always work. (COLLAB-6188)

fixed - Ensure that active review time metrics are accurate. (COLLAB-6140)

fixed - Notifications fail when users have two LDAP-mapped email addresses. (COLLAB-5907)

fixed - The "location" field in Reports showed the original location of the defects created for the spreadsheet cells instead of current one. (COLLAB-5789)

fixed - GitLab integration could create review for empty pull request. (COLLAB-5710)

fixed - Text strings that should be identified by automatic links feature were not being transformed into hyperlinks when used within checklists. (COLLAB-5547)

fixed - Clicking a hyperlink in a checklist item changes check-box status (COLLAB-5546)

fixed - Collaborator was partially parsing the File Pattern and subscribing users when it should not. (COLLAB-5256)

fixed - Automatic SCM validation with incorrect password could lock-up source control user accounts. (COLLAB-5120)

fixed - Clicking on checklist items could reopen closed reviews by mistake. (COLLAB-4828)

fixed - The addp4diffs command does not ignore the identical files. (COLLAB-4685)

fixed - Checklist items could be rendered incorrectly when they included newline or tab characters. (COLLAB-3970)

fixed - Long URL in checklist items were truncated and did not work (COLLAB-2266)

### 12.4.12400 - February 5, 2020

**New Features**

- Links to Azure DevOps work items. If a pull request is associated with some work item on Azure DevOps side, the link to work item will be added to the review's Remote System Links 353 section. Moreover, if the integration's "Auto close related work items 875" option is enabled, the respective work item will be automatically closed when pull request is merged. (COLLAB-6139)

- More granular permissions for roles allowed to modify review's General Info section. Two new role settings were added: "Allowed to modify review checklists 284" and "Allowed to modify review participants 285" (COLLAB-4868, COLLAB-5740)

- New "Enable Signature in SLO request 139" setting of SAML SSO configurations that specifies whether to append digital signature to single logout requests.

**Bug Fixes**

fixed - Underscore characters were removed in comments and defects created via JSON API. (COLLAB-6580)

fixed - Files added as "Binary file types" and uploaded via Web UI could be removed from the review if the user's session was terminated. (COLLAB-6535)

fixed - Icon could be missing for comments or defects made in Word documents having several revisions that were staged to the same Git commit. (COLLAB-6477)

fixed - Underscore character inside text fragments converted style to Italic in cases where it shouldn't. (COLLAB-6467)

fixed - Access restriction violation may occur in existing reviews when new group was created, users were moved to that group and "Restrict Access to Review" option was set to "Group Based". (COLLAB-6366)

fixed - Report were not generated if the "Restrict Access" column was checked and the "ID" column was unchecked. (COLLAB-6364)

fixed - The comments and accept icons under the "Review Material" section could be displayed in a wrong user column. (COLLAB-6268)

fixed - When parent of some Git commit was merge commit, LOC metric in Overlay View was calculated ignoring number of deleted lines. (COLLAB-6246)

fixed - NullPointerException could occur if review materials had comment or defect with exactly the same coordinates. (COLLAB-6203)

fixed - Adding a comment to an Excel spreadsheet having multiple worksheets and rows could take significant time. (COLLAB-6180)

fixed - Problem with highlighting identifiers with "_" in all supported syntax schemas. (COLLAB-5899)

fixed - Diff Viewer revision selector shows multiple items for the same revision that was uploaded multiple times. (COLLAB-5617)

fixed - The command-line client in debug mode was printing passwords in plain text. (COLLAB-5548)

fixed - SAML logout requests did not contain the NameID element required by the SAML 2 specification. (COLLAB-5310)

fixed - A custom field with a "validator" that doesn't allow empty values was not mandatory in Eclipse Plug-in. (COLLAB-4211)

fixed - The material section takes too long to display when opening Excel workbook with multiple worksheets and comments. (COLLAB-3887)

fixed - An JSON API error on attempts to create new reviews via command-line when the Overview custom field is disabled in default template. (COLLAB-3606)

**12.3.12306 - May 12, 2020**

fixed - No Loading page status when opening subsequent revisions of documents with comments or defects. (COLLAB-6690)

**12.3.12305 - March 31, 2020**

fixed - Null pointer exception could occur when uploading subsequent revisions of documents with comments or defects. (COLLAB-6899)

fixed - Azure DevOps integration endpoints did not consider that Azure DevOps Server (on-premises) could be installed to a non-default location. (COLLAB-6665)

**12.3.12304 - January 21, 2020**

fixed - An issue with reviews created for changes in source-control repositories: Collaborator always set an Administrator as a review author. This happened because it failed to automatically link Collaborator and source-control accounts that had the same names. (COLLAB-6542)

**12.3.12303 - January 14, 2020**

added - Now when document conversion fails because of missing images, files or directories, Collaborator will attempt to recover automatically. (COLLAB-6437, COLLAB-6415, COLLAB-6434)

fixed - GitHub integration was unable to create or update a review for commits with more than 300 files when some files were larger than 1 Mb or some files have been deleted. (COLLAB-6444, COLLAB-6441)

fixed - Render Web UI home screen even if the incoming/outgoing information is not calculated yet. (COLLAB-6436)

fixed - Case-sensitive table names issue when upgrading Collaborator server from 11.4.11400 to 12.2.12203 with MySQL on Unix\Linux (COLLAB-6337)

fixed - Adding a comment to an Excel spreadsheet having multiple worksheets and rows could take significant time. (COLLAB-6180)

fixed - Obfuscated review dumps generate ClassNotFound exception on load (COLLAB-6103)

**12.3.12302 - December 24, 2019**

added - Links to files having the same name but different path/repo were leading to one of the files. Now links correctly handle file paths. Besides, they include SCM or changelist identifiers, so that they point to a particular revision of a file. (COLLAB-1580)

added - Repository hosting integrations now use thread pool and process up to 10 webhook events in parallel (COLLAB-6371)

fixed - Creating a new defect with exactly the same coordinates or line number as the existing one could break reviews. (COLLAB-6431)

fixed - SAML SSO: Infinite redirect on identity provider side if user was disabled in Collaborator. (COLLAB-5543)

fixed - Checklist and participant custom fields were not aligned with the other fields in exported review detail report. (COLLAB-5474)

### 12.3.12301 - December 5, 2019

fixed - JSON API error in addfiles command. (COLLAB-6403)

fixed - Reviews on pull requests containing large number of files were created too long. (COLLAB-6372)

fixed - An "ambiguous column" error during "All Defects" report generation against Oracle DB (COLLAB-6235)

### 12.3.12300 - December 4, 2019

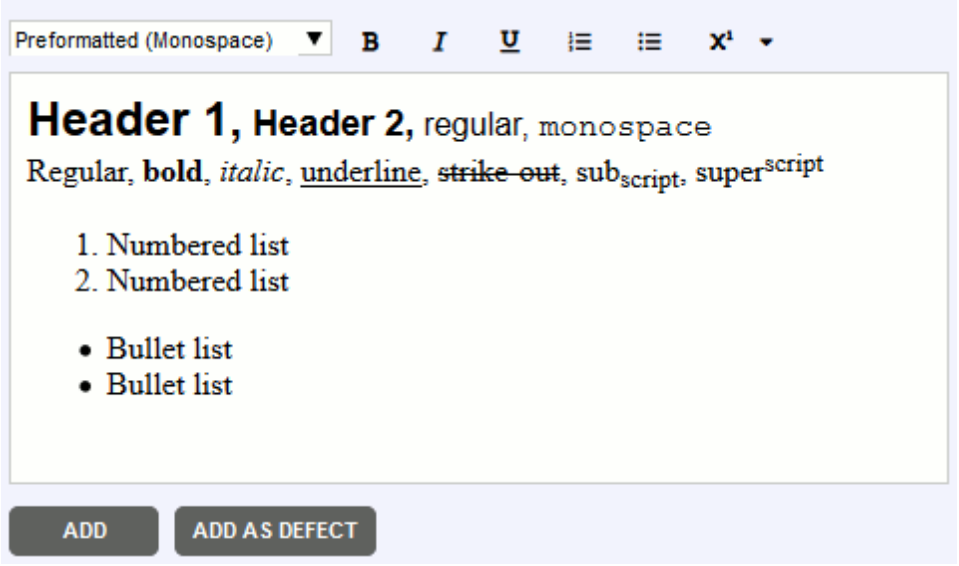
**New Features:**

- **Azure DevOps Integration**

Now you can use Collaborator to review changes in Git repositories hosted at cloud Azure DevOps Services 875 (dev.azure.com or visualstudio.com), as well as on on-premises Azure DevOps Servers 875. (COLLAB-3344)

- **Rich-text and Markdown Comments**

Comment and defect fields of Collaborator Web UI now support rich-text and Markdown formatting 447. (COLLAB-3320)




**More improvements:**

- Eclipse Plug-in: Custom fields of drop-down type now have the "type to filter" feature. (COLLAB-4517)
- Now a dialog is displayed when you try to leave Diff Viewer or Review Screen while having any un-submitted comments or defects. (COLLAB-5959)
- Updated Apache Tomcat server to version 9.0.5 to resolve a number of performance and security issues (COLLAB-5865)
  Tomcat 9 had deprecated the `org.apache.coyote.http11.http11protocol` class. To establish SSL connection with Collaborator 12.3 and later you should now use the `org.apache.coyote.http11.Http11NioProtocol` class instead. See detailed instructions in Configure HTTPS 114.

**Discontinued Support:**

- With the release of Collaborator 12.3, we will no longer be able to support customers running Microsoft SQL Server 2008 or MySQL 4.1, 5.0, 5.1, 5.4, 5.5. Please upgrade your database to a newer supported version when upgrading Collaborator.

To learn about further plans on deprecating functionality, see Sunset Announcements 32.

**Bug Fixes**

fixed - Add possibility to send `Destination` parameter in SAML SLO request. (COLLAB-6365)

fixed - Visual Studio Extension couldn't connect to Azure DevOps in Visual Studio 2019. (COLLAB-6347)

fixed - The accepted icons could be dropped for files that have not changed which were uploading along with other files that were changed. (COLLAB-6250, COLLAB-6308)

fixed - Review materials were not displayed if an exception had occurred while processing merged, re-based or reverted commits. (COLLAB-6277)

fixed - UI elements for member selection in group editor did not work. (COLLAB-6233, COLLAB-6230)

fixed - Diff Viewer doesn't show commit with DOCX changes if previous commit was not uploaded to review. (COLLAB-6223)

fixed - Diff Viewer: problem with Last commit display mode for changes in DOCX. (COLLAB-6221)

fixed - Exception on opening DOCX file in Diff Viewer for Git commits added in reverse order. (COLLAB-6220)

fixed - Some icons were not rendered after upgrading Font Awesome from 4 to 5. (COLLAB-6216)

fixed - Issue with metrics for DOCX or image files added to a plain Git repository. (COLLAB-6210)

fixed - Drop-down for review subscription and template subscription was not displayed. (COLLAB-6208)

fixed - If the Collaborator server had a manually created group with the "All Users" option set to Yes, the newly created users could access the groups to which they had no permissions. (COLLAB-6189, COLLAB-6163)

fixed - Collaborator Logo and Home button could overlay when navigating from Reports or Admin sections in Chrome browser. (COLLAB-6176)

fixed - Login ticket timeout was not updated when user logged-in via Web Client. (COLLAB-6110)

fixed - The "Ignore Whitespaces" setting affected listed number of chats, preventing some chats be marked as read and closing reviews. (COLLAB-6109, COLLAB-5875, COLLAB-5800, COLLAB-5606)

fixed - Fixed exporting of customizable reports to different formats. (COLLAB-6059)

fixed - Cannot generate PDF versions of the Review Detailed Report. (COLLAB-6040)

fixed - If the "Subscription mode" option was set to "Requested", participants added by the "Template subscription" feature were removed from the participant list. (COLLAB-6036)

fixed - Empty history for files that were deleted while resolving merge conflict. (COLLAB-5948)

fixed - NullPointerException when loading active reviews after migrating from a legacy version of Collaborator. (COLLAB-5920, COLLAB-5490)

fixed - Duplicate pins could be created while reviewing 1000+ page documents. (COLLAB-5918)

fixed - SSO logout may not work due to missing `ds:Signature` parameter. (COLLAB-5843)

fixed - The Upload list of Review Materials was rendered incorrectly if there were more than 40 server-side source control integrations. (COLLAB-5834)

fixed - git-lfs support made uploading commits from the client very slow. (COLLAB-5739)

fixed - SQL error on Oracle DB: CLOB in WHERE clause. (COLLAB-5699)

**12.2.12207 - December 12, 2019**

fixed - Creating a new defect with exactly the same coordinates or line number as the existing one could break reviews. (COLLAB-6431)

**12.2.12206 - December 5, 2019**

fixed - JSON API error in addfiles command. (COLLAB-6403)

**12.2.12205 - November 28, 2019**

fixed - Reviews on pull requests containing large number of files were created too long. (COLLAB-6372)

**12.2.12204 - November 20, 2019**

fixed - The accepted icons could be dropped for files that have not changed which were uploading along with other files that were changed. (COLLAB-6250, COLLAB-6308)

**12.2.12203 - October 30, 2019**

fixed - Review materials were not displayed if an exception had occurred while processing merged, re-based or reverted commits. (COLLAB-6277)

fixed - UI elements for member selection in group editor did not work. (COLLAB-6230)

fixed - Drop-down for review subscription and template subscription was not displayed. (COLLAB-6208)

fixed - Collaborator Logo and Home button could overlay when navigating from Reports or Admin sections in Chrome browser. (COLLAB-6176)

**12.2.12202 - October 22, 2019**

fixed - Groups were not displayed in the UI properly due to cache issues. (COLLAB-6233)

fixed - Some icons were not rendered after upgrading Font Awesome from 4 to 5. (COLLAB-6216)

**12.2.12201 - October 15, 2019**

fixed - Issue with metrics for DOCX or image files added to a plain Git repository. (COLLAB-6210)

fixed - Exception on opening DOCX file in diff viewer for Git commits added in reverse order. (COLLAB-6220)

fixed - Diff Viewer: problem with Last commit display mode for changes in DOCX. (COLLAB-6221)

fixed - Diff Viewer doesn't show commit with DOCX changes if previous commit was not uploaded to review. (COLLAB-6223)

### 12.2.12200 - October 4, 2019

**New Features:**

- **Easier user association for pull request systems**

Earlier users had to specify remote user name for each remote repository. Now it is enough to specify one username 885 for each remote system or organization. If one login is not enough, users may specify multiple aliases, each on a separate line. (COLLAB-3966)

Old clients are incompatible with Collaborator server 12.2 and later. To work with it, you will need to upgrade your clients to version 12.2 or later. (COLLAB-6207)

- **Synchronize Groups with GitLab**

Now Collaborator can synchronize group membership and permissions with GitLab server. Once enabled, on every logging-in, Collaborator will check user membership in groups on the GitLab server, create new groups (if needed), and automatically add this user to the corresponding groups on the Collaborator server. See Syncing Groups and Their Members 239 for details. (COLLAB-5555)

**More improvements:**

- Added support for Visual Studio 2019 (COLLAB-5745)
- Previously some passwords could be exposed by Collaborator administrators via browser's console. Now all password and token handlers were revised and always hide passwords and tokens. (COLLAB-4933, COLLAB-6038)

**Discontinued Support:**

- JIRA Legacy integration is no longer supported since version 12.2.12200. This integration allowed creating Collaborator reviews from JIRA tickets' linked commits or use JIRA as notification channel. It was introduced in earlier versions of Collaborator and was later replaced by newer implementation of JIRA integration 887. Existing configurations will remain functional, and could be enabled or disabled on the Remote Systems Integration page of Admin UI. (COLLAB-6124)

**Sunset Announcements:**

- Collaborator's Bitbucket integration will no longer be able to support Mercurial starting on June 1, 2020, when Atlassian will officially remove Mercurial features and repositories from Bitbucket and its API.

**Bug Fixes**

fixed - JIRA Legacy: Missing trailing slash in review links from JIRA tickets. (COLLAB-6113)

fixed - Setting the "Default Send To State:" to "Someone Pokes Me" or to "Activity by Author Occurs" could prevent reviews from completing. (COLLAB-6111, COLLAB-5757)

fixed - Fixed exporting of customizable reports to different formats. (COLLAB-6059, COLLAB-5908)

fixed - Improved logic for identifying if user has unread conversations. (COLLAB-6052)

fixed - JIRA integration: Links to Collaborator reviews used base URL instead of External URL. (COLLAB-6039)

fixed - Files marked as "renamed" by the git mv command were not uploaded to reviews. (COLLAB-6023)

fixed - Wrong placement of findings in Excel sheets (COLLAB-5998)

fixed - The Review Summary page shows the Current state: null state if the Activity by Author Occurs wait option is selected. (COLLAB-5932)

fixed - If a database throws an exception, while running the "Content cache cleanup unused" command, it could delete files, even if they are still used in some reviews. (COLLAB-5898)

fixed - Author state is changed to Active instead of Completed when the review with materials is completed. (COLLAB-5825)

fixed - A review gets stuck on Loading materials animation if the file was deleted from the content cache folder. (COLLAB-5804)

fixed - If a push or merge request in GitHub affects more than 300 files, the integration will process only first 300 files from the request and add those files to a review. (COLLAB-5787)

fixed - Incorrect Check In date for ClearCase files that were uploaded by Add My Activities action (COLLAB-5763)

fixed - ClearCase: The "Accepted" icon was dropped for the files that were added to the changeset but remain unchanged. (COLLAB-5761)

fixed - When review had multiple participants, it could take up to 30s to open a review material in Diff Viewer (COLLAB-5704)

fixed - Low performance when spreadsheet had multiple comments. (COLLAB-5682)

fixed - Remote systems: always show merge commit on review summary. (COLLAB-5657)

fixed - Review file could contain multiple pushpins with the same coordinates (COLLAB-5616)

fixed - Review dump was broken for reviews with rebased commits (COLLAB-5598)

fixed - External Diff Viewer didn't open on Linux (COLLAB-5397)

fixed - The session identifier retains its value after logging in (COLLAB-5370)

fixed - The Add Files trigger returns incorrect summary of changed lines for those Perforce changelists that contains files that were marked for edit for the first time. (COLLAB-5134)

fixed - Improve dump generation for databases with billion and more records. (COLLAB-5017)

fixed - Embedded font didn't display in PPTX file. (COLLAB-4854)

fixed - The Tray Notifier does not re-connect automatically. (COLLAB-4389)

fixed - Some reviews are not showing the same LOC values on the reports as shown in the review material section. (COLLAB-4255)

fixed - Improve spreadsheet diff by comparing cell contents. (COLLAB-4215, COLLAB-4800)

fixed - The "Branches to track" field and a "Ignore pushes for branches" fields of Easy Add page were limited to 64 characters. (COLLAB-3892)

fixed - Remote Systems: The "Test connection" button actually saves the configuration after testing (COLLAB-3459)

fixed - Highlighting: Inconsistent result with Excel files (COLLAB-2518)

### 12.1.12101 - August 19, 2019

**New Features**

added - Collaborator now caches results of some remote repository API calls (commits, pull request diffs, commit diffs). This will decrease the overall number of calls to repository hosting servers, and reduces chances to hit the server's rate limit. (COLLAB-5913)

**Bug Fixes**

fixed - Integration issues with on-premises GitLab instances. (COLLAB-6012)

fixed - Errors when converting Office documents to PDF. (COLLAB-5719)

fixed - A database conversion issue when upgrading from v. 11.3.11301 to 12.x related to certain Bitbucket Server configurations. (COLLAB-5915)

fixed - An incorrect loading order of Collaborator modules on some Unix/Linux systems that resulted in the SAML service not starting up. (COLLAB-5510)

### 12.1.12100 - August 1, 2019

**New Features**

added - Pagination controls and group search filters to Admin > Groups page. (COLLAB-5679)

added - Remote System Integrations: Decrease number of API calls for getting file content. (COLLAB-5669)

added - To diagnose issues with remote system integrations, administrators can now enable detailed logging of remote system actions 216. (COLLAB-5667)

added - Filter box was added to Action Items 333 page of Web UI. It allows quick filtering of reviews by their title, status, author and so on. (COLLAB-5074)

added - Support for SQL Server 2017 (COLLAB-4284)

**Bug Fixes**

fixed - Collaborator handles '@' symbol in usernames incorrectly. (COLLAB-5892)

fixed - Issue with processing non-formatted messages in Git log. (COLLAB-5741)

fixed - Cells with “Wrap text” enabled are not being shown correctly. (COLLAB-5723)

fixed - Email messages had incomplete URLs to target reviews. (COLLAB-5717)

fixed - GitHub integration: Getting info about file larger than 1Mb takes a lot of time (COLLAB-5711)

fixed - TFS Integration: Collaborator doesn't indicate deletion if the whole folder was deleted. (COLLAB-5673)

fixed - Collaborator treats folders as files and generates additional files for the "Download" WebUI feature (COLLAB-5672)

fixed - Remote Systems: respond '202 Accepted' to webhook immediately (COLLAB-5668)

### 12.0.12000 - July 12, 2019

**New Features**

- **Atlassian Crowd Single Sign-On**

Native support for Crowd OpenID single sign-on is now available. Read Single Sign-On 130 to learn more about single sign-on authentication and Configure Single Sign-On via Crowd OpenID 145 for detailed instructions on enabling this type of authentication. (COLLAB-4240)

- **Automatically Add Code Owners**

Now Collaborator can process CODEOWNERS files in GitHub and GitLab repositories and will add the respective code owners as participants of Collaborator reviews. (COLLAB-4880)

- **OpenJDK Support**

Collaborator now supports OpenJDK - open-source implementation of Java Development Kit. (COLLAB-4320)

- **Prompt for JDBC Driver**

When configuring database connection 79 server installer now asks a path to JDBC driver and copies it to the appropriate location. In previous versions administrators had to copy driver files manually. (COLLAB-5489)

**More improvements:**

- Repository hosting integrations now have a new option that controls whether to wait for review signature before merging pull request. (COLLAB-4649)
- To diagnose license usage, administrators can now optionally display a list of currently logged-in users 180 when a new user fails to login because there are not enough licenses. Besides they can show a list of active floating-seat users 185 on the Licensing page of Admin UI. (COLLAB-1150, COLLAB-5628)
- 6 new commands to manage a list of group templates (templates available to group members when creating new reviews) and add child groups from Command-Line Client. (COLLAB-2681)
- The JSESSIONID cookie of Web client have been renamed to JSESSIONID_COLLAB in order to avoid XSRF conflicts with other systems. (COLLAB-5653)
- The Remote System Integrations settings were divided into two sub-categories: Repository Hosting Services and Issue-Tracking Services. (COLLAB-5506)
- Web client now displays loading indicator while performing group synchronization during login (COLLAB-4967)
- Two new options that define whether to use secure login cookies 196 and whether to clear them when the browser is closed 196. (COLLAB-3698, COLLAB-3697)

**Discontinued Support**

- Collaborator server and clients no longer use XML-RPC API (deprecated since Collaborator 9.4).

  Old clients are incompatible with Collaborator server 12.0 and later. To work with it, you will need to upgrade your clients to version 12.0 or later. (COLLAB-2983)

- With the release of 12.0 we no longer ship Eclipse and RTC plug-ins built with Java 7. To integrate with Collaborator you will need to upgrade to Eclipse or RTC that use Java 8.

**Bug Fixes**

fixed - The red flags in the admin section should show up if and only if a user is actively consuming a license. If a user is not consuming a license, then there should be no flag. (COLLAB-5816)

fixed - Synchronize number of active users and the number of consumed licenses. Update last activity even if tab is out of focus. (COLLAB-5813)

fixed - Client performance penalties while loading list of existing reviews if there are a lot of in-progress reviews. (COLLAB-5641)

fixed - Exception on loading review when "logicalVersion" parameter is null. (COLLAB-5636)

fixed – Previous versions of Collaborator were allowing users to access some of Collaborator's pages without consuming a license. An initial fix was introduced in 11502 (but not documented). It is likely that Administrators will see an increase in license consumption due to this fix. (COLLAB-5628)

fixed - The default notification state for the "Send To State" user preference should be "File Activity Occurs". (COLLAB-5625)

fixed - Perforce: Using the `smartbear.ccollab.upload.ignore.binary.file` option doesn't prevent client from loading version information. (COLLAB-5618)

fixed - Remote Systems: Commits that were merged in from another feature branch could be missing in pull request review. (COLLAB-5604)

fixed - The "ccollab addchanges new ." command did not work for Git on Linux. (COLLAB-5603)

fixed - Bitbucket Cloud integration could reach rate limit of API calls. (COLLAB-5595)

fixed - Revise consuming licenses for JSON API calls (COLLAB-5583)

fixed - Support formatting and line breaks in the spreadsheet cells in DiffViewer. (COLLAB-5552)

fixed - Remote repository integrations cannot assign reviewers if non-default set of roles is used on Collaborator side. (COLLAB-5540)

fixed - The "Accepted" icon was dropped for files that were added to changeset but their content remain unchanged. (COLLAB-5273)

fixed - The update site of Eclipse plugin provides an older version of the plugin. (COLLAB-5095)

fixed - When LDAP group sync was disabled, Collaborator was still polling LDAP server for the list of groups. (COLLAB-4968)

fixed - When a review moved to the Rework phase, the "Wait until.." setting was reset to "Wait for Any" ignoring its previous value. (COLLAB-4961)

fixed - Several issues of VHDL syntax highlighting. (COLLAB-4602)

fixed - Eclipse Plug-in ignored the Due By Phase setting of Checklist custom fields. (COLLAB-4572)

fixed - Pressing the Test Connection button for an RTC configuration validates the RTC server connection even when the server has certificate errors. (COLLAB-4490)

fixed - Time spent in the Annotating phase was not included in Time metrics. (COLLAB-4477)

fixed - Exception on creating new review for new branch on Bitbucket server. (COLLAB-4460)

fixed - The LOC values for a modified file added by command line are calculated as if the file is new. (COLLAB-4454)

fixed - Performance issue while editing groups or viewing user list when there are more than 30K users (COLLAB-3911)

fixed - The GUI client cannot reliably work on Linux distributions with GTK+ 3.22. (COLLAB-3082)

fixed - Start and end dates of Content Archiving should be based on when the review was completed instead of when it was created. (COLLAB-3040)

fixed - ClearCase Remote Client: Modern CM API modules require HttpClient 4 (COLLAB-2667)

## 11.2.3 Version 11

### 11.5.11511 - May 11, 2020

fixed - Improve file metrics cache to purge invalid data when DB connection fails. (COLLAB-7228)

fixed - Diff Viewer could show empty content in Last commit display mode for changes in DOCX files. (COLLAB-6221)

fixed - Exception on opening DOCX file in Diff Viewer in Commits display mode for Git commits added in reverse order. (COLLAB-6220)

### 11.5.11510 - February 7, 2020

fixed - Users had to login twice when they had Reports page open or minimized for more than 1 hour. (COLLAB-6610, COLLAB-6289)

fixed - Report were not generated if the "Restrict Access" column was checked and the "ID" column was unchecked. (COLLAB-6364)

### 11.5.11509 - December 17, 2019

fixed - Creating a new defect with exactly the same coordinates or line number as the existing one could break reviews. (COLLAB-6431)

### 11.5.11508 - November 20, 2019

fixed - An "ambiguous column" error during "All Defects" report generation against Oracle DB (COLLAB-6235)

fixed - SQL error on Oracle DB: CLOB in WHERE clause. (COLLAB-5699)

**11.5.11507 - October 11, 2019**

fixed - Copy org.eclipse.osgi library to client plugin folder. (COLLAB-6211)

fixed - Drop-down for review subscription and template subscription doesn't work. (COLLAB-6208)

fixed - When review had multiple participants, it could take up to 30s to open a review material in Diff Viewer (COLLAB-5704)

fixed - Low performance when spreadsheet had multiple comments. (COLLAB-5682)

**11.5.11506 - October 2, 2019**

fixed - Equinox common library was updated to version 3.10. (COLLAB-6059)

**11.5.11505 - September 16, 2019**

fixed - JIRA Legacy: Missing trailing slash in review links from JIRA tickets. (COLLAB-6113)

fixed - Fixed exporting of customizable reports to different formats. (COLLAB-6059)

fixed - Improved logic for identifying if user has unread conversations. (COLLAB-6052)

fixed - Improve spreadsheet diff by comparing cell contents. (COLLAB-4800)

**11.5.11504 - July 19, 2019**

added - To diagnose license usage, administrators can now optionally display a list of active floating-seat users on the Licensing page of Admin UI. (COLLAB-5628)

fixed - The red flags in the admin section should show up if and only if a user is actively consuming a license. If a user is not consuming a license, then there should be no flag. (COLLAB-5816)

fixed - Synchronize number of active users and the number of consumed licenses. Update last activity even if tab is out of focus. (COLLAB-5813)

fixed – Previous versions of Collaborator were allowing users to access some of Collaborator's pages without consuming a license. An initial fix was introduced in 11502 (but not documented). It is likely that Administrators will see an increase in license consumption due to this fix. (COLLAB-5628)

**11.5.11503 - May 15, 2019**

added - JIRA integrations will use API tokens instead of passwords. Atlassian has deprecated password-based authentication and suggests using API tokens. To use JIRA integrations your administrator will need to generate JIRA API token and specify it in issue-tracking configuration. See Configuring JIRA Integration 897 for instructions. (COLLAB-5642)

fixed - The ccollab addchanges command failed for Git on *nix platforms (COLLAB-5603)

fixed - While making single review dump cut off data not connected with this review (COLLAB-4539)

fixed - RTC changesets linked to workitem and located in stream are lost during review creation (COLLAB-4397)

fixed - If a user had more than 1000 open reviews, home page failed to show list of reviews on SQL Server 2016 database (COLLAB-4054)

fixed - Do not enable new Automatic links in templates automatically (COLLAB-3926)

fixed - Issues with width of Files column in WEB UI (COLLAB-3361)

**11.5.11502 - April 16, 2019**

added - Bitbucket Integration 826 has migrated to Bitbucket API 2.0. Bitbucket will deprecate their API 1.0 starting from April 29. You will need to upgrade to Collaborator version 11.5.11502 to continue using integration with Bitbucket. (COLLAB-5520)

added - The "Allow to reopen review" option of remote repository integrations was renamed to "Reopen review when" and now allows to control what events will reopen completed reviews. (COLLAB-4830)

fixed - The GUI client picks an incorrect shelveset after the shelvesets are sorted by date (COLLAB-5558)

fixed - In some cases Collaborator did not release license on logout (COLLAB-5529)

fixed - The comment/code change arrows are grayed out when you first open the diff viewer. (COLLAB-5485)

fixed - The update site of Eclipse plugin provides an older version of the plugin (COLLAB-5095)

fixed - NullPointerException on adding diffs from CVS repository (COLLAB-2280)

**11.5.11501 - March 26, 2019**

added - Update company and product logos in server and clients. (COLLAB-5099)

added - New admin setting that defines the format of the NameIDPolicy parameter 139 in SAML authentication requests. (COLLAB-4505)

fixed - NullPointerException when loading active reviews after migrating from Windows to Linux server. (COLLAB-5490)

fixed - Eclipse: Cannot open files in content merge viewers of RDi V9.6 (COLLAB-5278)

fixed - Invalid GUID message displayed on attempt to select a group to which current user does not belong (COLLAB-5266)

fixed - Diff Viewer ignores newline characters when rendering cell contents (COLLAB-4906)

fixed - Renamed/moved files in TFS are not marked as deleted in Collaborator (COLLAB-4893)

fixed - Filter "Show only logged-in users" does not work when there are more than 100 users (COLLAB-4843)

fixed - Collaborator does not send the "phase - changed" notification in certain scenarios (COLLAB-4745)

fixed - Visual Studio plugin reverses the revisions in the DiffViewer (COLLAB-4102)

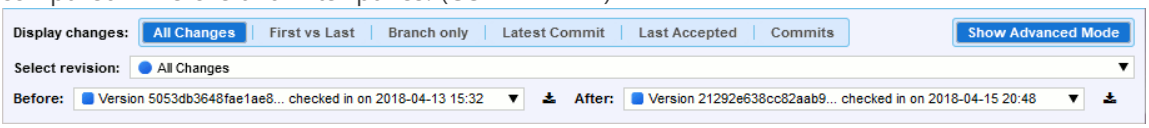
fixed - GUI Client does not sort shelvesets alphabetically (COLLAB-3093)

fixed - DiffViewer: URL link contains '&#8203' character (COLLAB-2675)

### 11.5.11500 - March 1, 2019

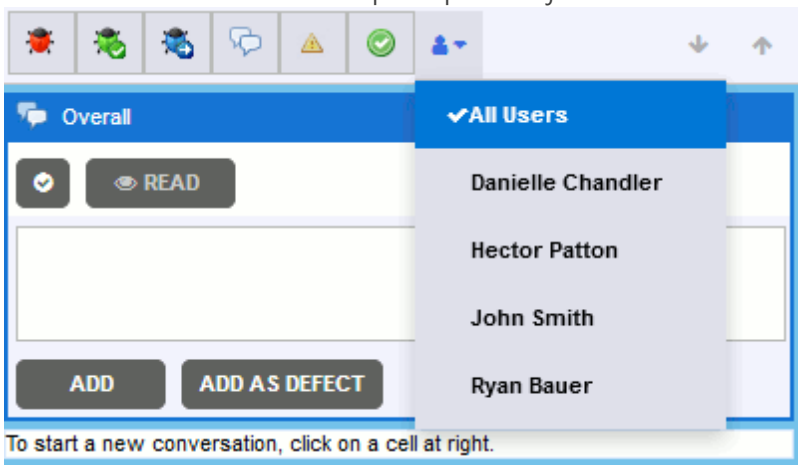
**New Features:**

- Now the **Display changes** selector and **Select revision** drop-down are always visible. Additionally, you can enable Advanced mode which allows manual selection of revisions to be compared in Before and After panes. (COLLAB-4414)




- Diff Viewer contents now synchronizes with the Overlay / Separate mode of Review Screen. In Overlay mode of review materials, Diff Viewer will display overall changes made to the file as specified by the *Default Revision Comparison of Diff Viewer* 324 setting, while in Separate mode it will display changes made by a particular changeset. (COLLAB-4975)

- Diff Viewer's Chat section now has a new user-list filter that specifies whether to display comments and defects from all participants of just from the selected participant. (COLLAB-1127)




- Existing review participants will be retained upon changing group if they belong to new group. (COLLAB-5246)

- Existing values of custom fields will be retained upon changing review template if the custom field exists in both templates. (COLLAB-3815)

- Adding Git commits from GUI Client, Command-Line Client, Eclipse Plug-in or Visual Studio Extension will also add part of Git log information to build properly ordered list of changes on Collaborator server side. This will ensure that in Review Screen and Diff Viewer Git commits are displayed in the same order as in the Git log. To use this functionality you will need to update both server and clients. (COLLAB-4997, COLLAB-2717, COLLAB-1897)

- Review Screen, Diff Viewer, Eclipse Plug-in and Visual Studio Extension now display Subversion, Perforce, TFS and RTC atomic changelists in chronological order (from older to newer), regardless the order in which they have been uploaded to review. (COLLAB-4002, COLLAB-3980, COLLAB-2717, COLLAB-1897)

- New "Automatically create new groups" setting defines whether to create new groups automatically during LDAP/AD group synchronization or just map users to existing groups. Another new setting "Automatic group creation filter" specifies what exactly groups should be created automatically: if group FQDN matches the specified regular expression it will be created. Otherwise it will be excluded. (COLLAB-4238, COLLAB-4239)

- Ability to use Oracle Text component instead of regular expression query on Oracle databases 67 for full-text search from Web Client. This component is disabled by default, administrators can enable it on the Oracle side or through Collaborator admin UI. (COLLAB-5356)

- New Default Scale for Documents in Diff Viewer 326 user setting which specifies the default zoom level when reviewing documents. (COLLAB-3862)

- New Default sent to state 321 and Default wait state 320 user settings that allow specifying the default notification level for Sent To and Wait actions. Administrators can also specify these settings for the entire group via command-line and JSON API. (COLLAB-2911)

**Bug Fixes**

- fixed - TrayNotifier was able to login back after logging out manually (COLLAB-5214)
- fixed - Sometimes reviews containing group pool participants and custom participant fields do not open (COLLAB-5157)
- fixed - Upgrading from 11.3 to 11.4 on Oracle database did not clear existing indexes (COLLAB-5122)
- fixed - The "First vs Last" option may work incorrectly for Perforce (COLLAB-5121)
- fixed - Visual Studio Extension: Invalid .vsix certificate (COLLAB-5026)
- fixed - MacOS Client installer for 11.4.401 -11.4.402 was reported as damaged. (COLLAB-5008)
- fixed - Participant custom field validation should ignore values of other participants (COLLAB-5001, COLLAB-5232)
- fixed - Error while parsing LDAP group attributes of binary type (COLLAB-4990)
- fixed - Several issues caused by incorrect data caching in web browsers. (COLLAB-4976)
- fixed - The Accept button is disabled when uploading arbitrary diff (COLLAB-4964)
- fixed - In certain cases, file comments and defects added in "All changes" mode were not displayed in the Review Screen (COLLAB-4949)
- fixed - File renames corrupt review with Bitbucket integration (COLLAB-4947)
- fixed - Need to support TFS server workspaces in addition to local workspaces (COLLAB-4939)
- fixed - Cannot accept file changes if some of them were cherry-picked from another branch (COLLAB-4899)
- fixed - User list not displaying on Oracle 11g, Oracle 12c and MS SQL 2008 (COLLAB-4896)
- fixed - DiffViewer highlights the entire latest uploaded reversion instead of showing the actual differences in "All changes" mode (COLLAB-4889, COLLAB-4953)
- fixed - Unable to export reports to PDF or XLS (COLLAB-4866)
- fixed - In certain cases, DiffViewer did not display all actual changes unless "All changes" mode was selected (COLLAB-4810)
- fixed - Add C language schema for syntax highlighting (COLLAB-4797)
- fixed - Incorrect/empty patterns in highlighting schemas cause serious performance penalties (COLLAB-4720)

- fixed - Clicking on an existing push-pin could create a new chat instead of selecting an existing one. (COLLAB-4693)
- fixed - Collaborator fails to add comments to defect conversations which have [0,0] coordinates (COLLAB-4690)
- fixed - Collaborator fails to move broken pins, but copies them (COLLAB-4688)
- fixed - The "un-clickable" cross sign icon is shown in the "Search" field in Internet Explorer (COLLAB-4676)
- fixed - GitHub, GitLab and Bitbucket webhooks could not access Collaborator server if single sign-on was enabled (COLLAB-4662)
- fixed - Incomplete tool-tip text for "Completing the rework phase" status (COLLAB-4657)
- fixed - Cannot hide the System-wide message in the Diff Viewer (COLLAB-4655)
- fixed - Refresh action after we return back to Review Screen from DiffViewer (COLLAB-4652)
- fixed - The "Go to next location" and "Go to previous location" buttons were enabled even before the PDF page/ image was fully loaded (COLLAB-4639)
- fixed - Exception occur during Reviews by Changelist report generation (COLLAB-4638)
- fixed - Status checks on regexp branches are not working (COLLAB-4621)
- fixed - DiffViewer may fail to display file changes if letter case in file-path was changed. (COLLAB-4616)
- fixed - Some issues in VHDL highlighting schema (COLLAB-4602)
- fixed - GitLab integration: Newly added files were not uploaded to review when the main branch was behind the develop branch. (COLLAB-4583)
- fixed - Pull Requests: show changes if they merged in from another feature branch (COLLAB-4525)
- fixed - ClearCase "AddVersion" command uploads only one version even when multiple versions were specified. (COLLAB-4314)
- fixed - Tray Notifier could lock out the user's account in Active Directory if incorrect credentials were specified (COLLAB-4168)
- fixed - The participant custom field should be locked from modifying by other participants (COLLAB-4059)
- fixed - Perforce: Problem with triggering files marked for adding by ensure-diffs-reviewed (COLLAB-3914)

- fixed - GUI Client "Add changes" command for Subversion ignored the selected files at first attempt to upload files (COLLAB-3609)
- fixed - Visual Studio Extension could lock out the user's account if the credentials have been changed on server-side while working in IDE (COLLAB-3371)
- fixed - Diff Viewer: Cannot select text within commented lines (COLLAB-2757)
- fixed - Comment promotion failing to detect correct line in new version (COLLAB-1382)

**Discontinued Support**

With the release of Collaborator 11.5, GitHub Polling integration becomes deprecated. This type of integration implies polling the GitHub server at regular time intervals to retrieve information about changes in repositories. This approach has certain time lag and has become obsolete now. We highly recommend using webhook-based GitHub integration type instead of GitHub Polling. Transfer prompt will be displayed on the Remote System Integration page of Admin UI.

### 11.4.11405 - March 15, 2019

added - Ability to use Oracle Text component instead of regular expression query on Oracle databases[67] for full-text search from Web Client. This component is disabled by default, administrators can enable it on the Oracle side or through Collaborator admin UI. (COLLAB-5356)

fixed - Options for Oracle text search is shown for any database. (COLLAB-5387)

fixed - The SearchSettingsActivity should check if a customer has unsaved changes. (COLLAB-5415)

fixed - Check whether Oracle Text component is available before search. (COLLAB-5425)

fixed - Do not create Oracle CTX indices automatically during DB upgrade or fresh installation. (COLLAB-5420, COLLAB-5442)

fixed - Oracle search settings saving was fixed. (COLLAB-5421)

### 11.4.11404 - December 3, 2018

added - Implement VM options for pre-defined Oracle search scopes (COLLAB-5015)

fixed - The participant custom field should be locked from modifying by other participants (COLLAB-4059)

fixed - Participant custom field validation should ignore values of other participants (COLLAB-5001)

fixed - User Login Prompt is displayed only in a single line (COLLAB-4988)

fixed - Visual Studio Extension: Invalid .vsix certificate (COLLAB-5026)

#### 11.4.11403 - November 14, 2018

added - A Java VM option to suppress revision order check-up and enable the Accept button for diff uploads. (CC-15927, COLLAB-4964)

fixed - User list not displaying after upgrade to 11.4.11400 on Oracle 11g, Oracle 12c and MS SQL 2008. (CC-15687)

#### 11.4.11402 - October 12, 2018

fixed - Collaborator ignored the content-cache setting in the Tomcat configuration file. (CC-15676)

#### 11.4.11401 - October 10, 2018

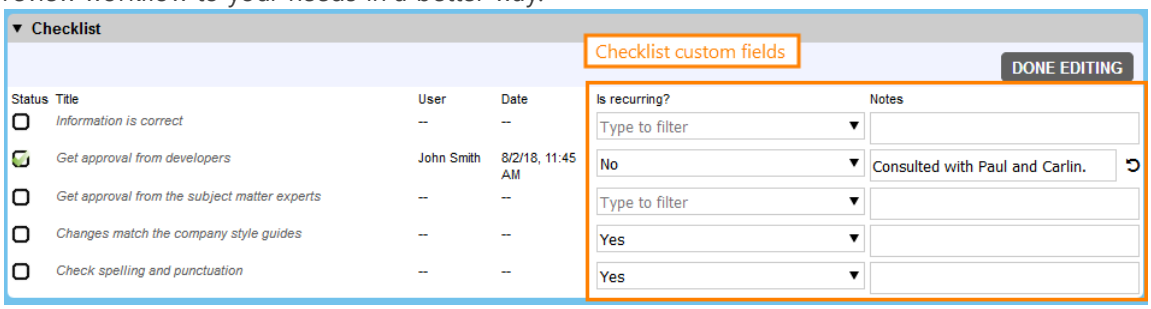
fixed - In certain cases, Collaborator failed to export reports in the PDF and Excel formats. (COLLAB-4866)

### 11.4.11400 - September 26, 2018

**New Features**

- **Smarter support for pull requests and rebased or squashed commits.** After a review for some pull request was created, you or your teammates continue working with your source control and continue committing changes. It is quite possible that some commits mentioned in the pull request could be rebased or several commits could be squashed into one commit. The new version of Collaborator now tracks all these changes and automatically includes or excludes commits from the review, keeping the review materials actual. This saves your time and helps you concentrate on reviewing, not on maintaining the list of materials up-to-date.
- **Support for Git submodules.** Repository hosting integrations 826 can display changes in the submodules of tracked Git repositories.
- **Collaborator Plug-in for Bitbucket Server is no longer needed.** Starting from version 5.4 Bitbucket Server have built-in support for webhooks, so Collaborator can interact with remote repositories without this helper plug-in.

- **Custom fields in checklists.** Now you can use custom fields[256] in checklists[272] to adjust the review workflow to your needs in a better way.




- **Checklist events in the timeline.** The review timeline now includes checklist events like selection or clearing checklist items. This helps you better understand the actions review participants performed during reviewing.

- **Importing group membership from LDAP/AD servers to Collaborator for SSO logins.** If a user logs in to Collaborator using SSO, then Collaborator will search for the user name in your LDAP/Active Directory server and will automatically import groups, to which that user belongs. This feature can significantly save time needed to create user groups. The effect is especially noticeable if you have lots of users in your organization. See "Synching User Group Membership with LDAP/Active Directory[143]" for details.

- **Support for non-default Tomcat configurations.** Earlier versions of Collaborator considered that the Collaborator Server URL consists of the host name only and has no additional part in it. So, for example, the Recent Review menu items used links like `/ui#review:id=123` to open reviews. This caused problems, if the Collaborator Server's URL was like `http://<site>/some-additional-path/`: the page's full path was like `http://<site>/`*`some-additional-path`*`/ui#review:id=123`, and the menu commands did not work. Collaborator 11.4.11400 supports additional paths in the Server URL, and the following server URL, for example, works fine now: `http://<my-site>/collab/`.

- **Improved server security.** We have improved internal algorithms in Collaborator Server and eliminated vulnerabilities of the following types:

  - Slow HTTP headers (for GET and POST requests)
  - Cross-site request forgery
  - Unprotected directory
  - Autocompletion for sensitive forms

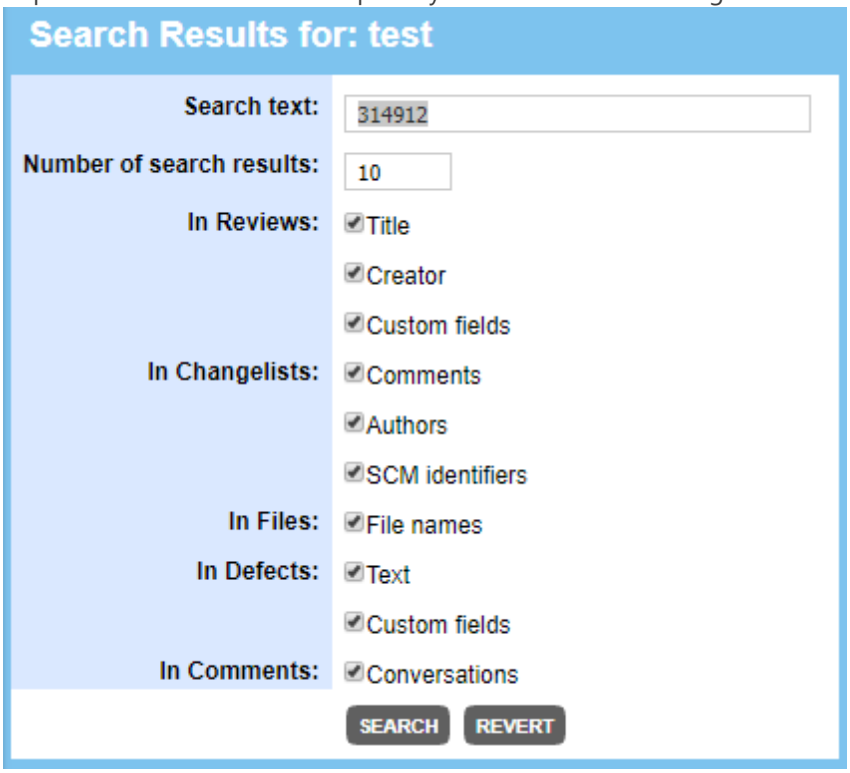
- **Minor releases compatibility.** Earlier, we recommended that you use the Collaborator Server and Client of exactly the same ***major.minor.build*** versions. We have changed that requirement, and now all matching ***major.minor*** versions of Servers and Clients can work together, regardless of the ***build*** version.

- **Adding URLs to reviews.** You can use the **Upload > URL** command of the Review Materials menu to append URLs to reviews easily.

- **Git LFS Support for GUI and command-line clients.** Git Large File Storage (LFS) replaces large files with text references to the files and links those to Git repository, while the file contents is stored on another server. Collaborator GUI and command-line clients now can fetch the actual files on local machine by their references and upload those files to Collaborator reviews.

- **Improved search from command-line.** The `ccollab admin find review` command now can search for the specified text within review titles, user comments and custom fields.

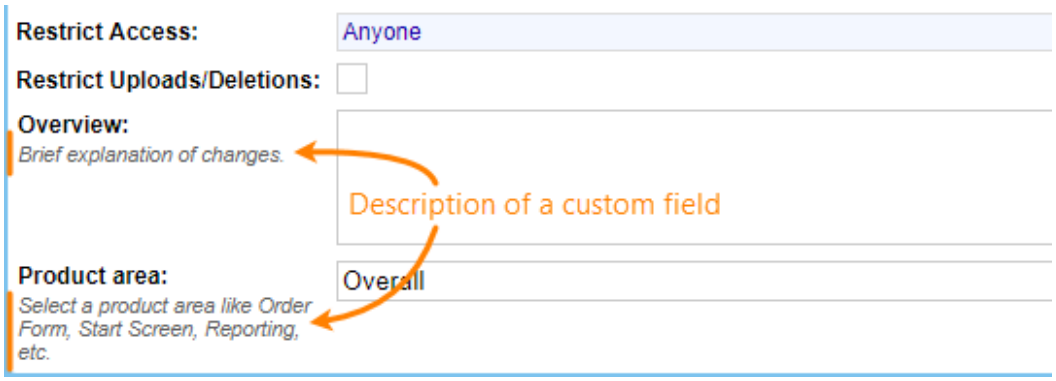
- **Improved search on Oracle database.** On Oracle databases, Collaborator will no longer search the contents of custom fields by default (since this significantly reduces search performance). Instead, the search results page display additional fields that define in what areas to perform new search. In this panel you can enable searching in custom fields.




- **Updated Login page** now contains links to the "Terms of Use" and "Privacy Policy" pages.

- **UI improvements:**

  - **Clearer custom fields' view.** Custom fields' descriptions are now displayed under the field labels, not under the field values:




- **Hiding Remote Links section.** If you do not use issue-tracker integrations or remote repository integrations, the Remote System Links section of the Review Summary Screen does not make much sense. Now administrators can control whether to display or hide this section via review template settings.
- **User Lists.** Added pagination controls and user search filters to Admin | Users page.
- **Clearer UI.** Version 11.4.14000 uses a slightly larger distance between paragraphs and sections, making it easier to read text in the General Info and Release Materials sections on the review screen.
- **Shorter commit messages.** The Changelist info section of Review Summary screen now displays only the first 70 chars of commit message by default and an ellipses button which shows the entire message.
- **Clearer reports.** Earlier, if some reviewer found an issue and clicked "Send to Rework", Collaborator displayed the "Send to Rework" string for that reviewer in reports. This might cause confusion in cases, when your workflow and Collaborator settings allow reviewers to complete a review even if some other reviewer sent it to rework. To avoid confusion in reports, in version 11.4.14000 we have changed the "Send to Rework" string to "Approved (by workflow)" in reports.

### Bug Fixes

- fixed - Sometimes, the most recent commit was not added to review because of too short pull-request delay of Bitbucket integration. The default delay value was increased, and became configurable via VM option. (CC-14194)
- fixed - After upgrading from a legacy Collaborator version, Diff Viewer does not open for some reviews. (COLLAB-4673)
- fixed - Collaborator fails to load document in the Diff Viewer if pushpins are invalid. (COLLAB-4659)

- fixed - If .doc, .docx, .pdf file types are added to the "Binary file types" list, Collaborator fails to open the files in the Diff Viewer and shows the "Cannot read property" error. (COLLAB-4648)
- fixed - Optimize Jira API calls while getting ticket summary and status (COLLAB-4629)
- fixed - The "Existing authentication configuration" setting was disabled in installer wizard. (COLLAB-4618)
- fixed - Review creation date could change while modifying other review fields (COLLAB-4556)
- fixed - When JIRA projects had coincident names (like, MYPRJ1 and PRJ1), Collaborator could append extra links to JIRA tickts to reviews. (COLLAB-4550, COLLAB-4321)
- fixed - The jackson-databind component was updated (COLLAB-4546)
- fixed - The Customizable Defect Reports did not return any results if the "Report Access" setting was set to "Respect permissions" (COLLAB-4543)
- fixed - Uninformative error message was displayed by Eclipse plugin when uploading files while Collaborator server connection was not configured yet. (COLLAB-4481)
- fixed - The Review Materials list displayed commits in different orders in the Overlay and Separate modes. (COLLAB-4394)
- fixed - the gated check-in trigger now supports TFS 2017 and 2018 (COLLAB-4392)
- fixed - The contents of review screen sections in Web UI were hard to read in edit mode. (COLLAB-4372)
- fixed - When the revision displayed in After pane was predating the revision displayed in Before pane, the Accept button was disabled without any explanation. Now the Diff Viewer shows a tooltip that explains why the button is disabled. (COLLAB-4366)
- fixed - GitHub comments were attached to wrong source lines in reviews. (COLLAB-4345)
- fixed - Issued could occur on Collaborator server deployed into a non-root location of Tomcat (COLLAB-4309)
- fixed - The user name was not displayed in footers of administrator and reports pages (COLLAB-4307)
- fixed - Custom Fields' descriptions were appended to content's field (COLLAB-4305)
- fixed - Update GitLab API. Old API became deprecated and integration stop working. (COLLAB-4301)
- fixed - In certain cases, Collaborator failed adding a user from a group to a review. (COLLAB-4272)

- fixed - In certain cases, an error occurred when you scrolled the DiffViewer window with keys. (COLLAB-4266)
- fixed - The "User List" report generated empty CSV file (COLLAB-4259)
- fixed - Collaborator fails to upload the .doc/.docx file that aren't checked out in ClearCase via the "Add Activities" mode of GUI client. (COLLAB-4244)
- fixed - Sometimes coordinate comment pairs on Before and After panes were positioned incorrectly. (COLLAB-4243)
- fixed - File subscriptions now can include repository name in file path (COLLAB-4227)
- fixed - The caption of the Defects section did not change according to the value specified in the "Defects Label (plural)" option. (COLLAB-4204)
- fixed - Temporary files were cleaned up only when Collaborator server was stopped. (COLLAB-4195)
- fixed - For mono type fonts Diff Viewer had problems with alignment of changed lines. (COLLAB-4190)
- fixed - Report results did not wrap review ID into hyperlinks to those reviews. (COLLAB-4188)
- fixed - The Defect-Custom-Fields' default values were not selected when creating defects (COLLAB-4177)
- fixed - The "User Login Prompt" was not displayed on the Login page. (COLLAB-4169)
- fixed - If the Due By Phase setting is selected for custom field, it should force users to specify a value even when the Validator setting is blank. (COLLAB-4166)
- fixed - Connections established by JIRA or repository hosting service 826 integrations (except for GitLab) now respect proxy settings 89 if they are specified for Collaborator server. (COLLAB-4085)
- fixed - The Accepted check-mark was not cleared when uploading a new revision of the file. (COLLAB-4025)
- fixed - Reviews created on pull requests could display incorrect differences after the original branch was merged and subsequent file revisions were uploaded. (COLLAB-3982)
- fixed - Sometimes, in reviews created on pull requests comments were displayed at wrong lines (COLLAB-3981)
- fixed - The Participants category of the Review Summary page doesn't show the Sent to Rework group and its users if the Group by State option is set. (COLLAB-3974)
- fixed - The Action Items list does not display the participant's role for users that were added via review subscriptions. (COLLAB-3950)

- fixed - Collaborator built into Visual Studio did not allow review participants to edit participant custom fields of other review participants. (COLLAB-3888)
- fixed - Some of the JSON API methods could run without the user authentication. (COLLAB-3857)
- fixed - Bitbucket Server integration could create two reviews for the same pull request. (COLLAB-3849)
- fixed - When opening a file with only single change, Diff Viewer could spontaneously switch from Expert to Simple view mode. (COLLAB-3840)
- fixed - Deleted files in the TFS changeset were not marked as "deleted" in the Review Materials section (COLLAB-3801)
- fixed - If some changes have been made both in feature branch and in upstream branch (for example, by cherry-picking), Bitbucket Server integration now mimics the behaviour of the original Bitbucket Server and ignores those simultaneous changes in Branch Only mode of DiffViewer. (COLLAB-3717)
- fixed - ClearCase integration created reviews in some custom way and didn't calculate LOC metrics. (COLLAB-3671)
- fixed - Checklist items now can contain up to 1024 characters (COLLAB-3649)
- fixed - The Login page does not prevent the browser from saving the user name and password. (COLLAB-3582)
- fixed - Updates to pull request does not reopen Collaborator review, potentially blocking merge (COLLAB-3560, COLLAB-3219)
- fixed - The Accept button was absent in Visual Studio's Diff Viewer unless at least one comment was added. (COLLAB-3485)
- fixed - GitLab integration failed to fetch subgroups and associated projects if the group name was specified (COLLAB-3232)
- fixed - Performance issues in Users page of Administrator settings when displaying more than 30K users. (COLLAB-3110)
- fixed - If commits containing changes to a file were uploaded as part of the diff range and then additional commits were uploaded to the same review and contain reverting the changes to the file, the file still appeared in the modified state in the Collaborator review (COLLAB-2737)
- fixed - Multiple tray icons could be displayed on Windows 10 (COLLAB-2691)
- fixed - The “accepted/chatting ” check-mark automatically changed to "accepted" for user who has uploaded the changes (COLLAB-1488)

- fixed - When file name was matching a pattern specified for Automatic links, Collaborator applied Automatic links rather than file links (COLLAB-1415)

- fixed - The "Go to next/previous location" buttons became available before the document page was fully loaded and stuck Diff Viewer. (COLLAB-899)

## 11.3.11302 - March 28, 2019

fixed - In some cases Collaborator did not release license on logout (COLLAB-5529)

### 11.3.11301 - May 16, 2018

fixed - Defect-Custom-Fields' default values are not selected when creating defects (COLLAB-4177)

fixed - User Login Prompt  is not shown on the "Login" page (COLLAB-4169)

fixed - Collaborator Community Edition Shows a "This version of the product doesn't support external authentication" message (COLLAB-4161)

## 11.3.11300 - April 24, 2018

added - **Modern WebUI.** User interface of Web client 312 have been significantly improved. Home Screen, Review Summary Screen, Diff Viewer and other pages were reworked to produce nicer look and feel and produce greater usability (COLLAB-3193)

added - **WebUI themes.** Several pre-defined UI themes that control the visual style of Web client 312. Administrators may specify the default WebUI theme 185 for the entire server, while regular users can select their own preferred WebUI theme 322 in User Preferences. (COLLAB-3316)

added - **Simplified revision selection in Diff Viewer.** New mode provides easier way to select which revisions should be compared in Diff Viewer 369. (COLLAB-3301)

added - **Creating bugs in JIRA issue-tracker.** Now, you can create tickets in your JIRA issue-tracker 887 directly from the Collaborator reviews. (COLLAB-2908)

added - **Automatic user and group synchronization with LDAP and Active Directory.** 239 Administrators now can configure user and group mapping between Collaborator and the LDAP directory or Active Directory. In this case, Collaborator will retrieve user properties (name, phone, email, and so forth) and their membership in groups when the users login.  If the user or group does not exist on server, Collaborator will create it automatically. (COLLAB-1238)

added - **RTC support in Visual Studio Extension**. Now you can create reviews for RTC changesets and work items directly from Visual Studio  (COLLAB-2610)

added - **Document review enhancements.** Collaborator now supports reviewing Microsoft PowerPoint presentations 395, Microsoft Visio graphics 400, as well as documents, spreadsheets and presentations in OpenDocument format. (COLLAB-3227)

> **Important:** Document reviews (except for text-based and image files) is only supported on 64-bit Collaborator servers. On 32-bit platforms, Collaborator may fail to process the documents due to insufficient memory.

added - **Wildcard and regexp support in branch names.** The "Branches to track" and the "Ignore pushes for branches" settings of repository hosting configurations 840 now accept the "*" wildcard to match all branches and regular expression to match specific branch name patterns. (COLLAB-3203)

added - Command-line and JSON API commands to manage group subscriptions (COLLAB-2800)

added - Validate support for Java 9 (COLLAB-3186)

added - New `--notification-level` parameter of the `ccollab admin user edit` and `ccollab admin group edit` commands allows managing e-mail notifications on user and group levels. (COLLAB-3111)

added - Validate support for Microsoft SQL Server 2016 (COLLAB-2683)

added - Email notification template for the review deadline change event (COOLLAB-2521)

added - Validate support for Windows Server 2016 (COLLAB-2460)

added- Search-as-you-type functionality for drop-down list 266 custom fields. That is, you may type-in first few letters of item name to narrow down the list of available items. (COLLAB-1833)

fixed - Cross-site scripting vulnerability of Log-in page (COLLAB-3696)

fixed - Excel: Need to support colors and text formatting (COLLAB-3624, COLLAB-2352)

fixed - Perforce addchangelist new throws error if "Overview" custom field is disabled. (COLLAB-3606)

fixed - Add warning to Eclipse when changing templates (COLLAB-3600)

fixed - Need to implement additional logic for deleted files in history (COLLAB-3577)

fixed - Remote Integrations: changes lost while processing merge commit with specific history (COLLAB-3563)

fixed - Cannot add materials from Visual Studio plugin when using collection URL with a space in the name (COLLAB-3554)

fixed - Team Edition: Full reporting functionality (COLLAB-3447)

fixed - Truly disable news with -Dsmartbear.news.disable=true (COLLAB-3434)

fixed - Fix SSO Assertions, etc for NASA (COLLAB-3381)

fixed - Chat links do not recognize filenames correctly (COLLAB-3360)

fixed - Typo in Visual Studio plugin "chagesets" (COLLAB-3356)

fixed - Eclipse client cannot open a perspecive (COLLAB-3343)

fixed - Java 9: We ignore Java 9 on machine (COLLAB-3287)

fixed - The "Respect permissions" value of "Report access" setting was ignored (COLLAB-3281)

fixed - Put the defect ID between the bug icon and the defect description (COLLAB-3280)

fixed - Re-position screen when adding defect so Add Defect box does not get clipped (COLLAB-3277)

fixed - The completed Review is not shown in "Completed" in the Home Page (COLLAB-3248, COLLAB-3183)

fixed - Diff Viewer displayed collapsed references in documents as expanded (COLLAB-3202)

fixed - Insecure request redirect (COLLAB-3160)

fixed - Visual Studio plug-in: We should use only one user for searching TFS shelvesets (COLLAB-3112, COLLAB-3531)

fixed - Perforce: Content triggers should ignore BOM in files (COLLAB-3074)

fixed - Handle uncaught JavaScript errors (COLLAB-3072)

fixed - Diff Viewer: Word file which contains charts cannot be opened (COLLAB-3036)

fixed - Cannot add changelist because of configpath issue in PTC MKS integrity (COLLAB-2992)

fixed - GUI Client crashes on Linux (COLLAB-2979)

fixed - Diff Viewer is not displaying the line numbers correctly. (COLLAB-2963)

fixed - Server can still set malformed V6_CLIENT_LOGIN_TICKET_HEADER (COLLAB-2962)

fixed - Eclispe plugin does not work if Subversive plugin is installed (COLLAB-2948)

fixed - Syntax Highlighting: The "New Pattern" command modifies existing pattern (COLLAB-2927)

fixed - JSON API: Problem with processing document conversion errors (COLLAB-2878)

fixed - Permanent descriptions for custom fields (COLLAB-2867)

fixed - Sometimes comments are marked as read in Diff Viewer and as unread in Review Summary. (COLLAB-2835)

fixed - Tray Notifier requires an access to the cacerts file to establish HTTPS connections (COLLAB-2756)

fixed - "Untitled Review" Notification sent by the system to author/creator of the review (COLLAB-2744)

fixed - Reports: 'CSV' retrieves data which does not belong to a report (COLLAB-2734)

fixed - Minimizing Diff Viewer window should not hide its buttons (COLLAB-2656)

fixed - File subscriber who does not belong to a group is added to a review having "Group Based" access restriction. (COLLAB-2587)

fixed - Review Custom Fields: Drop-down series limited by 51200 symbols (COLLAB-2552)

fixed - Defect custom fields are not aligned and resizeable (COLLAB-2547)

fixed - Automatic links of URLs do not convert trailing slashes (COLLAB-2530)

fixed - User receives notification when P4V creates a review even with 'Minimal' notification level. (COLLAB-2514)

fixed - javax.net.ssl.SSLHandshakeException if you connect to https collaborator server (COLLAB-2508)

fixed - Improper "Review complete" notification because "Accepted" comments were treated as unread comments. (COLLAB-2457)

fixed - iCal Everyone: Validator for Start/End date/time should be added (COLLAB-2448)

fixed - Spreadsheets: Merged cells are unreadable in Diff Viewer (COLLAB-2088)

fixed - Collaborator cannot convert certain types of images embedded into Word document (COLLAB-1858)

fixed - Eclipse 4.5: Mouse scrolling does not work in editor windows due to Focus change (COLLAB-1290)

fixed - Now reviewers in "waiting" state will receive notifications of other participant/author activity during the Rework phase, by default. (in previous versions they did not). New VM option was added that controls whether "waiting" reviewers will get notifications in this case or not. (COLLAB-1063)

**Discontinued Support**

With the release of Collaborator 11.3, we will no longer support Microsoft Internet Explorer versions 8.x. and 9.x. Besides, the Eclipse Plug-in will no longer support integration via the Subversive plug-in versions 3.x.

## 11.2.11201 - November 24, 2017

added - In-product chat 33. To help trial users get started with the product faster and shorten the learning period, we added a new Chat panel to Collaborator. Trial users can speak with SmartBear engineers here. To start a conversation, click the Chat button in the bottom-right corner of Collaborator Web Interface and enter your message. (COLLAB-3399)

fixed - Fixed integration with GitLab v10.1.1 (COLLAB-3424)

fixed - Fix BitBucket integration for non-standard domain root/context path (COLLAB-3421)

fixed - Fix upgrade issue with bad pushpin data (COLLAB-3350)

## 11.2.11200 - October 10, 2017

added - Native SAMLv2 support for the web interface is now available. Read Single Sign-On 130 to learn more about single sign-on authentication and Configuring SSO via SAML 132 for detailed instructions on enabling SAML single sign-on. (COLLAB-1795)

added - TFS Work Items Integration 887. Now Collaborator can update reviews with links to Team Foundation Server work items associated with reviewed changesets/shelvesets. And vice versa, it can update TFS work items with links to reviews on corresponding changeset/shelveset. (COLLAB-3119)

added - Visual Studio Extension now share SCM configurations with GUI Client and command-line client. You can create new configurations and modify existing SCM configurations in the Options dialog 569. (COLLAB-3032)

added - The Getting Started wizard is now displayed when you open the Collaborator View window and the Visual Studio Extension is not yet configured. (COLLAB-2995)

added - The Remote System Links section was added to the Review Summary Screen 591 of Visual Studio Extension and to the Review Editor 538 of Eclipse Plug-in. This section lists all pull requests, direct pushes and issue-tracking items associated with the review, as well as the current status of those items. (COLLAB-2958, COLLAB-2959)

added - Visual Studio Extension now uses the C# TFS SDK to interact with Team Foundation Server. You can now retrieve a list of changesets and shelvesets 581 and select them visually, rather that entering changeset IDs by hand. (COLLAB-2607, COLLAB-2608)

added - Support for Microsoft SQL Server 2016 (COLLAB-2683)

added - Debug logger was added to Visual Studio Extension (COLLAB-3050)

fixed - RTC plugin for Visual Studio created corrupted links to reviews (COLLAB-3185)

fixed - Review Pools did not receive email notifications when minimum number of participants was reached (COLLAB-3170)

fixed - The Select Team Project drop-down list of GUI Client was not sorted (COLLAB-3169)

fixed - Reviews were not created for the Bitbucket cloud repositories when the repository owner was a team (COLLAB-3163)

fixed - Disabled review templates are still available for selection in Eclipse (COLLAB-3142)

fixed - 'My Role' was not shown for 'Completed' reviews on home page (COLLAB-3128)

fixed - Web client displays a blank page after saving defect custom fields (COLLAB-3113)

fixed - Typo in Review Detail Report (COLLAB-3052)

fixed - Defect Log section contains two similar 'Date' columns (COLLAB-3035)

fixed - Visual Studio extension did not show the user name if the "Full name" field was not filled. (COLLAB-2981)

fixed - Bitbucket pull requests accumulate irrelevant build statuses from multiple reviews. (COLLAB-2915)

fixed - The "Save" button was disabled on the "Review custom fields" page in the Collaborator Team edition. (COLLAB-2762)

fixed - The "Orientation" and "Display order settings" were not saved (COLLAB-2711)

fixed - Member of review pool should have access to a review (COLLAB-2686)

fixed - Text wrapping issue in comments on the DiffViewer (COLLAB-2636)

fixed - Chat panel sometimes disappears in DiffViewer (COLLAB-1972)

**Discontinued Support**

With the release of Collaborator 11.2, we will no longer support the SOAP API. It was deprecated in version 9.4 and has now been removed.

## 11.1.11103 - November 24, 2017

fixed - Fix upgrade issue with bad pushpin data (COLLAB-3350)

fixed - Fix open redirects (COLLAB-3350)

## 11.1.11101 - August 13, 2017

fixed - DiffViewer: Highlighting is not in line with a content (COLLAB-3024)

## 11.1.11100 - August 1, 2017

fixed - During server upgrade, the installation wizard now verifies Java Database Connectivity (JDBC) drivers that your server uses to connect to MySQL, SQL Server or Oracle databases. If these are legacy JDBC drivers, the wizard removes them. In this case you may need to download and install new drivers as described in Database Installation 59 sections. (COLLAB-2467)

added - Template Subscriptions 328 - Users can automatically become participants in reviews having a particular template. (COLLAB-2377)

added - Author-based subscriptions 233 and file subscriptions 233 for Review Pools - You can now subscribe a review pool group to reviews of some particular author as well as to reviews where some particular files are reviewed. (COLLAB-1537, COLLAB-2838)

added - Ability to archive 299 review materials that are no longer used in active reviews. This will allow to periodically reduce the size of server's content cache folder which stores all review materials. (COLLAB-1228)

added - You may now create a new review by copying any of existing reviews. This approach allows to retain some data (participant list, custom fields values, review materials) from the original review. See Copying Previous Review 450 for details. (COLLAB-2312)

added - JIRA Integration 887 now allows to associate issues with reviews and vice versa, hyperlink mentioned issues, move or mirror defects from Collaborator to your issue-tracking system. Previous implementation of JIRA integration, which allowed creating reviews from JIRA tickets' linked commits or using JIRA as notification channel, remains functional but was renamed to JIRA Legacy Integration. (COLLAB-2594)

added - new Remote System Links 353 section to the Review Summary Screen. This section lists all pull requests, direct pushes and issue-tracking items associated with the review, as well as the current statuses of those items. (COLLAB-2794)

added - Support for Team Foundation Server 2017 (COLLAB-2696)

added - Collaborator Visual Studio Extension now supports for Visual Studio 2017 (COLLAB-2695)

added - Collaborator GUI and command-line clients now use TFS SDK to interact with Team Foundation Server integrations. For GUI Client this made possible to retrieve a list of changesets 744 and select them visually, rather that entering changeset IDs by hand. (COLLAB-2606, COLLAB-2609)

added - The logic of "Visible Phase" and "Due by Phase" custom field 256 options was re-adjusted to meet some of our customer use-cases. (COLLAB-2462)

added - Web Client now displays date and time values in "yyyy-MM-dd hh:mm" format. However, the default date and time format can be changed via user preferences 321. (COLLAB-792)

added - RTC client plug-in[716] now supports RTC servers versions from 4.x to 6.03. Previously, because of RTC API changes in RTC 6.0.1, for RTC 4.x-6.0 servers you had to use one version of plug-in and another version for RTC 6.0.1 servers. (COLLAB-2579)

added - Now administrators can create a dump with all reviews of some particular user[175] (COLLAB-2637)

fixed - GitHub, GitLab, Bitbucket Cloud: "File not found" exception occur when merging a branch where some file have been deleted in its' parent branch (COLLAB-2815)

fixed - GitHub: pull request comment not reflected in review (COLLAB-2805)

fixed - GitLab, Bitbucket Cloud: make possible to use repo clone URL in remote systems (COLLAB-2781)

fixed - Disable password autocomplete on login form (COLLAB-2723)

fixed - GitLab: Comments not added to a review (COLLAB-2718)

fixed - Git's "The file will have its original line endings in your working directory." message is not ignored (COLLAB-2707)

fixed - Oracle: Slow search in CLOBs (COLLAB-2692)

fixed - The "Respect permissions" value of "Report access" setting was ignored (COLLAB-2687)

fixed - Poking a user in Annotating phase makes that Review invisible to that user under my reviews from the Collaborator Homepage (COLLAB-2679)

fixed - Bitbucket Server commit may contain no more than 25 files, Bitbucket Cloud commit may contain no more than 10 files (COLLAB-2662)

fixed - Redesign addchangelist flow for Perforce, remove redundant use of "p4 change" command (COLLAB-2654)

fixed - Visual Studio Extension does not support line comments for .cshtml files (COLLAB-2633)

fixed - Cannot upload .doc and .docx files if they are added to the "Binary file" list (COLLAB-2597)

fixed - Cannot restore dump file with huge .xml files in it (COLLAB-2593)

fixed - Perforce: the update-changelist trigger does not use the most recent review for description (COLLAB-2583)

fixed - Large Oracle database may cause Home Screen performance issue while loading action items (COLLAB-2578)

fixed - PDF's in Diff Viewer blurry at 1000% Magnification (COLLAB-2565)

fixed - The "ccollab admin wget" command-line command allows to archive reviews for users who have no access to this review (COLLAB-2540)

fixed - There's no option to drop the --devpath argument usage while in PTC integration (COLLAB-2526)

fixed - The [Supported Character Sets] link does not work (COLLAB-2501)

fixed - The default review template value for groups is not updated (COLLAB-2499)

fixed - Reviews can move to completed by clicking Wait button first (COLLAB-2461)

fixed - A file uploaded to Collaborator should have the same filename when it is downloaded from the DiffViewer (COLLAB-2435)

fixed - Author should be notified if no materials are attached when begins a review (COLLAB-2335)

fixed - Disabling Collaborator news feed causes SEVERE error entries in the error.log file (COLLAB-2332)

fixed - Link to Collaborator review from RTC workitems is broken (COLLAB-2328)

fixed - Email notifications does not contain SSO-compatible URLs (COLLAB-2323)

fixed - Changing group in review resets all information (COLLAB-2300)

fixed - The "None" item in the Default review template drop-down list does not make sense (COLLAB-2293)

fixed - Deleting the JIRA Project custom field does not delete the field's items (COLLAB-2279)

fixed - Incorrect order of files displayed in the Diff Viewer (COLLAB-2167)

fixed - "Add Perforce diffs" command incorrectly marks files as deleted/added (COLLAB-2027)

fixed - "Save to ZIP" creates "0" size report (COLLAB-1978)

fixed - Eclipse: Collaborator perspective hangs up (COLLAB-1938)

fixed - Command-line commands fail to find users with numerical logins (COLLAB-1877)

fixed - RTC: Wrong order for loading collab server jar files by Tomcat 8 (COLLAB-1855)

fixed - ccollab admin review-xml command does not return the deadline value (COLLAB-1848)

fixed - Moving review to another phase does not change "Author" state (COLLAB-1794)

fixed - Incorrect locations of pushpin defects in CSV Reports (COLLAB-1768)

fixed - P4DashGParser cannot parse changelists with huge descriptions (COLLAB-1688)

fixed - When non-standard role configuration is used, reviewers may need to approve the same review twice (COLLAB-1673)

fixed - Participant state was shown as "Approved" even if they have sent a review to rework (COLLAB-1436)

fixed - Participant drop down does not show full list of users when this list is too long (COLLAB-1205)

fixed - Better resolution for documents in DiffViewer (COLLAB-709)

**Discontinued Support**

With the release of Collaborator 11.1, we will no longer support uploading URL's as review materials.

## 11.0.11000 - March 7, 2017

* FIRST RELEASE OF v11.0! *

**New Features**

- **GitHub, GitLab and Bitbucket**

In this release we are introducing support for GitHub, GitLab and Bitbucket. Our integration support allows for users to easily create Collaborator reviews whenever a pull request is initiated. In addition to creating the review, Collaborator can merge the pull request or close its branch once the Collaborator review is completed. Finally, we've made it easier to add repositories by basing their management off of webhooks.

See Repository Hosting Service Integrations 826

- **Custom Syntax Highlighting Schemes**

Collaborator has always supported syntax highlighting for a number of languages. However, this latest release adds the ability to create custom syntax highlighting schemes. This change now allows Collaborator to support syntax highlighting for any language. To learn how to create custom syntax highlighting schemes, read Syntax Highlighting 304.

- **Document Review Improvements**

In Collaborator 11.0, users can now Pan, Zoom and Scroll using their mouse. The addition of these features makes reviewing both documents and images easier and more intuitive. Diff Viewer 367 was reworked to support these mouse actions.

- **Hide Files Under Review**

Often times changelists include files that do not necessarily need to be reviewed. Though they do not need to be reviewed, they are still part of an atomic changelist and cannot be deleted from the review. Hide Files allows users to selectively determine which files are displayed on the Review Summary screen. Read, Hiding Files From Review 453 to learn how to use this feature.

**More improvements:**

- Collaborator now support file paths up to 1024 characters long. Earlier versions have file path limitation of 255 characters. (COLLAB-1459)
- "Triggers > Executable" and "Triggers > Parameters" fields now can be up to 1024 and 2048 characters long, respectively. (COLLAB-1692)
- Upgrade pdfbox library to 2.0.4 (COLLAB-2115)
- A new VM option 1240 that instructs Collaborator to use LIKE predicate instead of REGEXP_LIKE predicate in queries to Oracle database. REGEXP_LIKE predicate works slower and may cause performance issues on large Oracle databases. (COLLAB-2543)

**Discontinued Support**

With the release of Collaborator 11.0, we will no longer be supporting Collaborator versions 9.x and older.

**Bug Fixes**

- fixed - Remove "abandon" DB options added as part of the upgrade to Tomcat 8 (COLLAB-2545)
- fixed - Tomcat 8 is scanning all files in the content-cache, causing extended server start times (COLLAB-2544)
- fixed - Running ccollab admin group sync causes NPE in 10.2 (COLLAB-2430)
- fixed - Impossible to save state for "Restrict Access to Review" option (COLLAB-2340)
- fixed - Problem with navigating between Review Materials and Diff Viewer (COLLAB-2317)
- fixed - JSON API should create only one changeset when multiple files changed (COLLAB-2313)
- fixed - "Newline" replaced with '#13;#10;' after migration (COLLAB-2304)
- fixed - Multi-select custom fields do not list all selected values (COLLAB-2242)
- fixed - External Diff Launcher crashes with "java.lang.NoClassDefFoundError" (COLLAB-2236)
- fixed - Long defect text and multiple defect custom fields can push the General Info page's bounds out of the browser's window (COLLAB-2210)

- fixed - "ccollab addchangelist ask" cannot authenticate (COLLAB-2194)
- fixed - Remove redundant log error message from RemoteSytemTrigger (COLLAB-2179)
- fixed - CLI 'ccollab addfiles' sends notification to 'Author' (COLLAB-2176)
- fixed - Typo "You are pocked" in VS Extension (COLLAB-2173)
- fixed - Github: Upgrade to 10.1 produces exception when completing review (COLLAB-2151)
- fixed - RTC: Link between RTC and Collab Review (COLLAB-2140)
- fixed - 'Admin > General > Select Group Prompt:' should be depreciated (COLLAB-2003)
- fixed - Client installer fail to install CCRC jars (COLLAB-1991)
- fixed - Collaborator does not clean store-*.zip temporary files (COLLAB-1970)
- fixed - Improper diff variable values for PTC (COLLAB-1966)
- fixed - addgitdiffs added "(none)" Author to "Review materials section" (COLLAB-1947)
- fixed - Multiple authors do not receive notifications (COLLAB-1890)
- fixed - "Delete" button is still enabled after files are uploaded as separate changelists/upload processes. (COLLAB-1869)
- fixed - Detailed report does not retrieve conversation (COLLAB-1868)
- fixed - VS Extension: Use the command line client instead of the GUI client (COLLAB-1862)
- fixed - Collab GUI client cannot connect to server via HTTP proxy (COLLAB-1840)
- fixed - ccollab admin review-xml wrong metrics (COLLAB-1681)
- fixed - DiffViewer: Performance issue in case "wrap lines" unchecked and enabled "Synchronize Scrolling" (COLLAB-1595)
- fixed - Author is not notified when review is completed in case of minimal notification level (COLLAB-1561)
- fixed - Detailed Report: Add a new report to Defect Log section (COLLAB-1543)
- fixed - RTC: Separate view shows changelist(s) twice (COLLAB-1533)
- fixed - Do not allow to duplicate Group Name (COLLAB-1493)
- fixed - Improve messaging to user that an upload has failed (COLLAB-1456)
- fixed - Subscriptions mode does not work correct (COLLAB-1444)

- fixed - LOC does not correctly consider after the changes (delete or change lines). (COLLAB-1416)
- fixed - Customizable Review Reports retrieve wrong LOCs (Changed, removed, added, modified) (COLLAB-1315)
- fixed - Reviewer receives notifications when in "Active" state (COLLAB-1270)
- fixed - Long review title pushes edit buttons off the screen to the right (COLLAB-1267)
- fixed - Eclipse: Inconsistent behavior of "Delete"/"Cancel"/"Reopen" review button in Eclipse. (COLLAB-1186)
- fixed - LOC (Changed) is not calculated in "Detailed Report" (COLLAB-1184)
- fixed - LOC not matching between review materials, diff viewer and reports (COLLAB-650)

## 11.2.4 Version 10

### 10.2.10200 - December 23th, 2016

- added - Server Branding 168 - Now Collaborator Enterprise server can display a custom company logotypes on Login, Home and Review Summary screens. Just enable the respective options and upload image files with your company logo. (COLLAB-1713)
- added - Support for Calendar notifications 451 - You can send an iCalendar invitations to review participants. Calendar notifications may be useful for scheduling formal meetings on review. (COLLAB-1842)
- added - Deadline notifications 297 - New type of notifications that inform about reviews approaching deadline. (COLLAB-2260)
- added - New ccollab admin find review command that allows users to search Custom Field data from command-line (COLLAB-2259)
- added - Updates in end-user license agreement (EULA) (COLLAB-2113)
- fixed - "Allow duplicate group names" impacts on "group sync" (COLLAB-1980)
- fixed - NPE when change "Allow Create Review to" == "Group Members only" (COLLAB-2282)
- fixed - Visual Studio Extension could crash the IDE when the solution files contain the characters that are not allowed in path names. (COLLAB-2156)

### 10.1.10101 - November 14th, 2016

- fixed - Remove date stamp from News feed (COLLAB-1996)
- fixed - GitHub integration may reach API calls rate limit and freeze till it is reset (COLLAB-2106)
- fixed - Installer fails to set admin user in ROOT.xml (COLLAB-2137)
- fixed - Fix Defect related graph in Savings Report (COLLAB-2153)
- fixed - Collaborator does not count an author in the participant list in custom templates (COLLAB-2166)
- fixed - An error may occur while upgrading server database to 10.1.10100 (COLLAB-2187)
- fixed - Collaborator does not display "Visible Phase" and "Due by Phase" options for the custom fields (COLLAB-2186, COLLAB-2188)

### 10.1.10100 - October 4th, 2016

- added - Bitbucket Integration 826 - Now you can use Collaborator to review changes on Bitbucket Cloud server (COLLAB-1509)
- added - Advanced User Permissions 225 - You may grant certain users with elevated permissions, so that they become able to perform some of administrative tasks: manage and create user groups, manage templates, custom fields, checklists, roles, automatic links (COLLAB-1229)
- added - Savings Report 301 - New type of report that displays how much money you have saved by using Collaborator compared to non-code reviewed process (COLLAB-1951)
- added - Admin Settings have been restructured and divided into several tabs to group related items and reduce the amount of scrolling (COLLAB-1943, COLLAB-1945)
- added - Template management: default review template (COLLAB-1859)
- added - New Diagnostic function to clear up the bad pushpin data in comment and defect table (COLLAB-1813)
- added - Support for PGP signed commits in Git (COLLAB-1831)
- added - SSL implementation in JSON (client-side API handling) (COLLAB-1492)

- added - new JSON (`ReviewService.getPinCoordinates`) and CLI commands (`ccollab pinCoordinates`) for obtaining pushpin locations and numbers (COLLAB-1759)
- added - JIRA integration: Add support for creating empty reviews (COLLAB-1979)
- added - JIRA integration: Support FishEye servers that are not at the domain root (COLLAB-1876)
- added - GitHub integration: Ability to track branches like 'prefix/name' (COLLAB-2020)
- added - GitHub integration: Options to close pull request and/or delete feature branch when the corresponding review completes (COLLAB-1047)
- added - GitHub integration: checking for empty pull requests (COLLAB-1954)
- added - VS Extension: Support HTTPS connections (COLLAB-11748)
- added - VS Extension: Support "Dark" theme (COLLAB-1860)
- added - VS Extension: Support e-signatures (COLLAB-1815)
- added - RTC: Select which change set to add (COLLAB-996)
- fixed - GitHub Enterprise: SSL issue (COLLAB-2090)
- fixed - ClearCase Remote Client integration issues (COLLAB-2034)
- fixed - Obfuscating LDAP/database password issues (COLLAB-1988)
- fixed - Unix's GUI client cannot work on pure 32-bit systems (COLLAB-1956)
- fixed - Fun Fact logic on Home page was refactored (COLLAB-1940)
- fixed - Unable to create review if the 'Restrict Access to Review' setting was set to 'Participants only' (COLLAB-1937)
- fixed - VS Extension: Incorrect review state was displayed (COLLAB-1936)
- fixed - Comments/Conversation model refactoring (COLLAB-1871)
- fixed - VS Extension: Use the command line client instead of the GUI client (COLLAB-1862)
- fixed - Command "`ccollab admin user create`" throws InternalError if user login already exists (COLLAB-1797)
- fixed - Property "`com.smartbear.ccollab.license.noperiodicupdate`" does not function (COLLAB-1779)
- fixed - Disabled user can perform review via CLI (COLAB-1763)

- fixed - No Content-Type header in responses to UI requests (COLLAB-1706)
- fixed - PTC change packages do not include all changed files (COLLAB-1566)
- fixed - Problem with Perforce environments where both P4CONFIG and P4ENVIRO are defined (COLLAB-1545)
- fixed - Exception occurs if enter incorrect password during GUI client installation (COLLAB-1544)
- fixed - Cannot upload changelist to a review unless you are a participant (COLLAB-1407)

### 10.0.10001 - June 28th, 2016

- fixed - security issue in JSON API (SessionService.getLoginTicket). Affected versions are 9.5.9500, 9.5.9501 and 10.0.10000; all users of affected versions are urged to update to 9.5.9502/10.0.10001 as appropriate. The team would like to thank Enrique Tobis for the notice.
- fixed - Connection to GitHub Enterprise using SSL (COLLAB-1866)

### 10.0.10000 - June 8, 2016

* FIRST RELEASE OF v10.0! *

#### New Features

- Using a brand new **Collaborator extension for Microsoft Visual Studio** 566, you can create and participate in Collaborator reviews directly from within the Visual Studio IDE. The extension installs into the Community, Professional and Enterprise editions of Visual Studio 2015 and 2013. The extension uses a stand-alone installer that is available for download from our web site:

  ⇒ http://support.smartbear.com/downloads/collaborator/

- SmartBear now offers **three Collaborator editions** making it more suitable and affordable for teams of different sizes:
  - Collaborator Community (formerly named CodeReviewer) is intended for small developer teams,
  - Collaborator Team is for medium teams, and
  - Collaborator Enterprise is for large enterprise-level companies.

For complete information on differences between the editions, see the comparison page 3.

- **Improved GitHub support**:
  - Collaborator now allows viewing the whole changes history in uploaded files. You can use the Diff Viewer to look through the GitHub change lists to find out the upload history and browse differences between all versions. See User Preferences 317 to learn about different modes of the Diff Viewer.
  - If you encounter any issues while working with GitHub, you can now check the log to find out the GitHub version and API endpoint information for current connection and use it for troubleshooting.
  - Reviews created for push requests now have a link to the appropriate commit. In earlier versions of the product, only "pull request" reviews had such a link.
- **Enhanced Rational Team Concert support**:
  - Added support for **Rational Team Concert 6.0.1**.
  - The Collaborator plugin for Eclipse adds two new context menu items 704 to the Work Items view. They help you easily add files and folders associated with a work item and its child items to a review.
  - Now you can also have reviews created automatically on WorkItem status change, and attach changelists to them later.
- **Support for gated check-ins in Team Foundation Server**. Collaborator 10.0 offers two triggers that help you ensure that a review is started or completed for each changelist. See Gated Check-in Triggers 744 for details.
- **Easier navigation in Diff Viewer**. In earlier versions of the product, scrolling worked only within the current page of PDF and Word documents. To switch between pages, you had to use buttons at the bottom of Diff Viewer. Now, scrolling works throughout the entire document which makes navigation faster and easier a lot.
- More improvements:
  - You can view and edit reviews' **checklists in Eclipse** now.

- Added support for **Subversion 1.9.3**.
- Now it is possible to obtain the list of Collaborator users and groups through API.
- Added new `--pref-dir` 516 global option which allows to specify a custom location of Collaborator configuration files 495.
- Users who are using a deprecated `--use-legacy-api` command-line argument will be notified about this at the review creation time.

### Discontinued Support

The Collaborator GUI Client no longer supports the TLS v1.0 protocol.

### Bug Fixes

- fixed - Collaborator Server did not delete some data on reviews when a user deleted these reviews (COLLAB-585)
- fixed - Collaborator displayed an incorrect message when a user trying viewing a file that had been deleted from a review (COLLAB-648)
- fixed - An exception occurred in ccollab.exe when the --user command-line argument was specified and the Collaborator Server used SSL connections (COLLAB-732)
- fixed - If a user uploaded several Rational Team Concert changesets and one of them was empty, the entire upload failed (COLLAB-813)
- fixed - ccollab permits login with an incorrect password, if earlier logins were successful (COLLAB-1140, COLLAB-1676)
- fixed - Collaborator formed an incorrect link for adding a bug to an external issue tracker (COLLAB-1212)
- fixed - An exception occurred when files from JGit 4 were added to a review (COLLAB-1252)
- fixed - The Review screen did not display Unicode symbols correctly (COLLAB-1304)
- fixed - The ccollab addhgdiffs command did not work (COLLAB-1308)
- fixed - Collaborator's Diff Viewer failed detecting certain changes in files and did not highlight them (COLLAB-1474)

- fixed - The Diff Viewer did not render some PDF files correctly (COLLAB-1525)
- fixed - GitHub integration: the Diff Viewer's list of revisions could differ from the list of uploaded versions (COLLAB-1425)
- fixed - GitHub integration: if you created a pull request and a review after merging change from the main branch to your branch, the review also included the "merged" files from the main branch (COLLAB-1506)
- fixed - If you uploaded the same changeset or a shelveset from Team Foundation System 2015 twice, Collaborator saw files in them as different files (COLLAB-1572)
- fixed - The Collaborator GUI Client installer copied TFS support libraries to a wrong directory. This caused certain issues with adding shelvesets to reviews (COLLAB-1672)
- fixed - The drag-and-drop file functionality did not work in Internet Explorer 10 and 11 (COLLAB-1296)
- fixed - An attempt to upload files to a review page open in Internet Explorer 9-11 could make the review page inactive (COLLAB-1619)
- fixed - Upgrading Collaborator to a newer version changed the SSL port number in server settings (COLLAB-1626)
- fixed - Some actions with Perforce changelists required administrator permissions (COLLAB-1627)
- fixed - Collaborator did not support Unicode symbols in review dumps (COLLAB-1632)
- fixed - In certain cases, ccollab.exe did not display the debug window (COLLAB-1662)
- fixed - The Diff Viewer dialog did not display unsupported file types correctly (COLLAB-1801)
- fixed - In certain cases, users could see a blank screen when logging in to Collaborator (COLLAB-1718)
- fixed - Scm config was html escaped while saving new scm in database (COLLAB-1812)
- fixed - Addon for Perforce Visual Tool could open GUI client instead of "Add Changelist to Review" dialog (COLLAB-1550)
- fixed - Add defect state to the Detailed Report (COLLAB-1404)
- fixed - Typo in the result of "ccollab admin review create" (COLLAB-1554)

## 11.2.5 Version 9

### 9.5.9502 - June 28th, 2016

- fixed - security issue in JSON API (SessionService.getLoginTicket). Affected versions are 9.5.9500, 9.5.9501 and 10.0.10000; all users of affected versions are urged to update to 9.5.9502/10.0.10001 as appropriate. The team would like to thank Enrique Tobis for the notice.

### 9.5.9501 - April 1st, 2016

- added - configurable polling interval for GitHub integration (COLLAB-1320)
- added - link to GitHub Pull Request in reviews created from them (COLLAB-1389)
- added - enhance diff viewer support for COBOL files (COLLAB-1528)
- fixed - spurious reviews created when GitHub and JIRA integrations simultaneously configured (dependency issue affecting JIRA triggers) (COLLAB-1600)
- fixed - legacy xmlrpc API support broken in 9.5 (COLLAB-1598)
- fixed - NullPointerException when editing non-existent review via command line (COLLAB-1557)
- fixed - PDF file not rendered properly in diff viewer (requires 3rd party library update) (COLLAB-1525)
- fixed - ensure that duplicate GitHub remote system configurations are not allowed (COLLAB-1377, COLLAB-1603)
- fixed - allow multi-line values to be passed for custom fields from the command line (COLLAB-1239)
- fixed - 9.5.9500 breaks client compatibility with some SCMs requiring native libs (e.g. TFS) (COLLAB-1618)
- fixed - Error while uploading .docx files to review materials (COLLAB-1589)

### 9.5.9500 - March 8, 2016

- added - Integration with Single Sign-On servers 130 (COLLAB-361)
- added - JIRA integration was improved to comply with JIRA 7 API (COLLAB-1402)
- fixed - Updated Apache Tomcat server to version 8.0.28 to resolve a number of performance and security issues (COLLAB-583)

- fixed - DiffViewer: Previous diff showing in HTML code for review sourced from JIRA (COLLAB-1477)
- fixed - ccollab admin review edit command for custom field changes group to "All Users" (COLLAB-1451)
- fixed - ccollab adddiffs does not work because of JSON API issue (COLLAB-1450)
- fixed - Doubled pushpins with wrong locations between uploaded material's versions (COLLAB-1446)
- fixed - StringIndexOutOfBoundsException being thrown from GitHub event handler (COLLAB-1440)
- fixed - Wrong child group reflection in participant drop-down (COLLAB-1434)
- fixed - "ccollab admin review edit --deadline" successfully executed with '0' in the admin settings of the UI (COLLAB-1426)
- fixed - Server cannot be upgraded to 9401 because of pushpin issue (COLLAB-1421)
- fixed - MKSAPI.JAR contains an unsupported version of Apache HttpClient (COLLAB-1420)
- fixed - Client cannot parse the "p4 -G change -o" output if it contains Perforce's additional messages (COLLAB-1414)
- fixed - IllegalStateException raised when running a diagnostic (COLLAB-1403)
- fixed - Participant’s filter does not correctly show an account with a blank space in the “Full name” field (COLLAB-1397)
- fixed - Review custom fields of Multi Select type are not highlighted (COLLAB-1374)
- fixed - "ccollab admin review finish ask" does not work properly (COLLAB-1363)
- fixed - Cannot execute addchangelist if password is omitted (COLLAB-1340)
- fixed - CMVC, StarTeam, Surround, VSS and Vault support with older clients is broken on the server (COLLAB-1334)
- fixed - JSON API: UserService: login values are case insensitive (COLLAB-1327)
- fixed - Add "HTTPOnly" marker in the browsers session cookie (COLLAB-1243)
- fixed - Add "Secure" marker in the browsers session cookie (COLLAB-1242)
- fixed - "--restrict-access" parameter does not work with JSON API (COLLAB-1214)
- fixed - Wrong groupId="1" in review then creating review via GUI Client (COLLAB-1143)

- fixed - "Edit" button is grayed for uncompleted reviews (COLLAB-629)

## 9.4.9401 - December 16, 2015

- fixed - Unable to open file in diff viewer that was added to review by MS Word add-in (COLLAB-1393)
- fixed - improvements to GitHub pull request handling (COLLAB-1401)
- added - Digital signatures on Windows installers updated
- added - Windows and OSX installers use bundled JRE if none available (COLLAB-1392)

## 9.4.9400 - December 8, 2015

- added - Rejected and Cancelled reviews can also be archived to Zip 165 (COLLAB-1323)
- added - Support for Rational Team Concert 6.0 (COLLAB-999)
- added - Use CSRF tokens during web transactions to prevent cross-site request forgery (COLLAB-1241)
- added - Administrators can specify a list of file types restricted for upload 196 (to avoid malicious file uploads) (COLLAB-1234)
- added - Filter user-input to avoid cross-site scripting vulnerability (COLLAB-1233)
- added - Commands to administer remote system integrations via Command-Line Client (COLLAB-1306)
- added - File upload security improvement. (COLLAB-1102) Web clients, by default, use an HTML5-based component for file uploads instead of the older Adobe Flash method.
- added - GitHub integration: New "Ignore pushes for branches" setting to skip review creation for raw pushes to the specified branches.
- fixed - GitHub integration: Polling thread stops if a comment exists on the pull request before Collab processes it (COLLAB-1373)
- fixed - GitHub integration creates multiple reviews with a single push to main branch(COLLAB-1352)
- fixed - Critical error when upgrading from 9.0 if database misses version data for some comments (COLLAB-1338)
- fixed - JSON API: UserService: login values are case insensitive (COLLAB-1327)

- fixed - Cannot click on a pushpin and have it select the conversation (COLLAB-1310)
- fixed - GitHub integration is pulling extra commits into reviews (COLLAB-1299)
- fixed - GitHub integration commits in the diff viewer show wrong timestamp (COLLAB-1298)
- fixed - Some groups are missing from the pick-list of new reviews (COLLAB-1273)
- fixed - Use TLSv1.2 protocol on Java 1.6 clients (COLLAB-1213)
- fixed - Incorrect time stamp is displayed in DiffViewer for ClearCase Activity in 9200 Client (COLLAB-1174)
- fixed - Updated Apache PDFBox library to version 1.8.10 to resolve a number of PDF-related document handling issues (COLLAB-1091)
- fixed - Eclipse client dependency for JGit makes it difficult to install into RTC Eclipse client (COLLAB-962)
- fixed - DiffViewer does not show upload number correctly when uploading git diffs multiple times (COLLAB-885)
- fixed - The --non-interactive option is ignored when SSL certificate cannot be validated (COLLAB-357)
- deprecated - Print To Review virtual printer driver - Collaborator now offers the ability to attach Word 383, Excel 389 and PDF 414 documents directly to a review. Additionally, you can install Collaborator add-ins for Word and PowerPoint.
- deprecated - SOAP API - Collaborator introduced a new JSON API in January 2015 and the limited SOAP API is no longer needed.
- deprecated - XML-RPC API - Collaborator's XML-RPC API is no longer needed because of the introduction of the JSON API.
- deprecated - Visual Studio Add-in - We are currently working on a new add-in that will support Visual Studio 2013 and 2015 (and, hopefully, provide a much better user experience than the existing one).

## 9.3.9300 - October 13, 2015

- added - Completed reviews can now be archived to Zip 165. (COLLAB-1105)

- added - The style of coordinate comments and defects 425 (pushpins) have been changed.

  Now the pushpins display an integer number in their head: The number corresponds to the order in which that pushpin was added to the document page or image. (COLLAB-1179)
  **Note:** During the upgrade to Collaborator 9.3 all **in-progress** reviews have been updated and their pushpins (if any) were numbered. Converting only in-progress reviews during the upgrade, ensures that upgrading the server to 9.3 does not take too long. To append ordinal numbers to pushpins in all completed reviews, your administrators can use the respective command in the Diagnostic Utility. (This however may take some time to accomplish.)

- added - Collaborator News panel in the WebUI 313. (COLLAB-1104) This panel displays news, release announcements, webinar invitations and other information from the SmartBear.

- added - AES-256 encryption 104 for database and LDAP passwords. (COLLAB-1077)

- added - Send notification to the administrator when a user access was denied due to license limit 92. (COLLAB-985)

- added - Support for Oracle 12c database. (COLLAB-1121)

- added - Support for Microsoft Team Foundation Server 2015. (COLLAB-1144)

- added - Support for Microsoft Windows 10 (both server and client components of Collaborator). (COLLAB-1095)

- added - Support for Microsoft Edge browser in Web Client. (COLLAB-1096)

- fixed - Changes of document(s) uploaded to a review were not highlighted. (COLLAB-1145)

- fixed - JIRA Configuration did not remove the trailing slash from Server URI. (COLLAB-1138)

- fixed - Git's "The file will have its original line endings in your working directory." message was not ignored. (COLLAB-1107)

- fixed - Collaborator did not apply the ASCII expansion to Perforce wildcards. (COLLAB-1103)

- fixed - JSON API was not implemented for the "ccollab admin review-xml" command. (COLLAB-1084)

- fixed - Customer's completed reviews showed the "overdue" status. (COLLAB-1066)

- fixed - The `smartbear.ccollab.upload.ignore.binary.file` property was ignored by JSON APIs. (COLLAB-998)

- fixed - Private or non-SCM files prevented review uploads from Eclipse Plug-in. (COLLAB-945)

- fixed - The location data for defects within documents was displayed incorrectly in the Defect report. (919)

- fixed - Some users could log in even if all floating licenses was consumed. (COLLAB-694)

- fixed - The filters of the Diff Viewer's chat panel did not function properly. (COLLAB-710)
- fixed - The Before and After drop-down lists of the Diff Viewer displayed file ids rather than commit ids for GitHub files.
- fixed - Diff Viewer navigates between files alphabetically even when File View is set to "Compressed Tree". (COLLAB-389)

## 9.2.9200 - July 9, 2015

- added - JIRA integration
- added - GitHub integration 826
- added - Electronic signatures 193 can now be configured on templates basis (COLLAB-633)
- added (open source, beta): IntelliJ IDEA plugin (https://github.com/SmartBear/idea-collaborator-plugin)
- added - Eclipse Plug-in, GUI Client, and Command-Line Client all now use the JSON API completely and by default (to get old behavior specify `--use-legacy-api` option with Command-Line Client)
- fixed - fixed HostGUID handling by JSON API; primarily affecting Perforce environments (COLLAB-1053)
- fixed - Table naming issue in Oracle-based environments (COLLAB-1041)
- fixed - Defect marking in Eclipse conversation tab stops changing colors (COLLAB-1035)
- fixed - Action item list not refreshing after defect creation in Eclipse plugin (COLLAB-1034)
- fixed - Respect a Perforce changelist's comment when submitting it to Collaborator (COLLAB-1021)
- fixed - Diff viewer should handle empty and broken files (COLLAB-1005)
- fixed - Collab tray notifier UnsupportedOperationException error (COLLAB-1003)
- fixed - NullPointerException when ccollab addchangeslist is run with Perforce (COLLAB-966)
- fixed - Eclipse client could not create line defects (COLLAB-979)
- fixed - Failure to create new review in GUI client when debug was turned on (COLLAB-964)
- fixed - Always allow listed authors to upload content to a review, regardless of restrictions (COLLAB-959)
- fixed - Unclear GUI client error message during "test connection" if a 9.x client is used with a < 9.0 server (COLLAB-955)

- fixed - Improved client support for copy-participants operation (COLLAB-954)
- fixed - Improved client support for saving defect id on creation/editing (for "last" argument) (COLLAB-953)
- fixed - Made checking mandatory custom fields consistent for defect commands in CLI in both legacy and JSON API modes (COLLAB-952,1069,1070)
- fixed - Mandatory review custom field was not working for annotating phase (COLLAB-948)
- fixed - Mandatory review custom field was not made obvious with the typical red label for mandatory fields (COLLAB-947)
- fixed - Incorrect "send to completed" button and associated description for author in a "formal inspection" template review (COLLAB-937)
- fixed - Clarified docs and enhanced behavior of assignReviewPool API call (COLLAB-923)
- fixed - Made sure that review phase "Rework" always identifies as such w/ trigger substitution variables ${review.phase}/${review.phase.previous} (COLLAB-918)
- fixed - Cannot delete .docx documents from review materials (COLLAB-878)
- fixed - addcvsdiffs does not work w/ -N cvs option; use -skipN option to addcvsdiffs if needed (COLLAB-738)
- fixed - Verify that we can run on JRE 8 (COLLAB-682)
- fixed - Perforce update-changelist trigger fails when workspace has the "Host" field specified (COLLAB-667)
- fixed - Uploading a new version of a file changed chat icons from acceptance to chat icon (COLLAB-226)

## 9.1.9101 - April 15, 2015

- added - New "First uploaded vs. Last uploaded" option for the Diff Viewer Default Version Comparison 324 setting (COLLAB-841)
- added - JSON API: Allow defects and comments to be queried by review/user (COLLAB-840)
- added - JSON API protocol implementation for trigger commands (COLLAB-799)
- added - Allow pre-configuration of scale parameters during server installation (COLLAB-884)
- fixed - Download Client Installers button points to version 8 download page (COLLAB-910)

- fixed - Infinite Loading message when clicking Next button in diff viewer (COLLAB-888)
- fixed - Mandatory review subscriptions prevent review from moving to completed phase (COLLAB-797)
- fixed - Browser launching is broken in all *nix operating systems, including mac for all client apps (COLLAB-603)
- fixed - Author receives "Respond to comments" email when all review pool slots are taken and review begins (COLLAB-601)
- fixed - AccuRev addchanges fails when getting item transaction (COLLAB-503)
- fixed - Blank default start menu folder name for client install (COLLAB-917)
- fixed - addhgdiffs command with JSON causes NullPointer exception (COLLAB-886)
- fixed - addfiles command does not allow to load "local" files when SCM set to "perforce" (COLLAB-867)

## 9.1.9100 - March 25, 2015

- added - Most Command-Line Client commands (except for triggers) now support the `--use-json-api` global option.
- added - Support for JSON API in GUI Client. Now this API is the default for GUI Client (COLLAB-779)
- added - Support for JSON API in Eclipse Plug-in (COLLAB-855)
- added - Add participant state to the Detailed Report (COLLAB-804)
- added - Track user file activity from the Review Detail Reports (COLLAB-743)
- added - The caption of the Approve button now changes depending on the next phase (COLLAB-564)
- added - Add Collaborator to the list of web browser's search engines[463] (COLLAB-821)
- added - Support for Microsoft Internet Explorer 11 browser in Web Client. (COLLAB-818)
- fixed - Restrict upload if restrict access is set to "Yes" (COLLAB-849)
- fixed - Running the ccollab admin review create command with the --use-json-api option requires the template ID (COLLAB-848)
- fixed - Command line output for admin user create is confusing (COLLAB-833)
- fixed - ccollab logout fails on Linux (COLLAB-832)

- fixed - Eclipse client will not connect to 9.0 server (COLLAB-829)
- fixed - JSON API with addchangelist and invalid login ticket dies, instead of prompting for password (COLLAB-819)
- fixed - Reverted file are not removed from the review (COLLAB-567)
- fixed - Git/Eclipse: Cannot add multiple commits (COLLAB-811)
- fixed - RTC work item approvals do not get updated when a review is deleted in Collaborator (COLLAB-839)
- fixed - Word document-handling memory consumption issues (COLLAB-539)
- fixed - Perforce syncusers command issues; refactoring to fully support --use-json-api (COLLAB-801)

## 9.0.9001 - February 3, 2015

- added - JSON API: review deletion, adding urls to review, dismiss commit todo, move review to annotate (COLLAB-760)
- added - JSON API: findByGuid and getMembers added to GroupService (COLLAB-789)
- added - JSON API: findByGuid and findByLogin to UserService (COLLAB-790)
- added - JSON API: findById and getParticipants to ReviewService (COLLAB-788)
- fixed - JSON API: fixed admin group create (COLLAB-771)
- fixed - JSON API: fixed missing Action categories (COLLAB-715)
- fixed - review pool notifications should send even if review has started (COLLAB-741)
- added - added ability to stop users from deleting review pools via vmoptions (COLLAB-780)
- fixed - handling of first git commit (COLLAB-769)
- fixed - emailing everyone now includes all members of assigned review pools (COLLAB-742)
- fixed - changed review title (COLLAB-767)
- fixed - MKS products now referred to as PTC in UI after company acquisition (COLLAB-753)
- fixed - Eclipse plugin client author field autofills (COLLAB-758)
- fixed - review xml now includes review pool data (COLLAB-580)

## 9.0.9000 - January 20, 2015

* FIRST RELEASE OF v9.0! *

Major features:

- JSON API web service 943 - a fast and effective way to exchange data between your application and Collaborator server.
- Git integration with Collaborator Eclipse Plug-in
- Command-line APIs were refactored.
- Support for Visual Studio 2013, SQL Server 2014, SVN 1.8.
- [beta] Ability to move comments in MS Word documents (.doc and .docx) 387, PDFs 418 and images 413. Supported in Chrome and Firefox only.

- added — Command-Line Client now has the `--use-json-api` global option. If this option is specified, the Collaborator server will use a faster JSON API to execute commands.
- added — Display full path in a tooltip for truncated file path at top of diff viewer (COLLAB-452)
- fixed — Server-side debugging contains plain text passwords (COLLAB-699)
- fixed — Cannot approve a review due to unread comments, but there are no unread comments (COLLAB-721)
- fixed — CVS does not upload all the file as "cvs diff" (COLLAB-299)
- fixed — Child group loses its child relationship when modified by group admin (COLLAB-683)
- fixed — Spaces removed after PDF conversion (COLLAB-686)
- fixed — Schema check misidentifies standard Oracle triggers (COLLAB-557)
- fixed — Inviting a colleague sends an invalid link to the review (COLLAB-652)
- fixed — broken I/O behavior around password prompts (COLLAB-735)
- fixed — Changing review template wipes file-subscription's participants (COLLAB-534)

- fixed — Populated mandatory review custom field prevents review from changing phases (COLLAB-720)
- fixed — Square bracket in custom field name breaks database views in MS SQL (COLLAB-547)
- fixed — Unable to login after upgrade from v.8100.008: Logging install4j errors and ErrorCode404Handler info entries (COLLAB-56)

## 11.2.6 Version 8

### 8.5.8502 - November 11, 2014

- fixed — Coordinate locators limited to 1 million across the whole server (COLLAB-631)
- fixed — Web UI Slow loading additional lines of code when p4protects is enabled (COLLAB-634)
- added — Support for Subversive in Eclipse Luna (COLLAB-514)
- fixed — Notification Table Cleanup Task does not work (COLLAB-590)
- fixed — Action Item list retrieval is unacceptably slow (COLLAB-608)
- fixed — Add pending changelist for Perforce fails (COLLAB-656)
- fixed — Error parsing variables added as key: value in p4 set (COLLAB-651)
- fixed — Improve user message for ensure-diffs-reviewed errors (COLLAB-609)
- fixed — Disable "Delete" button when "Restrict Uploads to Review:" is set to Yes (COLLAB-661)
- fixed — Modifying and saving a local file from the diff viewer in Eclipse does not save the file to disk (COLLAB-626)
- fixed — Redact accidental inclusion of plaintext password from command-line debug log (COLLAB-616)
- fixed — Eclipse: error with svnkit 1.7+ (COLLAB-545)
- fixed — enhanced error messages from ensure-diffs-reviewed (COLLAB-609)
- fixed — clarified initial configuration screen on product installation (COLLAB-660)
- fixed — improve error messaging when git HEAD revision can not be found (COLLAB-640)
- discontinued support — we have stopped supporting the following outdated and rarely used SCMs: IBM CMVC, Borland StarTeam, Seapine Surround SCM, SourceGear Vault and Microsoft Visual Source Safe.
- Installers are now digitally signed on applicable platforms (Windows, OSX).

## 8.5.8501 - September 30, 2014

- feature — Allow obfuscation of LDAP passwords in ROOT.xml (COLLAB-330)
- fixed — addgitdiffs does not generate LOC metrics (COLLAB-597)
- fixed — Issue of converting pdf files with empty pages at the end (COLLAB-576)
- fixed — Mac Installer cannot find Java JRE (COLLAB-577)
- fixed — Orphaned row in assignment table breaks group admin page (COLLAB-572)
- fixed — Client Installer - Ubuntu- Installer does not create links to executables (COLLAB-559)
- fixed — Discarded MKS change packages showing in GUI client (COLLAB-252)
- fixed — PDF conversion error (COLLAB-594)

## 8.5.8500 - August 26, 2014

- feature — Collaborator Server now can run on MySQL version 4.1, 5.0, 5.1, 5.4, 5.5 and 5.6. (COLLAB-305)
- feature — Added syntax highlighting for XML and XAML files (COLLAB-494)
- fixed — p4 addchangelist uses too much memory when checking file integrations on some cases (COLLAB-454)
- fixed — Discarded MKS change packages showing in GUI client (COLLAB-252)
- fixed — ensure-diffs-reviewed selects contained diffs as possible conflicts (COLLAB-515)
- fixed — Prevent the uploading of certain binary files based on Admin settings 197 and Collaborator Client configuration 1240 (COLLAB-448)
- fixed — smartbear.ccollab.upload.truncate.size no longer works (COLLAB-440)
- fixed — Numbered/lettered bullet list adds extra spaces in diff viewer (COLLAB-469)
- fixed — Problem with php file on diff viewer due to syntax coloring display setting (COLLAB-477)
- fixed — Removed hardcoded timeout (30s) for triggers, timeout is configurable via VM option now (COLLAB-442)
- fixed — Participant custom fields cannot be modified in the completed phase (COLLAB-439)

- fixed — Defect and comment timestamp is not displayed in the Web UI on the non-compact view (COLLAB-436)
- fixed — Regression - We do not support parentless git commits on 8405, but it was supported on Collab 6507 (COLLAB-445)
- fixed — DOM error when clicking next file (COLLAB-487)
- fixed — Exception when clicking review from action items (COLLAB-486)
- fixed — Slow diff viewer refresh performance for Excel spreadsheets (COLLAB-460)
- fixed — Apparent unresponsiveness when interacting with comments in the diff viewer (COLLAB-460)
- fixed — Select Custom field with its values changed causes issue on reviews with the old values selected (COLLAB-473)
- fixed — Wrong file name and extension for review file from mercurial (COLLAB-489)
- fixed — Participant list gets truncated after 10k users (COLLAB-77)
- fixed — No action items for review pool participants (COLLAB-507)
- fixed — p4 - broken symbolic links cause issues with ensure-reviewed triggers (COLLAB-457)
- fixed — Document Diff viewer throws exception before document is finished converting (COLLAB-498)
- fixed — Support MKS Integrity 10 (COLLAB-411)
- fixed — Document conversion failure when restricted fonts are included (COLLAB-316)
- fixed — ClearCase \main\LATEST does not work anymore (COLLAB-535)
- fixed — Eclipse - Support for Subversive in Eclipse Luna (COLLAB-514)
- fixed — Update mysql jdbc driver, 5.1.6->5.1.31 (COLLAB-305)
- fixed — Review Pools: Can move a review forward without one (COLLAB-500)
- fixed — ClearCase "local" keyword does not work in Client (COLLAB-556)

## 8.4.8406 - June 17, 2014

- feature — Added new status for AccuRev incl/excl (COLLAB-269)
- feature — Added new setting to user roles configuration: "Minimum number required to finish review". It defines a minimum number of approval required in order to complete the review.

- fixed — Document converts incorrectly with extra lines (COLLAB-358)
- fixed — Include offending diff output on ensure-diffs-reviewed response (COLLAB-349)
- fixed — Link button in reports is saving URL's that are too large (COLLAB-435)
- fixed — Change default from 'Any Activity' to 'File Activity' (COLLAB-344)
- fixed — ensure-diffs-reviewed fails when there are shared changes between collab and the scm (COLLAB-406)
- fixed — New version of Chrome (and IE9) does not auto-scroll while typing in long diff comments (COLLAB-186)
- fixed — Workflow around Review Pools is not ideal (COLLAB-262)
- fixed — NPE in admin batch execution of adding a git changelist (COLLAB-397)
- fixed — Excel files cannot be viewed in IE8 (COLLAB-368)
- fixed — Populated and expanded defect custom fields hidden from view (COLLAB-418)
- fixed — ClearCase supports relative branch path in “AddVersions" (COLLAB-398); ClearCase should recover the input version information when error occurs (COLLAB-399); ClearCase AddVersion should provide an option to allow if unchanged CHECKOUT files are loaded (COLLAB-400),Show All Files > Branch Name not working properly (COLLAB-421), Review #10161
- fixed — Fixed Database Integrity errors and checks on the Assignment and FileMetrics tables (COLLAB-293)
- fixed — make initialization error page clearer (COLLAB-366)
- fixed — ClearCase > Add Versions > Show All Files > Branch Name not working properly (COLLAB-421)
- fixed — ClearCase supports relative branch path in "AddVersions"(COLLAB-398); ClearCase should recover the input version information when error occurs (COLLAB-399);ClearCase AddVersion should provide an option to allow if unchanged CHECKOUT files are loaded (COLLAB-400)
- fixed — Suppress stacktrace output for document conversion failure (COLLAB-331)
- fixed — Diff Viewer - Display order of after before in bottom detail pane (COLLAB-131)
- fixed — Extra spaces show up in diff viewer in IE11 (COLLAB-362)
- fixed — Perforce Server integration. Adding changelists from server bypasses user access (COLLAB-359)

- fixed — Cannot sort Defect ID column on the defect log of the review summary page (COLLAB-356)
- fixed — addfiles sends an email for each file (COLLAB-303)
- fixed — scmFindOrCreate does not send all info to the server (COLLAB-378)
- fixed — RTC - Incorrect "Pending" label on the "Add to Review" dialog (Case COLLAB-324)
- fixed — The external URL admin setting is reset when the server starts with no DB connection (COLLAB-392)
- fixed — Environment P4CLIENT overrides command's P4CLIENT on p4v (Windows) (COLLAB-369)
- fixed — Make git ignore warning: lines on stderr from commands
- fixed — AccuRev - Incorrect conversion of changelist date (COLLAB-388)
- fixed — Change default from 'Any Activity' to 'File Activity' (COLLAB-344)
- fixed — Fail to load in the dump file which contains the specific character (COLLAB-372)

## 8.4.8405 - May 13, 2014

- fixed — Improved performance of chat message/defect rendering for large reviews (COLLAB-313)
- fixed — Solved issue where addfiles command improperly calculated LOC metrics for non-SCM files (COLLAB-373)
- added — Made comment locations clickable in search results (COLLAB-294)

## 8.4.8404 - May 6, 2014

- fixed — Minimum client version set as "0" in server vs client version error message (COLLAB-308)
- fixed — License logging does not work and need to record license denials in the license.log file (COLLAB-335 & COLLAB-348)
- fixed — Notifications - Back in Inspection Phase email sent at the wrong time (COLLAB-281)
- fixed — Cannot add to existing review with Collaborator and SVN or Git (COLLAB-317)
- fixed — Replace the loading indicator (COLLAB-310)
- fixed — Non-SCM files, documents and spreadsheets are calculating LOC metrics when they should not (COLLAB-140)

- fixed — Problem with compare/sort on 1.7+ JREs (COLLAB-338)
- fixed — ClearCase Addversion fails when there is only one input version (COLLAB-353)
- fixed — License logging does not work and need to record license denials in the license.log file (COLLAB-335 & COLLAB-348)
- fixed — Calculation for number of comments in review detail report includes "accepted" comments (COLLAB-312)
- fixed — Failed uploads result in a missing path in VersionData (COLLAB-337)
- fixed — Calculation for number of comments in review detail report includes "accepted" comments (COLLAB-312)
- fixed — WebUI displays (no comment) when there is not a comment (COLLAB-346)
- fixed — Open the diff viewer with the upload in which the defect was created as the "after" version (COLLAB-133)
- fixed — Collaborator sends a notification email for each uploaded file (ClearCase) whether the file changed or not (COLLAB-325)
- fixed — First cut at resolving client date issues (COLLAB-320)
- fixed — Diff viewer gives no clear indication that reworked files are new (COLLAB-253)
- fixed — Home Displays user has no Action Items on home page before it really determines if the user has Action Items (COLLAB-136)
- fixed — RTC - Changelists are uploaded in wrong order (COLLAB-53)
- fixed — Materials section. Some of the materials disappear when switching between overlay and separate (COLLAB-49)
- fixed — RTC - Changing Work Item state re-uploads unmodified files (COLLAB-289)
- fixed — RTC - Add context menu to add delivered change sets to a review from the History view (COLLAB-307)
- fixed — Group Admins cant set the 'All Users' flag of a group to true (COLLAB-212)
- fixed — Send author inspection notice to users with notification preference set to MINIMAL (COLLAB-268)

## 8.4.8403 - March 25, 2014

- added — ability to obscure database passwords in ROOT.xml (COLLAB-261)
- added — admin has ability to limit subscriber role (COLLAB-57)

- fixed — Eliminated duplicate logins when trailing spaces are included (COLLAB-69)
- fixed — Fixed issue with Previous/Next diff filter (COLLAB-78)
- fixed — Authors are now notified if they are not the creator of the review (COLLAB-85)
- fixed — Memory issue with AccuRev loading (COLLAB-126)
- fixed — Issue with non-participants having certain edit permissions (COLLAB-135)
- fixed — Issue with user selecting a new template and getting error (COLLAB-142)
- fixed — Database integrity diagnostic JoinReviewChangelist error after failed web upload in IE (COLLAB-148)
- fixed — UI issue with General Info area being overwritten (COLLAB-157)
- fixed — Footer incorrect (COLLAB-166)
- fixed — Issue with Group members creating a review that includes non-Group members as the reviewer. Can't check off items in checklist (COLLAB-167)
- fixed — Lack of warning when mandatory participant is removed from a review (COLLAB-174)
- fixed — Exception error with review pools (COLLAB-176)
- fixed — P4 passwords showing in Web UI debug logs (COLLAB-178)
- fixed — ClearCase issue with addversions if there are changes in subdirectory (COLLAB-181)
- fixed — Issue with plugins newer than the server version of Eclipse (COLLAB-183)
- fixed — Get email/full name data upon account creation (COLLAB-205)
- fixed — Field not cleared after creating a group (COLLAB-210)
- fixed — Style issue in the diff viewer – breaks if the conversation is resized (COLLAB-216)
- fixed — Ensure diffs reviewed does not automatically detect merged changes (COLLAB-218)
- fixed — Raise notification subject line to 256 characters (COLLAB-222)
- fixed — List Review Stalled notification template using Review Stalled Author template (COLLAB-227)
- fixed — Fields not cleaned after creating user (COLLAB-234)
- fixed — Reviews not appearing in action items for author (COLLAB-236)
- fixed — Setting "Review template" to the same template in the command line wipes out participants (COLLAB-239)

- fixed — Unable to remove group admins using group sync (COLLAB-240)
- fixed — Review general information: Admins could see only groups where they are members of. (COLLAB-245)
- fixed — Make all searches case insensitive (COLLAB-247)
- fixed — Memory issue with SCM.Uploadchangeset() with Subversion (COLLAB-250)
- fixed — Action item arrow is red even though no action can be done (COLLAB-254)
- fixed — Review Creator cannot edit General Info (COLLAB-271)
- fixed — GIT issue if there is a file named HEAD in the directory (COLLAB-275)
- fixed — P4 addchangelists broken for certain versions (COLLAB-286)
- fixed — Add option to allow uploading other user's perforce pending changelists (COLLAB-287)
- fixed — "AddChanges" fails in CVS (COLLAB-290)
- fixed — RTC: “Save change set links and Comments” follow up is now triggered. (COLLAB-54)
- fixed — RTC: No new reviews are generated if Work Items are linked to rejected or cancelled review (COLLAB-204)
- fixed — RTC - All changesets are not uploaded if one is empty (COLLAB-279)
- fixed — RTC: “Add to Review” to be available for changesets in RTC that have been delivered (COLLAB-208)
- fixed — Clearcase: Adding changes via addchanges from parent directories cause the changes to be added as "uploaded files" (COLLAB-238)
- fixed — ClearCase addversions/addchanges fails when using relative path for Windows (COLLAB-298)

## 8.4.8402 - March 6, 2014

- fixed — Oracle support hotfix (COLLAB-257)

## 8.4.8401 - February 19, 2014

- added — **Accepted** messages now shown on review reports (COLLAB-38)
- added — Added comment link in search results screen (COLLAB-42)
- fixed — Calendar week starts on Monday instead of Sunday (COLLAB-15)

- fixed — Single-Review dumps contain LabelLocation records for all reviews (COLLAB-27)
- fixed — PDF Conversion causes long query warnings (COLLAB-23)
- fixed — Vertical Tab control code breaks Review-XML parser (COLLAB-25)
- fixed — Checklist records not cleaned up when Review is deleted (COLLAB-29)
- fixed — Config only dumps sometimes will not load due to false positive on Duplicate Meta-Data (COLLAB-30)
- fixed — DOCX upload never completes (COLLAB-40)
- fixed — Deleting Admin email now generates error instead of going to first run initialization screen (COLLAB-41)
- fixed — Hourly time calculations in reports stop at 24 hours instead of 30 hours now before rolling into days (COLLAB-55)
- fixed — Notification errors in some cases in Collab.log (COLLAB-24)
- fixed — Adding a review pool group to two different parent groups breaks participants list (COLLAB-76)
- fixed — MS Office graphics objects do not render in diff viewer (COLLAB-79)
- fixed — Broken keyboard shortcuts in Diff Viewer (COLLAB-90)
- fixed — Changing template in completed review causes a hang (COLLAB-112)
- fixed — Diff convos scrolled off screen when you click them (COLLAB-122)
- fixed — Could not select the participant from specific group (COLLAB-124)
- fixed — Loading AccuRev history used too much memory (COLLAB-126)
- fixed — Overview text box is now larger (COLLAB-127)
- fixed — ClearCase 'Local' does not work in 'addversions' command (COLLAB-130)
- fixed — DB import fails with "java.io.IOException: Data format error: Character reference "&#" (COLLAB-134)
- fixed — Cannot display timeline of deleted reviews (COLLAB-145)
- fixed — Perforce client-ownership check does not respect case-sensitivity setting (COLLAB-152)
- fixed — MS SQL Server 2012 - unable to import system dump files (COLLAB-168)
- fixed — Review deadline is showing dummy data on the Reports (COLLAB-170)

- fixed — RTC - Reopening a review does not change the Work Item approval status (COLLAB-175)
- fixed — RTC - Cannot upload change sets when the number of changes is hidden in the RTC client (COLLAB-187)
- fixed — Option to filter users when adding them to groups does not work properly (COLLAB-192)
- fixed — CodeReviewer webpage showing when clicking Help from Collaborator (COLLAB-194)

## 8.3.8301 - December 10, 2013

- added — Major performance boost for files with many convos (Case 66490)
- added — Delete work item assignments when a reviewer is removed from a review (Case 76684)
- added — RTC - automatic Work Item approvals (Case 71231)
- added — Respect max rows in printable and csv report versions (Case 76670).
- added — Invite emails mentioned in comment to Review (Case 76479)
- added — RTC - Add Collaborator reviewers as Work Item assignments (Case 73575)
- fixed — Checklist - If you enter two CL items with same name, you simply get kicked to a new screen with an error, lose work (Case 76056).
- fixed — 8.3 Triggers failing on locked clients on 2011.1 (Case 71642)
- fixed — Fixed server memory leak w/ large uploaded files. (Case 76672)
- fixed — 8.3 - Group Synch XML Functionality (Case 76644)
- fixed — Recognize emails with '+' (Case 76652)
- fixed — 8.3: Uploaded file chat message may link 'wrong' version of the file (Case 74198)
- fixed — Remove user agent XSS vulnerability (Case 76661)
- fixed — Search button gradient (Case 75989)
- fixed — Fix alignment of "Next Steps" and "Search" buttons (Case 76656)
- fixed — Explain existing Review load in 'Add to Review' wizard (Case 76651)
- fixed — Deadlines change automatically with timezones (Case 76640)

- fixed — 8.3 - Capturing connection errors deeper than one level(Case 76643)
- fixed — TEE clients were breaking when trying to get the properties or content of a deleted file (SF Case 8309)
- fixed — Support Directory in ClearCase(Case 69826, Review#9789)
- fixed — AddDiffs relative tag causing issues to arise in some cases when trying to add comments/defects in the diff viewer (SF case 8413)
- fixed — roles who are not allowed to modify the general information section of a review can now check/uncheck checklist items and modify participant custom fields (case 76610)
- fixed — deadline column not sortable on home page (case 76192)

## 8.2.8202 - October 29, 2013

- added — Support http://www.opensearch.org (Case 75699)
- added — Decouple "eval formulas" detail toggle and option (Case 75243)
- added — "Home" link on the dashboard refreshes action items (Case 75323)
- added — Support for Excel ".xlsm" file format (Case 75461)
- added — Sign/decline reviews in Eclipse (Case 74648)
- added — Command-line scripting commands for signatures (Case 74648)
- added — Add "Signature Status" to review list reports (Case 74649)
- added — View for reporting on Electronic Signatures (Case 74649)
- added — Create Views in embedded database (Case 74150)
- added — Show signature status in 'ccollab admin review-xml' (Case 74649)
- added — Show signature status in Review Detail Report (Case 74649)
- added — "Signed/Pending signature/Declined by" filter to reports (Case 74649)
- fixed — Reduced memory usage, improving server performance. Should reduce OOM errors with large document reviews. (Case 75814)
- fixed — Eclipse plugin: fail to create new review. (Case 76312)
- fixed — Display more detail about web upload errors (Case 75990)
- fixed — Reduce db queries for Review Summary page (Case 76239)

- fixed — Search cannot find drop-down custom fields in Embedded database (Case 75833)
- fixed — Show "N/A" instead of "null" when deadlines disabled (Case 76000)
- fixed — Grey disabled list options in IE8 (Case 75998)
- fixed — Load "existing reviews" in GUI just-in-time (Case 75606)
- fixed — Link community section to active forum (Case 75668)
- fixed — Better error message converting old Excel files (Case 75317)
- fixed — Reordering Excel sheets shows bogus modifications (Case 75937)
- fixed — Checklist - Add more line items functionality (Case 75209).
- fixed — RTC - Reviews are created for work items with no changesets (Case 74785)
- fixed — Add possibility to add group admins in group sync and fix documentation for group sync command (Case 75608)
- fixed — RTC - Cannot upload new changes from Work Item when previous review is deleted (Case 72633)
- fixed — Command-line help displays args in wrong order (Case 75743)
- fixed — Synergy - Make Windows-only parameter -n optional (Case 75851)
- fixed — Text diff options do not work after changing "Wrap Lines" (Case 75614)
- fixed — RTC: Reviews are generated for the work item creator and not the owner (Case 73064)
- fixed — Preserve review pool status for groups in group-sync when the review pool flag is not given again (Case 75562)
- fixed — Materials with the same name from different components are munged together (Case 75454)
- fixed — Only showing one tab in Spreadsheet diff (Case 75369)
- fixed — Changelist names and descriptions are wrong (Case 75061).
- fixed — Electronic Signature enabled, signatory not included in review participants (Case 75512)
- fixed — Changelist names and descriptions are wrong (Case 75061).
- fixed — Link downloads wrong version in diff viewer (Case 75738)
- fixed — Make web UI buttons look less "flat" (Case 75319)

- fixed — Review Summary page scrolls when checklist clicked (Case 75148)
- fixed — Overview shows participant count of previous review (Case 74904)
- fixed — Zoom image diff correctly on window resize (Case 75371)
- fixed — Make web UI buttons less "flat" (Case 75319)
- fixed — Better error message on old Excel formats (Case 75317)
- fixed — Show participant signature status in Eclipse (Case 74648)
- fixed — checklists with spaces make review-xml malformed (Case 75491).
- fixed — Retain leading whitespace in comments (Case 74970).
- fixed — Checklist - Add more line items functionality (Case 75209).
- fixed — Triggers failing on locked clients on 2011.1 (Case 71642)
- fixed — Suppress "encode illegal character" warning (Case 74809)
- fixed — "View As" label sometimes misaligned (Case 75170)
- fixed — Re-layout custom field sections on show/hide (Case 74406)
- fixed — RTC - Error uploading deleted file when file already exists on review and has comments (Case 75251)
- fixed — Unused column in admin/users view (Case 74884).
- fixed — Unused column in admin/users view (Case 74884).
- fixed — Navigate to next/prev diff in spreadsheets (Case 74961)
- fixed — Materials sections columns collapse in "compressed" mode (Case 74469)
- fixed — Commit Comments not displayed for ClearCase(Case 75806)
- fixed — Commit Comments not displayed for ClearCase(Case 75806)
- fixed — Problem in 8.2 when adding a changelist and a folder was modified but not moved (case 75556)
- fixed — comment text wrapping problems(Case 75598).
- fixed — Error uploading changelist in subversion when NodeKind was invalid (case 75444)
- fixed — Support a version-extended base filename for Clearcase Addversions(Case 69562, fix unit test)

- fixed — Eclipse - Review Pools do not show up in list of addable participants in Eclipse(Case 70282, fix unit test)
- fixed — reviews sometimes showing up multiple times in action items after a template change (case 75216)
- fixed — 8.2.8200 had upgrade issues with Oracle backed servers (case 75426)
- fixed — Allow upload of multiple changelists in web UI (Case 71476)
- fixed — can now add TFS shelvesets with a leading or trailing space (case 73868)
- fixed — Users are now allowed to accept if they have opened a defect in a chat thread and users can no longer accept the same thread multiple times in a row (case 72236)
- fixed — creating custom fields no longer automatically adds them to a template (case 74089)
- fixed — Participant status no longer switches to 'Approved' upon review cancellation or rejection, the status stays as it was (case 73163)
- fixed — Support a version-extended base filename for Clearcase Addversions(Case 69562, Review #9462)
- fixed — you can no longer set the maximum value for the length of a custom field to be less than the minimum value (case 74440)
- fixed — invalid subscriptions are no longer allowed to be created, added diagnostic to remove invalid subscriptions already in place (case 68450)
- fixed — Support a version-extended base filename for Clearcase Addversions(Case 69562)
- fixed — Eclipse - Review Pools do not show up in list of addable participants in Eclipse(Case 70282)

## 8.2.8201 - August 28, 2013

- fixed — Corrected a field name that prevented Oracle databases from starting with 8.2.8200 (Case 75426)

## 8.2.8200 - August 27, 2013

- added — Excel Support
- added — Checklist Support
- added — Make custom fields mandatory by a phase in the review (Case 71847)
- added — RTC - Implement switch to add reviewers to review from work item assignments (Case 73574)
- added — Git support to "ensure-reviewed" and "ensure-review-started" triggers (Case 68627)

- added — RTC - Indicate which work item types under which states should trigger generating/ updating review (Case 73572)
- added — Installer includes Active Directory authentication option (Case 73443)
- added — RTC - Allow multiple Collaborator servers per RTC server (Case 73571)
- added — Verify Database Schema now run at server start (Case 69166)
- added — Diagnostics now add their results to the log file. (Case 69166)
- fixed — Perforce: Error when submitting a deleted symlink (Case 74037).
- fixed — Clicking file convo location links with spaces do not focus&scroll convo in diff viewer (Case 71652).
- fixed — Collaborator: Community section should be disabled by default (Case 74924)
- fixed — "Next Steps" shows too many buttons in Planning with review pools (Case 70425)
- fixed — Custom Defect Field Description Placement (Case 70746)
- fixed — Problem opening reviews due to invalid role ID (Case 70705)
- fixed — Review Summary - View separate - Status shows 'Initial' on all changelists (Case 70703).
- fixed — Can add user to review even when user does not have access (Case 73065)
- fixed — 'local' modifier is broken for perforce addversions (Case 73758)
- fixed — Defect Custom Fields Ignore Default Value Params (Case 74591).
- fixed — Workflow should stay in Rework until Author kicks it out (Case 65518)
- fixed — Group and Template inputs can only display 25 elements (Case 73546).
- fixed — Supportability: Set default lazy-upgrade value to '1' instead of blank (Case 74602).
- fixed — Changelists rolled up under 'View as: Separate' (Case 74174)
- fixed — no padding around community section. (Case 74288)
- fixed — do not block web UI while triggers running (Case 69159)
- fixed — Triggers run after database transaction complete (Case 74260)
- fixed — Possible deadlock promoting convos on documents (Case 73356)
- fixed — Add pref to disable decorating workspace files (Case 72153)

- fixed — Error opening diff viewer in Eclipse 4.3 Kepler (Case 74171)
- fixed — Cannot open diff viewer in Eclipse 4.x (Case 70492)
- fixed — Validate Automatic Link regex works in Javascript (Case 73901)
- fixed — Handle Automatic Link errors (Case 73901)
- fixed — Incorrect Quoting in ExternalDiffLauncher (Case 72222)
- fixed — Action items do not have urls if "external URL" not specified (Case 70688)
- fixed — Error uploading files with HTTPS and Proxy (Case 73807)
- fixed — Hide "take review pool" icon for non-pool participants (Case 72882)
- fixed — Cannot reject review with certain "reasons" (Case 73549)
- fixed — Both - Upload P4 shelved files from Web UI uploads empty diffs when there are pending files in the changelist. (Case 70421)
- fixed — P4 Passwords are output in plaintext by DebugCommandRunner (Case 73062)
- fixed — Web debug log stack traces not deobfuscated correctly (Case 73595)
- fixed — directory contents after a directory move not being shown when adding subversion revisions, initial commit. (Case 73161)
- fixed — Review Summary - View separate - Status shows "Initial" on all changelists (Case 70703)
- fixed — Page keeps reloading if fails "single server" check (Case 73467)
- fixed — Subversion triggers respect "Restrict Access" rules (Case 72912)
- fixed — Perforce triggers respect "Restrict Access" rules (Case 72912)
- fixed — Report on Review access restrictions (Case 73202)
- fixed — do not log scary warning about client log file (Case 67725)
- fixed — Uploading .pdf, .doc, and .docx files leaks memory. (Case 72595)
- fixed — Stuck at dancing bear if try to go "back" (Case 72007)
- fixed — Improved rendering speed for large text diffs (Case 72135)
- fixed — Disable "Email Everyone" when no one to email (Case 72710)
- fixed — Action items error when Review Pool user cannot access Review (Case 73200)

- fixed — Check Permissions for Report & Version access (Case 73040)
- fixed — Community section has misleading text when cannot show online version (Case 72940)
- fixed — Error leaving Review in Phase with invisible PCF (Case 74333)
- fixed — Defect log entries should open latest version of file (Case 73748)
- fixed — Defaults for custom fields are now stored properly so changes in a default do not change the history of reviews. (case 73999)
- fixed — Addchanges from the GUI client now works with SVN 1.8 (case 73980)
- fixed — CCRC 8.0 plugin support (Case 72806)
- fixed — review-xml now does not show metrics section if user does not have access to reports (case 70738)
- fixed — review, version, and changelist data accessable even when there is no user by navigating to data/urls. Initial commit (case 70738)
- fixed — Edit participant custom fields on Review Summary page (Case 72275)
- fixed — review-xml now does not show metrics section if user does not have access to reports (case 70738)
- fixed — review, version, and changelist data accessable even when there is no user by navigating to data/urls. Initial commit (case 70738)
- fixed — Review summary error with review pools and restrict access (Case 73201)

## 8.1.8100 - May 29, 2013

- Initial release of CodeReviewer and CodeReviewer Pro
- added — daily automatic backups for HSQLDB (Case 70617)
- added — FREE, PRO, Collaborator: Community page (Case 71082)
- fixed — "Recent Participants" links not displayed in Eclipse (Case 72881)
- fixed — Recent participants section not showing recent participants when 'All Users' Group is selected (Case 72022)
- fixed — Confusing version information for added files (Case 70637)
- fixed — Before and After not displaying correctly (Case 72263)
- fixed — Typing in comment box slow on IE8 (Case 72402)

- fixed — No markers displayed on PDF file when lots of defects are added (Case 72551)
- fixed — Prevent Cross-site-scripting attacks by correctly escaping parameters before re-displaying on some pages (Case 72282)
- fixed — Prevent cross-frame-scripting attacks by adding header X-FRAME-OPTIONS: SAMEORIGIN to HTTP Responses (Case 72283)
- fixed — Typing in comment box slow on IE8 (Case 72402)
- fixed — Document Review - Pushpins in wrong location (Case 70455)
- fixed — FREE: Free version links to support phone and email(Case 72904)
- fixed — FREE: Admin page "Additional Features" list is difficult to read(Case 72794)
- fixed — CodeReviewer - GUIClient shows all SCM instead of git and SVN only.(Case 72734)
- fixed — CodeReviewer: The "My Activity" button needs to be disabled.(Case 72736)
- fixed — CodeReviewer - Version Control: available features that should not be.(Case 72702)
- fixed — DiffViewer throws exception when Empty Document file(.docx) in version 0 (Case 72191)
- fixed — review not changing state to approved after reviewers were moved to observers (case 72023)
- fixed — Error Viewing PDF Documents After Upgrade(blank page pdf) (Case 70154)
- fixed — corrupted VSS diffs hang client (case 71910)
- fixed — TFS gated build changesets generate 'Missing changeset date' error (case 71790)
- fixed — NoClassDefFoundError creating TFS shelvesets (cases 71641, 71771)
- fixed — do not npe if changelist is missing (cases 71068, 71767)
- fixed — multiple issues when communicating with newer Subclipse clients using SVN 1.7 (case 70330)

## 8.0.8003 - April 15, 2013

- added — Detect OpenJDK (Case 71563)
- fixed — Review: Materials and Participant Section break after user takes review pool. (Case 71711)
- fixed — NPE when adding a review pool and mandatory subscriptions are enabled (case 71395)

- fixed — GUI client runs out of window handles (case 70850, case 71368)
- fixed — Increase default PermGen space to alleviate OutOfMemory errors (Case 70849)
- fixed — warn when participants cannot access review (Case 70794)
- fixed — Hidden mandatory custom field error trying to Begin Review (Case 71719)
- fixed — Renaming Root.xml does not work in 8.0 (Case 71591)
- fixed — Error uploading .docx to review (Case 66861, Case 70376)
- fixed — Could not Begin Review when there is invisible custom field selected (Case 71451)
- fixed — phase view does not include new 7.0 phases (case 70869)
- fixed — Issue with "Done Editing" (Case 71033)
- fixed — CCRC 8.0 and CCRC getVersions support(Case 66839)
- fixed — client installer effectively freezes when installing p4 tools under certain conditions (Case 70975)
- fixed — disable review pools in "participants" drop-down if not allowed to add (Case 70424)
- fixed — Review Materials Section: Columns are not lining up (Case 70046)
- fixed — HCF with no valid values causes error (Case 71184)
- fixed — Errors when adding a changeset that contains an "evil twin" scenario (Case 71298)
- fixed — Could begin Review even when the required custom Fields has no value yet (Case 70976)
- fixed — Drop-down series custom field name overflows column (Case 71032)
- fixed — better error message for Moved Temporarily / Moved Permanently errors while accessing /contentupload (Case 70157)
- fixed — do not prompt user to enable debug mode (Case 69595)
- fixed — custom fields replicate when navigating from home screen and reviews (Case 70968)
- fixed — Update changelist trigger needs to run as change-submit (Case 70793)
- fixed — Problem adding this Word document to a review (Case 69591)
- fixed — Line numbers overlapping in diff viewer (Case 70302)
- fixed — Action Item missing link if External URL not configured (Case 70688)

- fixed — Fully reset search box when going to next file (Case 70831)
- fixed — Use "Enter" to navigate to next search result (Case 70319)
- fixed — do not scroll to center when clicking convo (Case 69780)
- fixed — Truncate phone number with ellipsis if too long (Case 70393)
- fixed — Better tolerance for missing or corrupted documents (Case 69706)
- fixed — Error using Group-based access restrictions when Review not associated with Group (Case 70530)
- fixed — Cannot submit error report from login or "upgrade db" views (Case 70573)
- fixed — Update browser window title when review title changes (Case 70207)
- fixed — Make Drop-down series custom fields horizontal (Case 70218)
- fixed — Problem adding this Word document to a review (Case 69591)
- fixed — Line numbers overlapping in diff viewer (Case 70302)
- fixed — Review Subs - Able to remove Mandatory do not Enforce subs from reviews (Case 70404)
- fixed — Stack overflow caused by circular dependency on version history (Case 70562)

## 8.0.8001 - March 5, 2013

- NEW — Merged CodeCollaborator and PeerReview Complete as Collaborator!
- added — The Review Summary screen has now replaced the old create and edit review screens (Case 62189)
- added — You can now search .doc and .docx files in the diff viewer
- added — Review pools: groups of users that can be assigned to 'take' a participatory position in a review
- added — Add revert control to fields on Review General Information in edit mode (Case 69589)
- added — Edit Review: Add updating spinner next to each field and update revert control image (Case 69796)
- fixed — Removed RPM installer support (Case 69471)
- fixed — Edit participant issues with disabled templates (Case 70280)

- fixed — Edit review - template selection - will not save change if template names are same (Case 70284)
- fixed — Defect HCF custom field area has strange red bar (Case 70105)
- fixed — Group select too short (Case 70187)
- fixed — "Done Editing" participants button does not work after going to file (Case 70061)
- fixed — Line up the Files and Defect boxes under the left edge of the Cancel button (Case 70172)
- fixed — Error sorting table by clicking header (Case 69987)
- fixed — Auto-save causes cursor to jump on text fields (Case 70063)
- fixed — The Edit buttons for general info and participants section do not line up (Case 69894)
- fixed — Better tolerance for missing or corrupted documents (Case 69706)
- fixed — Review Summary - No warning when changing review templates (Case 69805)
- fixed — Eclipse & RTC - Find and replace CodeCollaborator & PeerReview Complete logos and icons (Case 69813)
- fixed — Error refreshing diff viewer after changing versions (Case 69892)
- fixed — Bogus "pending updates" warning after editing title (Case 69662)
- fixed — "Download Diff" button returns backwards diff (Case 69869)
- fixed — Log warnings about missing document fonts (Case 69799)
- fixed — Page up/down should scroll the diff viewer (Case 66640)
- fixed — Error when clicking through diff viewer before file fully loaded (Case 69637)
- fixed — Edit button in participant section needs to respect whyNotAllowedToModifyReview (Case 69534)
- fixed — Edit review general info: validation issue with groups when the group select is not visible (Case 69765)
- fixed — Parts of the general info edit drop-downs not clickable (Case 69664)
- fixed — Calculate Download/Upload menu positions dynamically (Case 69593)
- fixed — Autocomplete URL with default protocol when Attaching URL's (Case 69663)
- fixed — "Directory Found not in Git Repo" when "git" not in PATH (Case 69434)

- fixed — Remove bogus 'lastCommentCreationDate' log warning (Case 69412)
- fixed — 'Missing changeset Items' error parsing TFS changeset (Case 67093)
- fixed — Review Summary - Hierarchical custom fields are not hierarchical anymore(Case 69986)
- fixed — Begin Review should be disabled when General Info wait for input of the required fields(Case 69927, Review# 8846)
- fixed — Review title cannot be null or over 255 characters long(Case 69850, Review #8819)
- fixed — Document CollabFormAuthenticator's new functionality (Case 69887)
- fixed — Review should not move to next phase if still in edit mode for either the general info section or the participants section(Case 69913)
- fixed — Better error when uploading a URL/changelist to a completed review(Case 69851)
- fixed — Template list contains disabled ReviewTemplates(Case 69787, Review #)
- fixed — Author subscription user dropdown not working(Case 69783)
- fixed — "There is no current object" and other document problems (Case 69591)
- fixed — Admin Review Templates - cannot create a new template(Case 69792)
- fixed — Admin - More intuitive UI for Review Template edits(Case 69011, Review #8687)
- fixed — Client - always use configured URL to contact the server(Case 68993, Review#8698)
- fixed — Fixed — Client - always use configured URL to contact the server(Case 68993)
- fixed — Increased supportability for various SCMs by removing the word 'optional' when users are inputting the source path in the GUI client (case 61880)
- fixed — NPE in p4 implementation when a filepath is expected to be a symlink and it is not (case 64001)
- fixed — Missing or wrong changelist "external diff" button (Case 68873)
- fixed — Admin - More intuitive UI for Review Template edits(Case 69011)
- fixed — When in "Side by Side", "After" and "Before" drop-down text do not change properly (Case 68917)
- fixed — Adding shelvesets from Team Explorer now has expected behavior (case 67093)

## 11.2.7 Version 7

### 7.3.7304 - July 2, 2013

- fixed --- Incorrect Quoting in ExternalDiffLauncher (Case 72222)
- fixed --- Action items do not have urls if "external URL" not specified (Case 70688)
- fixed --- Error uploading files with HTTPS and Proxy (Case 73807)
- fixed --- Both - Upload P4 shelved files from Web UI uploads empty diffs when there are pending files in the changelist. (Case 70421)
- fixed --- P4 Passwords are output in plaintext by DebugCommandRunner (Case 73062)
- fixed --- Web debug log stack traces not deobfuscated correctly (Case 73595)
- fixed --- Page keeps reloading if fails "single server" check (Case 73467)
- fixed --- Subversion triggers respect "Restrict Access" rules (Case 72912)
- fixed --- Perforce triggers respect "Restrict Access" rules (Case 72912)
- fixed --- Report on Review access restrictions (Case 73202)
- fixed --- do not log scary warning about client log file (Case 67725)
- fixed --- Uploading .pdf, .doc, and .docx files leaks memory. (Case 72595)
- fixed --- Stuck at dancing bear if try to go "back" (Case 72007)
- fixed --- Check Permissions for Report & Version access (Case 73040)
- fixed --- Typing in comment box slow on IE8 (Case 72402)
- fixed --- Errors when adding a changeset that contains an "evil twin" scenario (Case 71298)
- fixed --- better error message for Moved Temporarily / Moved Permanently errors while accessing /contentupload (Case 70157)
- fixed --- do not prompt user to enable debug mode (Case 69595)
- fixed --- Update changelist trigger needs to run as change-submit (Case 70793)
- fixed --- Problem adding this Word document to a review (Case 69591)
- fixed --- Line numbers overlapping in diff viewer (Case 70302)
- fixed --- do not scroll to center when clicking convo (Case 69780)

- fixed --- Error using Group-based access restrictions when Review not associated with Group (Case 70530)
- fixed --- Cannot submit error report from login or "upgrade db" views (Case 70573)
- fixed --- Error sorting table by clicking header (Case 69987)
- fixed --- Better tolerance for missing or corrupted documents (Case 69706)
- fixed --- Error refreshing diff viewer after changing versions (Case 69892)
- fixed --- "Download Diff" button returns backwards diff (Case 69869)
- fixed --- Log warnings about missing document fonts (Case 69799)
- fixed --- "Directory Found not in Git Repo" when "git" not in PATH (Case 69434)
- fixed --- Remove bogus 'lastCommentCreationDate' log warning (Case 69412)
- fixed --- 'Missing changeset Items' error parsing TFS changeset (Case 67093)
- fixed --- 'Missing changeset Items' error parsing TFS changeset (Case 67093)
- fixed --- review-xml now does not show metrics section if user does not have access to reports (Case 70738)
- fixed --- review, version, and changelist data accessable even when there is no user by navigating to data/urls. Initial commit (Case 70738)
- fixed --- corrupted VSS diffs hang client (Case 71910)
- fixed --- TFS gated build changesets generate 'Missing changeset date' error (Case 71790)
- fixed --- NoClassDefFoundError creating TFS shelvesets (Cases 71641, 71771)
- fixed --- do not npe if changelist is missing (Cases 71068, 71767)
- fixed --- phase view does not include new 7.0 phases (Case 70869)
- fixed --- multiple issues when communicating with newer Subclipse clients using SVN 1.7 (Case 70330)
- fixed ---"There is no current object" and other document problems (Case 69591)
- fixed --- Increased supportability for various SCMs by removing the word 'optional' when users are inputting the source path in the GUI client (Case 61880)
- fixed --- Missing or wrong changelist "external diff" button (Case 68873)
- fixed --- When in "Side by Side", "After" and "Before" drop-down text do not change properly (Case 68917)

- fixed --- adding shelvesets from Team Explorer now has expected behavior (Case 67093)

## 7.3.7303 - January 18, 2013

- added --- Syntax highlighting for .hss, .ss and .ts files
- added --- Syntax highlighting for ECMA script (.es) files
- fixed --- Files showing wrong upload order (Case 66561)
- fixed --- Generic error popup when not allowed to access review (Case 67902)
- fixed --- missing spaces on review summary page (Case 67496)
- fixed --- RTC: Uploading a merged changeset results in incorrect version diff (Case 67862)
- fixed --- Flat vs compressed tree (Case 62144)
- fixed --- Next/Prev file button troubles in new version (Case 68084)
- fixed --- Compact view causes word wrap in chat panel to not work (Case 68589)
- fixed --- Error editing partially reworked changelist in separate view (Case 68506)
- fixed --- better error handling for ReviewAccessException (Case 68200)
- fixed --- Stuck at dancing bear after leaving diff viewer (Case 67949)
- fixed --- Chat icons not showing in materials section (Case 68184)
- fixed --- Materials section does not refresh correctly after review event (Case 68382)
- fixed --- Diff Viewer - Prev/Next buttons do not see next diffs when a comment is made on the first diff(Case 67680)
- fixed --- Conversation toolbar disappears when you select different line during defect creation (Case 67931)
- fixed --- Change user agent matcher in Browser.java to match new Gecko version string (Case 67979)
- fixed --- Errors after leaving diff viewer before diff is loaded (Case 67949)
- fixed --- Use ticket instead of user/pass if user/pass is passed in (Case 67749)
- fixed --- Cannot get electronic signatures to consistently work (Case 66367)
- fixed --- Enabling electronic signatures adds completed reviews to action items (Case 68587)
- fixed --- Not using External URL in some places (Case 68153)

- fixed --- Diff Viewer - Next/Prev Line keyboard shortcuts do not work (Case 68218)
- fixed --- Diff viewer Add Comment and Add As Defect buttons should be disabled if create not allowed (case 68373)
- fixed --- Add "external name" to custom defect list reports (Case 56067)
- fixed --- Error showing new version in diff viewer (Case 68219)
- fixed --- Diff Viewer - Prev/Next buttons do not see next diffs when a comment is made on the first diff (Case 67680)
- fixed --- AddVersion(Clearcase) from command line does not work in 7.3 (Case 67726)
- fixed --- Loading port monitor dll from install directory (case 67338)

## 7.3.7302 - November 21, 2012

- fixed --- Add time zone info to dates in UTC reports (Case 67610)
- fixed --- Unread comments improved for accessibility features (Case 67648)
- fixed --- Clicking 'Create Bug in Bug Tracker' should open in a new tab (Case 65863)
- fixed --- Able to modify roles without error (Case 67348)
- fixed --- Allow some copy/paste functionality in text diff viewer (Case 65946)
- fixed --- Blended reviews (Case 67514)
- fixed --- Handling for symlinks in perforce (Case 67550)
- fixed --- UI limit expanded beyond 10,000 users (Case 66400)
- fixed --- diff viewer now shows side-by-side diffs of identically versioned text files (Case 65933)
- fixed --- Print To Review no longer clips left margin (Case 66562)
- fixed --- Debug logging no longer creates infinitely huge log files in temp (Case 65617)
- fixed --- Scrolling on Firefox improved (Case 67051)
- fixed --- Synchronized locking code to avoid overlapping file lock (Case 66667)
- fixed --- "Show More Lines" button behavior corrected when used with wrap lines (Case 67052)
- fixed --- Password showing in GUI client in plaintext (Case 66089)

- fixed --- Oracle upgrade issues resolved (Case 67040)
- fixed --- User_lastactivity updates while user idles on the home screen (Case 66986)
- fixed --- Corrected NPE with action items if the user had no items(Case 65565)
- fixed --- Changed log level from 'warn' to 'error' for some DB migration issues (Case 64473)
- fixed --- Updated 'addardiffs' command documentation to cover AccuRev limitations (Case 66130)
- fixed --- Catch error from P4 when files are not opened for edit (Case 66901)
- fixed --- Corrected AccuRev error when file system root is inaccessible (Case 66503)
- fixed --- Resize comment box as you type (Case 65616)

## 7.3.7301 - October 26, 2012

- Added --- Compact view options for Review Summary and Diff Viewer pages.
- Added --- Improved content information on collapsed sections of Review Summary page
- Added --- Authenticated SMTP support
- Added --- Easy, in-application upgrade path from CodeCollaborator to PeerReview Complete
- Added --- Support for Visual Studio 2012 and Windows 8
- fixed --- Oracle upgrade issue from 7222 to 723 (Case 67040)
- fixed --- Reduced load on database from diff viewer (Case 65373)
- fixed --- Reject review reason is no longer a custom field (Case 61600)
- fixed --- Surround 'Unexpected empty version for path' error handling (Case 66261)
- fixed --- Several IE bugs in diff viewer (Case 65579)
- fixed --- git addchanges now works with unchanged, managed files (Case 66417)
- fixed --- Accurev error handling when file system root is inaccessible (Case 66503)
- fixed --- Changing version in Eclipse compare editor no longer breaks connection (Case 65305)
- fixed --- Deleted files no longer show reverted icons in Review Materials section (Case 65580)
- fixed --- Reduced DB contention when updating user activity (Case 66273)

- fixed --- Support .DOC and .DOCX uppercase extension names (Case 66112)
- fixed --- Clearcase Server Integration (Case 64295)
- fixed --- Integration to CCRC with a Bad Password no longer causes lockout (Case 65551)
- fixed --- Pipe symbols in custom fields now work correctly (Case 66161)
- fixed --- Review Summary screen now displays while still loading materials
- fixed --- Auto-adding of unversioned directories now supported on newer subclipse clients
- fixed --- Eliminated a 'jump' in chat when working with defects (Case 65350)
- fixed --- Non-participant comments no longer highlighted until read (Case 65102)
- fixed --- Improved TFS date parsing by adding he_IL locale
- fixed --- Diff viewer refreshing when no longer visible (Case 66082)
- fixed --- Support for Synergy web mode (Case 63794)
- fixed --- TFS date parsing no longer fails for English India locale (Case 65220)
- fixed --- Handle null state passed by P4Version (Case 64548)
- fixed --- Resolve null pointer exception (Case 64653)
- fixed --- P4V and add pending changelist commands now grab shelved changes (Case 64460)
- fixed --- ClassCastException calling protects command (Case 63322)
- fixed --- P4 Error "Unicode clients require unicode enabled server is now resolved automatically (Case 64450)
- fixed --- Upgrade doc conversion utility (Case 64799)
- fixed --- AccuRev diff error "9002" or "9005" (Cases 65481, 65568, 65482)
- fixed --- Error uploading files with proxy to server not at root (Case 65078)
- fixed --- Clients now able to work on systems that do not support file locking (Case 65567)
- fixed --- Correct problems running multiple commands at one time (Case 56660)
- fixed --- Add template as one of the fields that can be added to a custom report (Case 64403)
- fixed --- Concurrent modification exception from Eclipse client (Case 65498)
- fixed --- Hide SVN "Require Client Cert Password" option on server (Case 65483)

## 7.2.7237 - August 16, 2012

- Added --- PDF document diff support! (PeerReview Complete, only)
- Added --- Multiple file upload support for IE from review summary page.
- Added --- Support additional file extensions for Gosu and xml file types (case 64942)
- Added --- "Synchronize Scrolling" option in non-wrapped side-by-side diff
- Added --- Added functionality to upload multiple files that works in IE (using Flash)
- Added --- Added another "mark all read" button at top of diff convos (Case 64110)
- Updated --- Known issues 978 for this release have been updated clearing out some of the beta notes and adding new information.
- fixed --- PrintToReview will now print documents for doc-diff that are more than 1 page (Case 65443, Case 65361)
- fixed --- ClassCastException calling protects command (case 63322)
- fixed --- Resolved null pointer exception on ensure-diffs-reviewed (Case 64653)
- fixed --- subclipse addfiles throwing nullpointer in some instances if auto add is enabled
- fixed --- make IE use standards mode, even on intranet sites (Case 65013)
- fixed --- Problems with running multiple commands at one time.(Case 56660)
- fixed --- better handling of web UI upload errors (Case 65036)
- fixed --- cursor position reset labeling new convo in IE (Case 65068)
- fixed --- error after switching to new version of file via popup (Case 64853)
- fixed --- pressing <enter> in login fields no longer tries to log in twice
- fixed --- Grabbing values set by P4CONFIG into the client connection except the ones we override (Case 64262)
- fixed --- Disable login button while waiting for login response
- fixed --- New users will not get created twice (Case 64356)
- fixed --- review details page now hidden if the user does not have access to reports (case 64940)
- fixed --- do not show system message in diff viewer (part 2 of Case 65013)

- fixed --- do not linkify javascript: and other unsafe URL's
- fixed --- trust admin-configured Automatic Links (Case 65011)
- fixed --- fixed issue trying to add a file with haveRev #none to a review (Case 64653)
- fixed --- StarTeam format diffs not auto-detected by 'ccollab adddiffs' (case 63719)
- fixed --- p4 submitoptions other than submitunchanged could not handle edit-less changelists (case 64897)
- fixed --- P4 Error "Unicode clients require unicode enabled server" is now resolved automatically (Case 64450)
- fixed --- Allow adding yourself to a review
- fixed --- Show better UI for license errors
- fixed --- add ability to enter license code when upgrading(Case 63831)
- fixed --- Preserve p4 jobs data in update-changelist trigger (Case 64252)
- fixed --- Making a comment with no label will no longer break the review screen (Case 64738)
- fixed --- Addhgdiffs no longer throws a null pointer if not inside hg repo
- fixed --- Error with defect custom field of type Drop-down Series (Case 64773)
- fixed --- Change "Character Encoding" option form radio group to select box
- fixed --- Chat History and Review Materials sections not displayed (case 64534)
- fixed --- Line number linkifying does not function(Case 57351)
- fixed --- StarTeam current diff revision should be specified by -cfgd (case 64255)
- fixed --- support cvs 1.12.9 date format (case 64129)
- fixed --- Added files in Surround committed changelists are blank (case 64149)
- fixed --- Setting system clock back no longer prevents login or logout (Case 64526)
- fixed --- old urls not pointing at new resource locations (Case 63149)
- fixed --- Installing RTC Eclipse plugin: Make HCF compatible with 6.5 plugin clients (Case 64232)
- fixed --- Home screen filter bug (Case 63804)
- fixed --- annotation dialog and defect text detail appear over login screen

- fixed --- modal error dialog does not "grey out" rest of screen
- fixed --- show "new version available" popup in diff viewer (Case 64268)
- fixed --- always list all available versions in diff viewer drop-down
- fixed --- git diff neglected deletes (Case 61431)
- fixed --- automatically refresh to work around race condition for fresh servers(Case 63598)
- fixed --- error clicking in diff viewer after adding more context (Case 65479)
- fixed --- subversion filenames with embedded at symbols (case 63425)
- fixed --- Cannot load obfuscated dumps (Case 63339)
- fixed --- Diff setting "latest accepted revision" seems to be broken (Case 62421)
- fixed --- Fix web mode parameter -s for Synergy (Case 63794)
- fixed --- allow global options to "ccollabtray" (for example, "ccollabtray --debug)
- fixed --- save zip of all log files when running ccollabgui in debug mode (Case 61874)
- fixed --- Line number linkifying does not function (Case 57351)
- fixed --- tweak appearance of diff viewer header (Case 64543)
- fixed --- requested file subscriptions now working
- fixed --- Action Items: Outgoing filter is not filtering outgoing(Case 64213 and Case 63804)
- fixed --- Support path which is both file and folder in web UI (Case 64265)
- fixed --- On the Review Summary page: selecting any "You are waiting and will be notified when" radio button: no longer causes IE7 screen to shake (Case 64243)
- fixed --- Selecting 'no file content' on a review dump no longer results in a zip that contains files (Case 64487)
- fixed --- The custom defect fields of a new defect no longer populate with the values of the previously created defect.(Case 63320)
- fixed --- lines wrap in text diff with "wrap lines" disabled (Case 64229)
- fixed --- text highlighting in single file no longer jumps around and highlights multiple line numbers (Case 63808)
- fixed --- do not skip changed lines when comparing a file verison with itself (Case 64076)
- fixed --- popup a warning and fallback if local storage quota exceeded (Case 64270)

- fixed --- Support "email all" with Outlook 2003 and 2007 (Case 63212)
- fixed --- Diff Viewer: Pusphins show over version drop down (Case 64317)
- fixed --- Send debug log message appears behind drag and drop screen. User cannot send debug log (Case 63883)
- fixed --- use admin-configured support email address (Case 61909)
- fixed --- Cannot rename custom fields. (Case 64109)
- fixed --- New pushpins not visible in the other version if on same page but off screen (Case 64312)
- fixed --- do not offer to reset password if using LDAP authentication (Case 64167)
- fixed --- making selected marker visible when hiding the rest (Case 64250)
- fixed --- Added files in Surround committed changelists are blank (case 64149)
- fixed --- Displaying Review Materials for large reviews is now faster. (Case 64090)
- fixed --- Participants section does not use up available width on FireFox (Case 64089)
- fixed --- navigate document diffs with prev/next button

## 7.2.7218 - June 25, 2012

- PeerReview Complete 2012 - Official release!
- Added --- Diff Viewer 364 completely overhauled with new and improved display and navigation options.
- Added --- Electronic signatures 363 (PRC only)
- Added --- .doc and .docx diff support 383 with comment promotion (PRC only)
- Added --- More detailed performance logging in debug mode for web UI
- Added --- Image over-under Diff view (Case 62865)
- Added -- Page Width and Full Page zoom options. Zoom options now a ListBox (Case 61239)
- fixed --- Diff Viewer no longer produces exceptions with images in IE8 (Case 64346)
- fixed --- Doc diffs display correctly in IE (Case 64210)
- fixed --- Consecutive Git commits should not be seen as a rebase
- fixed --- Eliminated error when clicking 'External Diff' (Case 64120)

- fixed --- CcollabGui client hangs forever on mac when closing when changes added to review from P4V (Case 63049)
- fixed --- latest rebase should not trump latest accepted for LATEST_ACCEPTED preference (case 62421)
- fixed --- preserve scroll when going back to review summary from diff
- fixed --- Syntax highlighting failure means php file will not display in Code Collaborator (Case 63930)
- fixed --- Overlapping icons in defect log section (Case 63686)
- fixed --- Check type of dragged item in order to determine overlay display (Case 63882)
- fixed --- Subclipse NPE on file with Subversion missing status (case 63812)
- fixed --- mercurial auto detect always returns true if .hg exists on the path ignoring errors
- fixed --- Fix for move/delete and move/added (Case 62688)
- fixed --- Chat panel will not place focus on selected comment if it is located toward the top or bottom of the panel (Case 63891)
- fixed --- Modified check for fileapi support asking for specific functions supported (Case 63753)
- fixed --- Corrected version Descriptions to avoid 'latest upload' error (Case 63923)
- fixed --- Subclipse NPE on file with Subversion missing status (case 63812)
- fixed --- respect "display order" pref in diff viewer, refactoring (Case 63611)
- fixed --- Document diffs carry over to PDF and Image diff viewers
- fixed --- Error selecting modified lines in side-by-side (Case 63809)
- fixed --- no error popup for unexpected server errors
- fixed --- duplicate text diff context if user clicks too quickly
- fixed --- Fixed issues with the tooltip creation and with tooltips dissapearing when going to next/prev files (Case 63625)
- fixed --- Subversive integration does not upload deleted files
- fixed --- Changing labels for Over/Under transparency control (Case 63585)
- fixed --- always expand selected convo (Case 63591)

- fixed --- Fix for glitchy behavior in the drag and drop div when you hover content over the 'Drop files here' legend. (Case 63576)
- fixed --- Reviews can be created from Eclipse when using a proxy on the Eclipse network connection settings (Case 62217)
- fixed --- add "--p4client" to "ccollab ensure-content-reviewed" doc (Case 63413)
- fixed --- improved cache performance when loading Review Overview
- fixed --- improved performance of DiffCache, used when loading reviews
- fixed --- do not wrap defect edit buttons

## 7.2.7208 - May 29, 2012

- PeerReview Complete 2012 Beta
- Added --- New diff viewer
- Added --- Initial support for doc/docx diffs

## 7.1.7111 - May 11, 2012

- fixed --- list reports are now displayed when selected (Case 63480)

## 7.1.7110 - May 7, 2012

- Added --- new user pref for saving overlay/separate display (Case 62262)
- Added --- drag/drop support for reviews
- Updated --- Performance improvements for new UI
- Updated --- Changed review summary screen to support collapsible sections (Not in IE7)
- Updated --- Diff viewer chat panel is now condensed and easier to use
- fixed --- Updated font in diff viewer chat panel (Case 63281)
- fixed --- Still display "Flat" as a view option from the prefs->Display tab (Case 63270)
- fixed --- Typo in web UI "form" -> "from" (Case 63256)
- fixed --- Drag and Drop div is visible in IE7 (Case 63263)
- fixed --- Disabled collapsible sections in IE7 (Case 63032)
- fixed --- Hierarchical custom fields now work for defects(Case 62193)
- fixed --- Pushpin placement has been corrected (Case 61491)

- fixed --- The highlight for the selected chat (Chat blue borders) is now displayed properly (Case 63195, Case 63207)
- fixed --- Change chat comment text to 10pt font(Case 63199)
- fixed --- Users should be able to add themselves to reviews if they are a member of the associated group (Case 63013)
- fixed --- subclipse 1.8 throwing null pointer exception when attempting add to review (case 62794)
- fixed --- Show comment input fields only for the active conversation.(Case 63094)
- fixed --- Refresh review summary page when diff viewer changes
- fixed --- "Submit error" UI should display the created case number (Case 63031)
- fixed --- Participant setting Group By None should be saved(Case 62533)
- fixed --- Links are no longer removed from the chat log when the overlay vs separate widget is clicked (Case 62792)
- fixed --- Fixed participant headers alignment for all supported browsers and reduced vertical space used by the file table (Case 63020)
- fixed --- Make login page work with browser "Remember Password" functionality (Case 61024)
- fixed --- Update CCRC .jar filename in manual (case 63047)
- fixed --- Conversation selection in diff viewer steals clicks (Case 62243)
- fixed --- Error pops up when opening review (Case 62982)
- fixed --- http proxies do not like /contentupload (case 61786)
- fixed --- As a user the separate vs overlay slider is confusing when there is only one changelist.(Case 62583)
- fixed --- Add header tags to important sections of the page (Case 62691)
- fixed --- Replace sprites with image elements for meaningful images (Case 62575)
- fixed --- performance improvements for diffcache misses. (Case 62893)
- fixed --- Support MKS variant branches in sandbox (case 62577)
- fixed --- RTC server side plugin missing dependency in feature.xml (case 62670)

- fixed --- Rework status now displays correctly in both overlay and separate views (Case 61955)
- fixed --- preserve scroll position of active element when page updates
- fixed --- Improve performance of GWT pages (Case 62142)
- fixed --- Root context not respected from links on report page (Case 61943)
- fixed --- addchangelist now parses git log messages with timezones greater than UTC (Case 62442)
- fixed --- IE7 issued warning on every page load over HTTPS (Case 62228)
- fixed --- CollabExternalDiff viewer breaking if the version/changelist ID contained invalid filename characters (case 62184)
- fixed --- filtering outgoing by author (case 62148)
- fixed --- 7.x clients once again respect SSL/proxy settings. (Case 62248)
- fixed --- Improve response time of the SimpleTable widget with deferred rendering (Case 62142)
- fixed --- better tooltip for 'Add and Include Defect' button, and use fix Label/Text inconsistency in defect log (cases 62285, 62532)
- fixed --- browse command now correctly only opens browser when specifically called
- fixed --- Class Cast exception while checking if the default changelist was empty (Case 62079)
- fixed --- Ignoring trailing strings after perforce year.release-number string in order to retrieve support for shelveset correctly (Case 61812)
- fixed --- Added specific client to fstat and resolved commands when executing from a different client (case 61752)
- fixed --- Web UI inaccessible over SSL VPN such as Juniper IVE based SA (Case 62508)
- fixed --- Added dismiss button when review is completed and marked for commit. (Case 62261)
- fixed --- Root context not respected from links on report page (Case 61943)
- fixed --- external diff viewer not working on OSX (case 56192)
- fixed --- Grabbing p4 charset options correctly in command runner (Case 62349)
- fixed --- Highlight whole row of unread conversation in materials section

- fixed --- author should be able to begin review from Annotating phase (case 62401)
- fixed --- Download Latest Client Installer Link in dashboard view (Case 62007)
- fixed --- Added text to clarify which waiting option is selected. Fixed bug that showed wrong selection when browsing through different reviews (Case 62266)
- fixed --- Handle Line wrapping (Case 61829)
- fixed --- reduce download size of Review Summary page by ~65% for large Reviews
- fixed --- Fix accessibility issue: add "alt" property to images (Case 62154)
- fixed --- Disabling filters for action items and adding loading indicator when the table is yet loading (Case 62232)
- fixed --- Perforce "renumber changelists" script hangs (Case 62206)
- fixed --- Reports page SQL and CSV links now working in IE browsers (case 61805)
- fixed --- Changed check for unchanged files so it only checks if the client has any open files (Case 61949)
- fixed --- edits now work on change-content-trigger (Case 61958)
- fixed --- links in redacted comments now look more redacted (Case 61840)
- fixed --- Hidden error popups block "init database" screen in debug mode (Case 61636)
- fixed --- Review summary page performance improvements (Case 61501)
- fixed --- Line wrapping corrected in diff viewer (Case 61829)
- fixed --- Ignoring trailing strings after perforce year.release-number string in order to retrieve support for shelvesets correctly (Case 61812)

## 7.0.7027 - March 19, 2012

- fixed --- replaced accessibility tag information (Case 62154)
- fixed --- http proxies should now work with /contentupload (Case 61786)
- fixed --- renames causing TimeBuilder to error. (Case 62317, 62359)
- fixed --- Perforce "renumber changelists" script hangs (Case 62206)
- fixed --- Fixed typo in approve message (Case 62224)
- fixed --- No Admin phone number causes Menu options still in V6 format to disapper (case 61914)

- fixed --- Support Surround dates with '.' separators (case 62034)
- fixed --- Archive Content Cache should go away (Case 61901)
- fixed --- Added specific client to fstat and resolved commands when executing from a different client (case 61752)
- fixed --- Ignoring trailing strings after perforce year.release-number string in order to retrieve support for shelvesets correctly (Case 61812)
- fixed --- Typo in CC licensing message (Case 61582)
- fixed --- 7.0 UI: "An unexpected error occurred" should not be presented for user errors (case 61605)
- fixed --- Approving a review when unread comments are loaded causes an error (case 61660)
- fixed --- RTC plugin failed to create review with long title (case 61740)

## 7.0.7024 - February 17, 2012

- added --- UI tweaks for diff viewer chat panel (Requirements 101488)
- fixed --- Menu headers are now correctly aligned and "Home" image (Case 61820)
- fixed --- Added links to several manual pages for added GIT information (61679)
- fixed --- RTC plugin failed to create review with long title (case 61740)
- fixed --- Approve button is disabled when there is nothing more to do. (case 61710)
- fixed --- Error with svn 1.7 when user does not have access to repo root (case 61682)
- fixed --- User is erroneously prompted to configure full name if full name and login name match (Case 61639)
- fixed --- Display better error message in web popup
- fixed --- Installer appears to hang when DB port is set to empty (Case 61692)
- fixed --- Show better error message if database misconfigured (Case 61657)
- fixed --- Commit from TFS reviews in a single changeset (case 61723)
- fixed --- Diff Compare Dopdown Is Broken (case 61549)
- fixed --- User has 'You have not configured' action item if full name matches login name (Case 61639)
- fixed --- Login names with spaces in them get munged oddly on logout (Case 61329).

- fixed --- We do not have a reject review button in Eclipse plugin (Case 61432)
- fixed --- Changelists displayed in wrong order by automatic refresh (case 61411)
- fixed --- Show "Loading" indicator while loading large reviews
- fixed --- Messaging and documentation describing incompatibility with MKS on 64-bit Windows JRE (cases 60680, 60968, 61326, 61333)
- fixed --- Debug Logging is not capturable on the Initialize DB screen (Case 61352)
- fixed --- Changing phase of authors that upload files after review has been started (Cases 60947, 60720)
- fixed --- Reduce server load of refreshing Review Summary page
- fixed --- The review summary shows the materials in one order but the diff viewer considers the materials in a different order. (case 61465)
- fixed --- Minimally sized GUI screen does not show Existing Reviews field (Case 61018)
- fixed --- addgitdiffs fails with Unexpected Exception (NPE) when not used in a git repository (Case 61424)
- fixed --- Log "long query" warnings on prepared statements
- fixed --- New Server Fails to get License w/o feedback (Case 61023)
- fixed --- Eclipse plugin allows you to begin review without setting a mandatory custom field (Case 56850)
- fixed --- Support Surround one-line changelist history records (case 61311)
- fixed --- Ignore StarTeam 'no revision of file' errors (case 61212)
- fixed --- File Extension issue on Print to Review attached files (Case 61316)
- fixed --- Another minor display issue with Custom Field (Case 60946)
- fixed --- Command 'admin review finish' displays previous phase name (Case 61327)
- fixed --- Mandatory subscriptions allowing changing of roles and sending multiple notifications on attempts to remove mandatory users (case 60176)
- fixed --- do not clobber 'External URL' on startup (Case 61435)
- fixed --- Improper cache headers require manual refresh (Case 61435)
- fixed --- Support Surround one-line changelist history records (case 61311)
- fixed --- Display shorter timestamps in chat, with detail on hover

- fixed --- Suppress extra defect expando header outline (Case 61412)
- fixed --- Use configured git-exe path (case 61385)
- fixed --- Rework counter showing incorrect values (case 60943)
- fixed --- NPE when adding empty changeset to review with RTC (Case 61255)
- fixed --- Removing duplicate versions from the timeline (case 60337)

## 7.0.7022 — 7.0 Initial release! January 31, 2012

* FIRST RELEASE OF v7.0! *

Major features:

- A fully revised and streamlined user interface
- Hierarchal custom fields
- Support for Perforce shelvesets
- Support for Git changelists
- Support for ClearCase plugin using CodeCollaborator Eclipse plugin
- Option to provide users with option to reject a review
- Ability to log debug information from Eclipse plugin

- added --- automatically list local (not in upstream) git commits in GUI
- added --- gitaddbranch to automatically add diff of what is in a branch that is not upstream as one squashed diff
- added --- review security option to allow participants and group members (inclusive) access to reviews
- added --- added Defect log to the review summary page
- added ---remember Participants section "Group by" settings
- fixed --- now correctly parse SVN server certification error messaging (case 60790)

- fixed --- support ClearCase Zulu format date strings (case 60137)
- fixed --- now correctly parse svn server certification error messaging (case 60790)
- fixed --- eclipse plugin chooses wrong default previous version (Case 60409)
- fixed --- validate content on client before caching. (Case 60157)
- fixed --- creator can now begin reviews via eclipse or command line even if they are not an author (case 59171)
- fixed --- download Diff does not properly handle (ignore) binary files (Case 59631)
- fixed --- ignore CVS banner in stderr (Cases 59676, 60698)
- fixed --- blank screen when logging in for some users (case 60602)
- fixed --- error with Subversive/JavaHL when no access to repo root (Case 60265)
- fixed --- check version before executing shelveset commands (case 60569)
- fixed --- reworked file now shows correct status (Case 60025)
- fixed --- added functionality to delay review loading after executing changes page (Case 58880)
- fixed --- TFS en_CA date now parses correctly (Case 60231)
- fixed --- limit width of "Fun Facts" box (Case 59959)
- fixed --- Get content of non-local resources in Eclipse even if out of sync (Case 60129)
- fixed --- Eclipse plugin update now correctly supports all versions (Case 57778)
- fixed --- null pointer exception in Eclipse client (Case 58849)
- fixed --- different OS's can now add the same file to a review without causing problems due to different line endings (Case 59412)
- fixed --- "Suppress Notification" checkbox stays checked (Case 59164)
- fixed --- OOBE when changelist is empty (Case 61404, Case 61382)
- fixed --- Retain external URL on startup (Case 61453)
- fixed --- Eliminate manual refresh of cached headers (Case 61435)

## 11.2.8 Version 6

### 6.5.6510 — July 25, 2012

- added --- Support for additional file extensions for Gosu and xml file types (Case 64942)
- fixed --- Chat History and Review Materials sections display correctly (Case 64534)
- fixed --- Obfuscated dumps can now be imported (Case 63339)
- fixed --- Fix web mode parameter -s for Synergy (Case 63794)
- fixed --- Line number linkification corrected (Case 57351)
- fixed --- Support CVS 1.12.9 date format (case 64129)
- fixed --- Added files in Surround committed changelists are no longer blank (case 64149)
- fixed --- Syntax highlighting failure no longer means php file will not display in Code Collaborator (Case 63930)
- fixed --- Eliminated Subclipse NPE on file with Subversion missing status (case 63812)
- fixed --- TFS implementation correctly parsing Canada locale date (case 63288)
- fixed --- List reports correctly displayed (case 63480)
- fixed --- TEE shelvesets now supported (Case 60306)
- fixed --- 'Page 1 does not exist in this document' message opening Print To Review .pdf (case 61316)
- fixed --- Update CCRC .jar filename in manual (case 63047)
- fixed --- http proxies do not like /contentupload (case 61786)
- fixed --- Performance improvements for diffcache misses. (Case 62893)
- fixed --- CollabExternalDiff viewer works if the version/changelist ID contained invalid filename characters (case 62184)
- fixed --- Browse command works in batch when called with global option no-browser.

### 6.5.6508 — March 26, 2012

- fixed --- Fixed class cast exception while checking for empty default changelist(62079)
- fixed --- Ignoring trailing strings after perforce year.release-number string in order to retrieve support for shelvesets correctly (Case 61812)

- fixed --- Added specific client to fstat and resolved commands when executing from a different client (case 61752)
- fixed --- Reports page SQL and CSV links not working in IE browsers due to IE URL character limits (case 61805)
- fixed --- Changed check for unchanged files so it only checks if the client has any open files, to avoid error when there is none. Changed submitted version fstat check to include the version number. Added unit tests (Case 61949)
- fixed --- edits do not work on change-content-trigger. Also added test case (Case 61958)
- fixed --- Support Surround dates with '.' separators (case 62034)
- fixed --- renames no longer cause TimeBuilder to error. (Case 62317, 62359)

## 6.5.6507 — February 17, 2012

- fixed --- SVN 1.7 support when user does not have access to repo root (61682)
- fixed --- Commit from TFS reviews in a single changeset (Case 61723)
- fixed --- Messaging and documentation describing incompatibility with MKS on 64-bit Windows JRE (cases 60680, 60968, 61326, 61333)
- fixed --- Changing phase of authors that upload files after review has been started (Cases 60947, 60720)
- fixed --- Minimally sized GUI screen does not show Existing Reviews field (Case 61018)
- fixed --- New Server Fails to get License w/o feedback (Case 61023)
- fixed --- Another minor display issue with Custom Field (Case 60946)
- fixed --- Support Surround one-line changelist history records (case 61311)
- fixed --- Ignore StarTeam 'no revision of file' errors (case 61212)
- fixed --- rework counter showing incorrect values (case 60943)
- fixed --- NPE when adding empty changeset to review with RTC (Case 61255)
- fixed --- test case for TimelineGenerator fixed (case 60337)
- fixed --- Removing duplicate versions from the timeline (case 60337)
- fixed --- Mandatory subscriptions allowing changing of roles and sending multiple notifications on attempts to remove mandatory users (case 60176)
- fixed --- Erroneous Concurrency Exception during DB init. (Case 60682)

- fixed --- Now correctly parse svn server certification error messaging (case 60790)
- fixed --- Problem disabling admin account
- fixed --- Support ClearCase Zulu format date strings (case 60137)
- fixed --- Can no longer change the role of a user who is in a review due to mandatory subscription
- fixed --- Error parsing regional (Canada) date(Case 60137), Review # 6095.
- fixed --- Eclipse plugin chooses wrong default previous version. (Case 60409)
- fixed --- PeerReview client installer copies CCRC .jar files to wrong location
- fixed --- Validate content on client before caching. (Case 60157)
- fixed --- Svn URL's are now properly encoded if there are special characters within a filename

## 6.5.6505 — December 21, 2011

- added --- SVN 1.7 support (Case 59698)
- added --- Updated Eclipse support
- fixed --- Ignore CVS banner in stderr (Case 59676, 60698)
- fixed --- Blank screen when logging in for some users (Case 60602)
- fixed --- Error with Subversion/JavaHL when no access to repo root (Case 60265)
- fixed --- Updated documentation for null return in IP4Conn (Case 60569)
- fixed --- Check version before executing shelveset commands (Case 60569)
- fixed --- Reworked files no longer show status as reverted (Case 60025)
- fixed --- Properly support old style p4 renames (Case 60294)
- fixed --- Add to Review P4V integration no longer populates add to existing review (Case 58880)
- fixed --- Conversations added to Review Summary now update correctly
- fixed --- TFS en_CA data parses correctly (Case 60231)
- fixed --- Get content of non-local resources in Eclipse even when out of sync (Case 60129)
- fixed --- More logging exceptions are without stacktraces (Case 60041)

- fixed --- Eclipse plug-in versions now appear correctly (Case 57778)
- fixed --- Ignore warnings about EOL-style from Git (Case 60112)
- fixed --- Eclipse client no longer produces NPE when trying to load comments (Case 58849)
- fixed --- Suppress Notification checkbox stays checked (Case 59164)

## 6.5.6503 — October 25, 2011

- added --- Users can now ignored 'accepted' comments in comment count (Case 59072)
- fixed --- Adding the same file from different operating systems no longer breaks a review (Case 59412)
- fixed --- Support adding files from ClearCase view-private directories (Case 58637, 59735)
- fixed --- Corrected a parsing error in pending tasks (Case 57096)
- fixed --- Rework counter showing incorrect values (Case 59553)
- fixed --- VSS reviews will no longer hang (Case 59048)
- fixed --- Line parser out-of-bounds exception corrected.
- fixed --- P4 diffs correctly identifies before/after paths (Case 59166)
- fixed --- Support ISO formatted TFS dates (Case 59242)
- fixed --- Authors can no longer be disabled for reviews (Case 58629)
- fixed --- Browse now takes the user to the review materials page instead of new review page if a review exists (Case 58501)
- fixed --- Bad URL's no longer break the review material page (Case 57559)
- fixed --- Correctly respect symlinks in web UI for SCM configuration (Case 59277)
- fixed --- Will now select the correct ClearCase version when addversions has no previous (Case 58988)
- fixed --- Corrected PRC manual favicon (Case 58624)
- fixed --- admin batch version-spec now handles filenames with spaces (Case 58988)
- fixed --- Corrected issue with max number of reviewers (Case 58709)
- fixed --- Action items and search results now respect permissions (58315)
- fixed --- Updated GUI screenshots for Synergy (Case 51728)

- fixed --- Grammar error in server installer authentication page (Case55791)
- fixed --- Time spent reviewing is now correct when doing code reviews (Case 54486, Case 56510)
- fixed --- Open commit reminders from a deleted review will no longer send the user to a blank home screen (Case 59137)
- fixed --- Corrected poking behavior for completed reviews.
- fixed --- Version content already set bug

## 6.5.6502 — September 9, 2011

- added --- PeerReview Complete will now import database exports from CodeCollaborator as long as minor version/build numbers match.
- added --- Updated manual to include alternate SSL option (Case 58976)
- added --- Syntax parsing for Gosu
- added --- Configuration options for synergy web-mode.
- fixed --- Search results, even by review ID, will not display reviews that the user does not have permission to access (uservoice request)
- fixed --- Updated manual screenshot for Synergy (Case 51728)
- fixed --- Surround committed changelist paths do not start with '/' (Case 56730)
- fixed --- Grammar error in server installer authentication page (Case 55791)
- fixed --- Time spent reviewing is now correct. (Case 54486, Case 56510)
- fixed --- Home page will no longer be blank when a review with an open commit reminder is deleted (Case 59137)
- fixed --- Date in version history for 6.5.6501 now correctly shows as 2011.
- fixed --- Better sorting of case-only changes in Ensure Content Reviewed Trigger (Case 58645)
- fixed --- Correctly handle Surround en_GB dates and fixes for history command parsing (Case 58882)
- fixed --- "Unlimited" number of reviewers is now limited to 100,000. (Case 58709)
- fixed --- Zooming in on a very large PDF that strains memory limits will now produce a friendly error message. (Case 57205)
- fixed --- Button text should now be consistent across reports. (Case 58794)

- fixed --- Accurev uploads should no longer complain about an A:\ drive. (Case 58982)
- fixed --- Updated System Status forms to remain within allotted space (Case 57616)
- fixed --- Print To Review 'File does not exist' error. (Case 58931)
- fixed --- No longer prevented from marking comments read/accepted when two comments collide on the same line of a diff. (Case 57504)
- fixed --- Users are no longer blocked from editing a review after it is due. (Case 53495)
- fixed --- Report Access Security no longer fails with Oracle. (Case 58794)
- fixed --- Fixed a file upload issue with TFS that had files showing up as if they had no content. (58738)
- fixed --- PRC server installer links to the correct online manuals. (Case 57677)
- fixed --- Rework/rebase counter on review overview page. (Case 57203)
- fixed --- Custom defect names should now show up everywhere that the defect label is used. (Case 58524, Case 57569)
- fixed --- Next button should now work when looking for defects. (Case 58542)
- fixed --- Added diagnostic to eliminate dangling user references. (Case 58213)
- fixed --- VHDL syntax highlighting.

## 6.5.6501 — August 22, 2011

- added --- ClearCase Remote Client (CCRC) Support 667.
- fixed --- File upload issue resulting in empty files.
- fixed --- NPE adding TFS files that have empty comments to review (Cases 58196, 58397, 58411)

## 6.5.6500 — July 22, 2011

* FIRST RELEASE OF v6.5! *

Major features:

- Report Access Security 187 features
- Improved file upload support

- added --- New vmoptions to adjust when review auto-update is disabled.
- added --- Users can now add changelists to a review if the regex matches the review (Case 57989)
- added --- Support for Eclipse alternate file systems
- added --- New Report Access Security
- added --- Support for batching XMLRPC requests
- fixed --- Eclipse plugin now correctly referenced in PRC server (case 58623)
- fixed --- Fixed a problem where going to a nonexistent review would create a NPE. (Case 57456)
- fixed --- Converted a help link which opened an XML file to raw text to compensate for browser incompatibility (Case 58547)
- fixed --- Improved error messages if group synchronization fails to parse correctly
- fixed --- Corrected a link on the Admin panel that was opening the wrong help section (Case 57780)
- fixed --- Corrected some online manual links that resulted in 404 errors (Case 57677)
- fixed --- Running 'addgitdiffs' on a bare repo will no longer create a NPE (Case 58240)
- fixed --- Corrected issue with Subclipse and SVN 'mergeinfo' support (Case 57555)
- fixed --- Updated change type in already existing versions for TFS (Case 57371)
- fixed --- Changed diff uploads to use streams
- fixed --- Updated fun facts to be more generic (Case 57972)
- fixed --- Several corrections to the owner's manual
- fixed --- Corrected missing space in 'ccollab addversions' documentation (Case 57964)
- fixed --- Removed request for 'mergeinfo unsupported' error with Subversive (Case 57555)
- fixed --- Corrected '%1 is not a valid Win32 application' error when installing Print To Review (Case 58006)

## 6.1.6104 — May 25, 2011

- added --- support Subversive 0.7.9.20110207-1700

- fixed --- More logging during server startup
- fixed --- Server triggers replacements failed for custom fields with whitespace (Case 57516)
- fixed --- All Defects report filtered by group did not show defects (Case 57033)
- fixed --- License expiration date is now always displayed in UTC with clear timezone indication (Case 57473)
- fixed --- TFS changeset deletes not showing as deleted (Case 51361, 57371)
- fixed --- Error processing filenames containing ampersands on windows (Case 57437)
- fixed --- Remove spurious extra logging in Eclipse plugin
- fixed --- Poor write performance when content-cache is located on a network share
- fixed --- Parsing of StartTeam diff filenames with embedded spaces (Case 57197)
- fixed --- Error reviewing subversion revision 1
- fixed --- Parse Surround history output with multiple continuation lines (Case 57288)

## 6.1.6103 — April 15, 2011

- added --- Installers for 64-bit JVMs on Windows
- fixed --- Print To Review not deleting image files on cancel (Case 54914)
- fixed --- "Download Diffs" fails for large reviews. (Case 56889)
- fixed --- "Download Diff" link now checks SCM permissions. (Case 57110)
- fixed --- "Open in external diff viewer" link now checks SCM permissions. (Case 57110)
- fixed --- Ticket verification should be case-insensitive (Case 57058, 57075)
- fixed --- Eclipse plugin Defect links truncated
- fixed --- Eclipse plugin should post chat asynchronously
- fixed --- Remove ccollab-cvs trigger from installer (Case 56876)
- fixed --- Support SourceGear Vault Professional Edition (Case 56205, 56737, 56863)
- fixed --- Fix Surround 'Archive error: -9014'
- fixed --- "Suppress Notification" checkbox does not stay checked. (Case 56929)

- fixed --- Chat pane does not load for binary files (Case 56749)
- fixed --- Support path-based access control for Apache-based Subversion servers (Case 56657)
- fixed --- Support blank option values on the command-line
- fixed --- attachments via web UI fail (Case 56783)
- fixed --- TFS commit issue (Case 56528)
- fixed --- TFS shelvesets author check should be case-insensitive (case 56436)
- fixed --- NPE when file has no base version (Case 56553)
- fixed --- NPE on alternate git status text (Case 56443)

## 6.1.6102 — March 9, 2011

- fixed --- Server fix for broken `ccollab admin wget` and Eclipse plugin w/ MySQL or Oracle. (Case 56315)
- fixed --- Add product name to password prompt dialog
- fixed --- Fix AppSetting not found error configuring Studio Addin for debug
- fixed --- Overwrite stale ticket in GUI client and Eclipse (Case 56426)
- fixed --- Error committing Subversion file with spaces (Case 56534)
- fixed --- Older (4.0) clients depleted database connection pool. (Case 56568)
- fixed --- Invalid regular expression in ExtJS (Case 55848)
- fixed --- RTC integration missing icon
- fixed --- NPE refreshing Review Summary Page (Case 56462)
- added --- Support MKS Source Configuration Path project types (Case 56327)

## 6.1.6101 — February 22, 2011

* FIRST RELEASE OF v6.1! *

Major features:

- Tasktop Certified Mylyn Integration
- Ticket-based logins
- Support Subversive Eclipse Subversion plugin

## 6.0.6018 — February 18, 2011

- fixed --- do not limit reviews to N changelists (Case 56011)
- fixed --- Eclipse Review Editor errors "Graphic is disposed" (Case 56393)
- fixed --- Attachments should be archived with other clients.

## 6.0.6017 — February 11, 2011

- fixed --- satisfy TaskTop requirement that the error icon used in the collaborators section of the review editor should be the one commonly used in eclipse
- fixed --- satisfy TaskTop requirement that required custom fields be decorated with a warning instead of an error icon.
- fixed --- LDAP auth with DOMAIN/username broken (Case 55481)
- fixed --- Browser hangs loading native document conversations (.doc file) (Case 56253)
- fixed --- typo in documentation for 'ccollab logout'
- fixed --- TFS deleted files showing as initial (Case 55981)
- fixed --- Case 56253
- fixed --- Explicitly support "M/d/yy h:mm:ss a" when parsing Surround changelists, instead of expecting this to be the default system date format (Case 56079)
- fixed --- AddFiles should not attempt to add a base version (Case 56157)
- fixed --- do not require Eclipse CVS plugin (Case 56175)
- fixed --- Supply Subversion username and password when adding arbitrary diffs
- fixed --- Allow users to further configure dynamic reports before running them. (Case 54888)
- fixed --- Dynamic reports will show more useful URL's in the URL bar if possible.

- fixed --- Apply and OK buttons disabled in the GUI client (Case 56120)
- fixed --- NullPointerException adding pending changelist with Eclipse plugin (Cases 55998, 56080)
- fixed --- remove extra prompt for password on ccollab login
- fixed --- Rename products in manual.
- fixed --- do not require access to Subversion repository root (Case 53783)
- fixed --- do not require access to Subversion repository root (Case 53783)
- added --- Support Vault Professional and 5.1.1 release (Case 56205)
- added --- diagnostic to add users to ALL_USER group if they are not already in it (Cases 56090, 56101). this is an empty commit because the original commit
- added --- documentation for $review.defectlog variable (Case 56196)

## 6.0.6016 — January 19, 2011

- added - Respect Subversive Plugin "Do not select externals" preference
- added - Subversion 'svn-recurse-externals' 799 option to recurse in to svn:externals directories
- added - Subversion svn-auto-add 799 global option to treat unversioned files as "added"
- added - Better command-line messages for ccollab admin group member add
- added - review_activity_summary 914 columns: active_reviewer_hours, passive_reviewer_hours (Case 55374)
- added - Print To Review 64-bit driver (Cases 55373, 54985, 52098)
- added - Extra logging around search
- added - Support for Team Foundation Server 2010 (Case 54907 and others)
- added - Radio buttons for Clearcase "Add versions" GUI file selection usability (Case 50471)
- added - Support Subversive Eclipse Subversion plugin
- added - Diagnostic to clean up duplicate values in metadatavalue* tables (Case 55633)
- added - Prompt from eclipse client and standalone GUI when bad auth credentials prevent succesful server responses.

- fixed - Handle TFS Russian date formats (Case 55797)
- fixed - Bad warning message when uploading unversioned file using 'ccollab addfiles'
- fixed - Suppress messages going to Subclipse console
- fixed - Error committing Subversion files in added directories
- fixed - Prevent duplicate Group members (Case 55623)
- fixed - Delete duplicate Group members in Fix Database Data diagnostic
- fixed - Eclipse resource decoration 544 and defect markers 544 fail to update
- fixed - Eclipse "Add to Review" menu item in Synchronize View missing for CVS and Subversion
- fixed - Exception when running "List Reports" in Oracle (Case 55721)
- fixed - GUI Client Git "Add changes" does not canonicalize filesystem path (Case 54598)
- fixed - NPE in "Add changes" when adding from subdirectory of Git repo (Case 55676)
- fixed - Handle missing Perforce 'haveRev' better (Case 55458)
- fixed - Reviewer can incorrectly finish until file activity (Case 55456)
- fixed - Reworked files showing 'Initial' status (Case 55557)
- fixed - NPE in installer during ROOT.xml generation (Case 55578)
- fixed - Better error message when LDAP authenticated user has not logged in to the web UI (Case 55538)
- fixed - RTC server plugin fails to create review when files paths cannot be constructed (Case 55547)
- fixed - Error saving Review custom fields in Eclipse plugin
- fixed - Performance - Only search "user" comments, not system comments
- fixed - Skip change set links that are missing a workspace hint in RTC Server Plugin
- fixed - Translate ClearCase LATEST to actual version
- fixed - Try harder to not run out of memory when running Perforce commands (Case 46879)
- fixed - Tray Notifier always thinks there are new Action Items
- fixed - Can not add pending Perforce changelist from Eclipse Plugin (Case 55064)

- fixed - Improve initial load speed of diff page (Case 55306)
- fixed - Command-line process execution fails with msysgit (Case 55322)
- fixed - Collapse paths in tree when selecting locally modified files in "Add versions" UI (Case 50471)
- fixed - Stop storing password supplied at command line prompt in config file
- fixed --- ClassCastException in "Fix Duplicate Users Differing By Case" Diagnostic. (Case 55910)

## 6.0.6013 — November 17, 2010

- added --- Support KB, MB and GB suffixes for smartbear.ccollab.upload.truncate.size (Case 54922)
- added --- Eclipse plugin support for reviewing URL's
- added --- Show "pins" in Eclipse Compare Editor for images and documents
- added --- Support for document review in Eclipse plugin
- added --- The assignment_state database view describes actioncodes in the assignment table. (Case 55049)
- added --- Find changes on ClearCase branch for Add Versions (Case 50471)
- added --- Edit participant custom fields in Eclipse plugin Review Editor
- added --- New "ccollab admin review edit --participant-custom-field" option
- added --- Clearcase: UI for add versions from branch (Case 50471)
- added --- Ability to "Poke" a Review participant in the Eclipse Plugin
- added --- Eclipse Plugin Review Editor show participant state using icons and font
- added --- Support Vault 5.0.4/Fortress 2.0.4 (Case 54773)
- added --- Double-click an entry in the Eclipse plugin Review Editor defect log to open the associated file
- added --- Show extended information in Eclipse plugin Review Editor defect log tooltip
- added --- "File" and "Location" columns in Eclipse Review Editor defect log section
- added --- Specify custom label for conversations in Eclipse Plugin

- fixed --- Throttle error messages about old/unsupported clients. (Case 55194)
- fixed --- Next/prev buttons in diff viewer failed after changing version comparison. (Case 55166, 54870)
- fixed --- NPE when converting null dates to strings (Case 55179)
- fixed --- Initial setup page changed user logins instead of adding new users. (Case 54771)
- fixed --- Support non-English locales with Subclipse integration (Case 55185)
- fixed --- Allow single-word user names (Case 54528)
- fixed --- Version change without content change shows 'Uploaded File' instead of change metrics (Case 55130)
- fixed --- Wrong version uploaded for ClearCase activities with unordered changes (Case 54648)
- fixed --- Change server.xml defaults to improve server behavior under load
- fixed --- Upgrade Tomcat to 6.0.29
- fixed --- NPE updating defect markers in Eclipse plugin (Case 55064)
- fixed --- User selections should include user logins for disambiguation
- fixed --- Links in dynamic reports were broken on first page load. (Case 54934)
- fixed --- Respect custom field phase visibility in Eclipse Plugin
- fixed --- Review overview fails to load - NPE with empty label location on Oracle backend (Case 54839)
- fixed --- Client installer removes other P4V custom tools (Case 54448)
- fixed --- Improve 'Invalid ClearCase version' message (Case 54975)
- fixed --- Visual Studio Addin unloads when solution is closed (Case 52634)
- fixed --- Visual Studio Addin should display error message if nothing to do (Case 54661)
- fixed --- Eclipse plugin mangles multi-line text custom fields (Case 54867)
- fixed --- Wrong base version for Vault rolled back versions (Case 53902)
- fixed --- Eclipse Plugin Review Editor NPE with CodeReviewer (Case 54790)
- fixed --- Prevent MKS host name truncation (Case 53470, 54780)
- fixed --- Prepopulate Print To Review document name (Case 54123)

- fixed --- Defect Custom field order not respected (Case 54717)
- fixed --- P4V integration does not prepopulate review title (Case 54449)
- fixed --- New comment text box in Eclipse plugin does not expand correctly as you type
- fixed --- Error uploading reverted CVS file in Eclipse plugin (Case 54587)
- fixed --- Be more lenient for administrator full names in support of non-Latin-1 character sets (Case 51433)
- fixed --- Error using relative paths with 'ccollabgui addchanges' (Case 54459)

## 6.0.6012 — October 1, 2010

- added --- Documentation for content-cache format prop file
- added --- Compare against any local file in Eclipse plugin
- added --- Select line in Eclipse plugin Compare Editor when conversation focused in Conversations View
- added --- Compare against any local file in Eclipse plugin
- added --- "Refresh" button to Eclipse Review Editor
- added --- Display links for URL's and emails in Eclipse Plugin Conversations View
- added --- Links in Eclipse Plugin Conversations View
- added --- Links in Eclipse Plugin custom fields and defect text
- added --- Linkify-as-you-type in Eclipse Plugin
- added --- Display Review ID in "Add To Review" wizard confirmation page (Case 54371)
- added --- Log IP address of old clients (Case 54392)
- added --- New reporting views. (Case 54438)
- added --- Option to suppress notifications by type/template
- added --- Interim 'ccollab logout' command that clears password (Case 54065)
- fixed --- HTML markup displayed in Eclipse plugin 'Moving On' section
- fixed --- Multi-selected files not uploading from addin
- fixed --- Defect links and comment box sometimes disabled improperly in Eclipse plugin

- fixed --- do not mark review editor dirty just because custom field values are not yet assigned
- fixed --- NullPointerException committing from tray notifier
- fixed --- Eclipse Review Editor "Moving On" section missing controls after pressing "Begin Review"
- fixed --- Make Conversation ruler background color match file in Eclipse Compare Editor
- fixed --- Missing checked graphic in diff viewer
- fixed --- Eclipse plugin Compare Editor "Too Many Differences" error"
- fixed --- Erroneous Concurrency Exception on upgrade (Case 54087)
- fixed --- Can no longer create multiple custom fields that differ only by case (Case 54087)
- fixed --- Corrected features-not-supported-in-CodeReviewer list
- fixed --- Email notifier log message says "seconds" instead of "milliseconds"
- fixed --- Display image for file type in Eclipse Compare Editor
- fixed --- Prefs page fails to render tabs in IE7
- fixed --- In Notification Template admin screen, sort the templates by display name.
- fixed --- Make 'browser' global option case-insensitive (Case 50407)
- fixed --- Keep Eclipse Plugin Action Items View and Editors in sync
- fixed --- Use custom label for "Defect" in Eclipse Plugin
- fixed --- Eclipse plugin "Could not get defect attribute from defect marker" error (Case 54349)
- fixed --- Sort next file/previous file the same as the review overview screen, collating case (case 54132)
- fixed --- p4 \r\r\n line endings confuse ensure-content-reviewed trigger (Case 54213)
- fixed --- Better error messages for Eclipse Plugin (Case 54350)
- fixed --- do not overwrite file content when restoring from backup (Case 54401)
- fixed --- Installer should automatically replace old P4V custom tool definitions (Case 54450)
- fixed --- Updated manual to document all views. Corrected view name to: defects_by_path (Case 54437)
- fixed --- Dynamic Defect Report filter by Review Completion Date (Case 54226)

- fixed --- Error upgrading server if no diffs cached (Case 54381)

## 6.0.6011 — September 8, 2010

* FIRST GA RELEASE OF v6.0!

Note: 6.0 Beta clients (6.0.6001 - 6.0.6009) are not compatible with the GA 6.0.6011 server (5.0 clients *are* compatible with 6.0.6011 server). If you were using a 6.0 beta client please upgrade your client to the GA version 6.0.6011.

- added --- do not allow saving password via ccollab set (use ccollab login instead)
- fixed --- Group Admins 231 may delete groups they admin (Case 52208)
- fixed --- Last Comment shows if redacted 434 (Case 53732)
- fixed --- Error entering new Comments on "local file 544" version in Eclipse plugin (Case 53820)
- fixed --- Create distinct changelists for unmanaged files uploaded via ccollab addfiles (Case 52713)
- fixed --- Append indicator page to end of truncated Print To Review documents
- fixed --- do not allow new conversations when comparing against changed local file 544 in Eclipse plugin
- fixed - Eclipse plugin Defect Markers 544 and Label Decorations 544 do not clear when Review deleted
- fixed - Print to Review displays "No Title" error (Case 53929)
- fixed --- NPE on Review Detail Report (Case 53682)
- fixed --- User Reports specify "System Admin" instead of ambiguous "Admin" (Case 53084)
- fixed --- Removed unused 'issue' table (Case 52652)
- fixed --- ccollab addchanges --diffbranch option should not return CHECKEDOUT versions (Case 53680)

## 6.0.6009 — August 8, 2010

- added --- New file content store format for better FS performance.
- added --- Print To Review and Studio Addin launch 'Add To Review' GUI wizard
- added --- Enable log level management through JMX
- added --- External diff launcher preset for BeyondCompare on Linux
- added --- Supported browsers now include Chrome
- added --- Display graphical UI for P4V integration
- added --- P4V integration upload multiple changelists at once
- added --- Show user login to disambiguate user names on Group admin page (Case 53589)
- added --- Click on a line in the Eclipse compare viewer and start typing to add comment
- fixed --- Added troubleshooting info to the manual for p4v/p4win plugin (Case 53579)
- fixed --- Added keyboard shortcuts for GUI client to manual (Case 53578)
- fixed --- Eclipse client new chat box does not size correctly with trailing line feed on Windows (Case 52804)
- fixed --- Review completion from the command line ignores unset required participant custom fields. (Case 51714)
- fixed --- Role Configurations cannot be created (Case 53418)
- fixed --- Double-click entry in Eclipse diff's structure view and the window shifts to the right (Case 51590)
- fixed --- SCM Configurations show bogus local path
- fixed --- Install P4 Plugins checkbox should be checked by default if Perforce installed.
- fixed --- Support redacted comment update in eclipse plug-in
- fixed --- Documentation on how to change the file cache location (Case 47974)
- fixed --- Restrict Uploads to Review cannot be changed on review edit (Case 52417)
- fixed --- Show redacted comments as redacted on review overview
- fixed --- Add documentation for configurable server log settings (Case 50107)
- fixed --- Comments sort in wrong order in document review. (Case 53540)
- fixed --- "Fix Defects" phase now uses custom label for Defects.

- fixed --- Eclipse client allows you to "wait" when you are already waiting (Case 51559)
- fixed --- Group member edit list of members is too narrow in IE (Case 50184)
- fixed --- ccollab admin review wait is --until argument is now optional; defaults to Any
- fixed --- Include disabled and reporting-only Groups in Review report filter
- fixed --- Organize Groups on Admin->Groups page (Case 53586)
- fixed --- Change the accepted version icon in the pulldown to differ by more than just color
- fixed --- Display "not yet supported" message for "Live URL" files in Eclipse Plugin
- fixed --- "Concurrency Exception" after system upgrade. (Case 53621)
- fixed --- Select current location when clicked in right side of Eclipse compare viewer
- fixed --- Update top-level Git page in manual (case 53395)
- fixed --- Focus correct conversation control when new conversation started in Eclipse client

## 6.0.6008 — August 6, 2010

- fixed --- Eclipse update site URL in 6.0 zipped update site pointing to 5.0
- fixed --- Addin under VS 2010 (cases 51282, 52100, and 53275)
- fixed --- Eclipse plugin "Recent Participants" have the wrong Role
- fixed --- Make clearcase addversions wizard upload multiple versions at a time
- fixed --- Restrict Access to Review has inconsistent wording (Case 52444)
- fixed --- Removed broken "Revert" buttons from Group Admin page
- fixed --- PHP not syntax highlighted when opening <?php is not closed (Case 50039)
- fixed --- In document review, clicking on a different page's comment does not select that comment (Case 52470)
- fixed --- Display an error on review edit when not all required assignments are present (Case 52008)
- fixed --- Diff viewer keyboard shortcuts help does not work in Chrome (Case 51820)
- fixed --- Eclipse client custom field error decoration clipped in Review Editor

- fixed --- Eclipse Review Editor appears dirty if string custom field is empty
- fixed --- Group selection does not appear in review editor if no group is assigned (case 52872)
- fixed --- ClearQuest activity review titles should default to headline (case 49180)
- fixed --- Use first commit hash as Git repo identifier
- fixed --- Allow non-ascii characters in username/password field (Cases 51434 and 52406)
- fixed --- Sort files alphabetically in GUI client "Add Changes" wizard (Case 53067)
- fixed --- Typos in documentation of "ccollab admin batch" command (Case 52711)
- fixed --- Redacted comments should not be counted or displayed in reports (Case 52256)
- fixed --- Alphabetize users and groups on Group edit page (Case 53067)
- fixed --- Redacting a comment on zoomed image resizes image (Case 52277)
- fixed --- Spurious P4V addons message at end of install process (Case 48043)
- fixed --- Move restrict process options to access restrictions section (Case 49848)
- fixed --- Error uploading Subversion revisions with directory changes in Eclipse Plugin
- fixed --- Performance - Case insensitive string indexes in Oracle (Case 52392)
- fixed --- Content Archive queries do not scale well
- fixed --- Enforce administrator dump restriction at dump time, not just display time
- fixed --- Surround changelist date AM/PM concerns (case 52522)
- fixed --- Default Value for Participant Custom Field broken (Case 52082)
- fixed --- Changed description of "Allowed to Modify Review" to better reflect true behavior (Case 51954)
- fixed --- When Eclipse plugin refresh job fails with network error, stop all refresh jobs
- fixed --- Order files alphebetically in ClearCase version spec page
- fixed --- Prevent multiple instances of server (multiple ROOT.xml files) from running at the same time
- added --- Button to Eclipse Review Editor to open Review in browser
- added --- Implement defect "Track Externally" button in Eclipse plugin

- added --- User Detail Report now contains a link to get a CSV file containing the User Activity data
- added --- Compare reviewed version with local file in Eclipse client
- added --- 'ccollab admin review defect mark-not-external'
- added --- Option to ignore integration history when caculating previous versions in perforce (Case 51845, 45704)
- added --- Always show line numbers by default in Eclipse plugin line-based Compare Editor
- added --- Print To Review support for 300 DPI color (case 52050)
- added --- Check for concurrent DB access
- added --- Change autofill UI in clearcase addversions wizard
- added --- Instructions in manual on how to create database and user in mysql using command line
- added --- Better error handling for server connection errors in Eclipse plugin
- added --- Submit comments / create defects with in Eclipse plugin

## 6.0.6006 — July 9, 2010

- added --- Command-line commands address existing Conversations by first Comment ID
- added --- Command-line commands 'ccollab admin review conversation mark-read' and 'ccollab admin review conversation accept
- fixed --- make template editable from eclipse review editor
- fixed --- Defect markers on wrong lines in Eclipse Client
- fixed --- parse diffs files have a Unicode byte order mark (Case 52660)
- fixed --- Enforce administrator dump restriction at dump time, not just display time.
- fixed --- New branched TFS files missing from shelveset review (case 48530)
- fixed --- Better detection of content type in Eclipse compare editor
- fixed --- Better file type icons in Eclipse client Review Editor
- fixed --- Fewer database queries to load diffs on cache miss (Case 52392)

## 6.0.6005 — June 29, 2010

- fixed --- Typing in Eclipse Plug-in chat and defect text boxes is slow
- fixed --- Eclipse Plug-in Review Editor 538 icon inconsistent
- fixed --- File URL's break Review overview (Case 51666)
- fixed --- GUI Client missing larger logo sizes
- fixed --- Diff Viewer Prev/Next buttons do not work (Case 52157)
- fixed --- If redacted comment 434 is most recent comment you cannot make the same comment (Case 52224)
- fixed --- Hidden markers 416 shown again when zoom 416 changes
- fixed --- Missing page notification fails to clear after zoom 416
- fixed --- Chat column 429 too narrow in IE8 (Case 52280)
- fixed --- Strange text in confirmation message box when deleting Defect (Case 51460)
- fixed --- User detail report missing one Group (Case 52110)
- fixed --- Javascript error loading Groups admin screen
- fixed --- Participant names are listed in phases where no custom fields are visible (Case 52042)
- fixed --- Eclipse Plug-in Review Editor 538 shows extraneous "treenode" icon on files
- fixed --- Eclipse Plug-in initialize custom fields 256 with default values
- fixed --- Wrong default value for selection custom field
- fixed --- Print to Review installer should update system path variable (Case 52390)
- fixed --- Eclipse Plug-in remove "edit" state for Defects
- fixed --- Unable to parse remote origin url from config file (Case 52589)
- added --- Documentation for Git integration 648 in the GUI Client
- added --- Documentation for redact comment 434 feature
- added --- Command-Line Client command "ccollab admin review set-participants"

- added --- Documentation for Mercurial 730 integration
- added --- Drag and drop participants 538 to change roles in the Eclipse Plug-in Review Editor 538
- added --- Right-click to edit participants 538 in the Eclipse Plug-in Review Editor 538
- added --- Display participant custom fields 258 in Review detail report
- added --- Eclipse Plug-in UI to add participants
- added --- Eclipse Plug-in Review Editor 538 can drag from users list to participant list to add participants
- added --- Drop-down menus to select which version to diff in Eclipse Plug-in Compare Editor 543
- added --- Print to Review documentation with troubleshooting info on Windows 7 and Vista (Case 52398)
- added --- UI field assist for custom field 256 errors in Eclipse Plug-in
- added --- Database views 914 for participant custom fields 258 (Case 52324)
- added - Eclipse Plug-in Conversations view 541 can be filtered to show only defects or comments, removed "Defects" and "Comments" views
- added --- Eclipse Plug-in allow multiple instances of Conversations view 541
- added --- File status (added/modified/deleted) icon decorations in Eclipse Plug-in Review Editor 540
- added --- Show Eclipse icon for file type in Eclipse Plug-in Review Editor 540

Fixes from 5.0 build 5033 1130:

- fixed --- P4 integration 764 ignores unmodified 'move/add' files when configured to 'revertUnchanged' (Case 52319)
- fixed --- Command-Line Client "ccollab addchanges" link in Subversion command line documentation (Case 52413)
- added --- Support adding MKS change packages from other authors (Case 52317)
- added --- Performance improvements for custom field queries
- added --- Option to specify how frequently server checks for stalled reviews (performance) (Case 52392)

Fixes from 5.0 build 5032[1131]:

- fixed --- TFS[738] Date parsing error in New Zealand (Case 51476)
- fixed --- Improve performance - remove some wasteful queries

Fixes from 5.0 build 5031[1131]:

- fixed --- Not finding CMVC track and find binaries on Solaris (Case 51930)
- fixed --- Error adding TFS[738] Shelveset to Review (Case 51386)
- fixed --- Performance issues with SQL Server[64] activity queries (Case 52086)
- fixed --- Not all defect activity pulls defect activity trigger[207] (Case 51898)
- fixed --- Dynamic reports now properly display MSSQL[64]/HSQL[60] dates (Case 49695)
- fixed --- do not include Group User members[232] in configuration-only dump
- added --- More debug logging for "add diffs[510]" commands

### 6.0.6001 (beta) — June 2, 2010

* FIRST RELEASE OF v6.0 BETA! *

Major new features include:

- Support for reviewing Microsoft Office documents
- An Add-In for Visual Studio - add files to a review from the Solution Explorer
- Significant enhancements to the integrations with ClearCase[666], Rational Team Concert, Git[645], and Mercurial[730]
- Ability to redact comments[434]

## 11.2.9 Version 5

### 5.0.5041 — February 18, 2011

- fixed --- Update Vault 5.0 references to 5.1 (Case 56205)
- fixed --- TFS 738 deletes showing up as initial (Case 55981)
- fixed --- NullPointerException adding pending changelist 551 with Eclipse plugin (Cases 55998, 56080)
- fixed --- Backport TFS 738 Russian date fix to 5.0 (case 55797)

## 5.0.5040 — January 11, 2011

- fixed --- Update Vault 5.0 references to 5.1 (Case 56205)
- fixed --- TFS 738 deletes showing up as initial (Case 55981)
- fixed --- NullPointerException adding pending changelist 551 with Eclipse plugin (Cases 55998, 56080)
- fixed --- Backport TFS 738 Russian date fix to 5.0 (case 55797)
- fixed --- Prevent duplicate Group members (Case 55623)
- fixed --- Reviewer can incorrectly finish until file activity (Case 55456)
- fixed --- Translate ClearCase LATEST to actual version
- fixed --- Support deleted TFS files
- fixed --- Error uploading reverted CVS file in Eclipse plugin (Case 54587)
- fixed --- Version change without content change shows 'Uploaded File' instead of change metrics (case 55130)
- fixed --- Wrong version uploaded for ClearCase activities with unordered changes (case 54648)
- fixed --- Improve 'Invalid ClearCase version' message (case 54975)
- fixed --- Wrong base version for Vault rolled back versions (case 53902)
- fixed --- Support Vault 5.0.4/Fortress 2.0.4 (Case 54773)
- fixed --- Prevent MKS host name truncation (cases 53470,54780)
- fixed --- Updated manual to document all views. Corrected view name to: defects_by_path (Case 54437)
- fixed --- "sort next file/previous file the same as the review overview screen, collating case" (Case 54132)

- fixed --- Prefs page fails to render tabs in IE7

## 5.0.5039 — September 17, 2010

- fixed --- defect triggers do not fill in substitution for ${actor.*} (case 53618)
- fixed --- '--diffbranch' option should not return CHECKEDOUT versions (Case 53680)

## 5.0.5037 — August 25, 2010

- added --- Configurable AccuRev history algorithm
- added --- VHDL syntax highlight (Case 53648)
- fixed --- Fix type coercion error in oracle compatibility (Case 52573)
- fixed --- Repair corrupted MetadataValueInteger fieldIds (case 52573)
- fixed --- Do not load inactive custom fields (Case 52573)
- fixed --- Chat Pane in FireFox 3.x too wide in Linux (Case 53580)
- fixed --- Update jPDFImages - PDF rendering never completes - 100% CPU load (Case 53486)
- fixed --- "Download Diff" generates invalid diffs. (Case 53479)

## 5.0.5036 — August 10, 2010

- fixed --- removed broken "Revert" buttons from the Group Admin page
- added --- Allow user to provide Subversion a HTTPS client certificate password at runtime (Case 53263)
- fixed --- Validate that a changelist can be detached (Case 53375)
- added --- Diagnostic to find and fix reviews that have illegally detached changelists (Case 53375)

## 5.0.5035 — July 29, 2010

- fixed --- Enforce administrator dump restriction at dump time, not just display time
- fixed --- Error uploading Subversion revision with directory changes in Eclipse Plugin
- fixed --- Performance - case insensitive string indexes in Oracle (Case 52392)

- fixed --- Content archive queries did not scale well
- fixed --- Auto-detect charecter encoding of user-provided diffs instead of using system default
- fixed --- do not flood output.log with regular log messages
- fixed --- Typos in documentation of "ccollab admin batch" command (Case 52711)
- fixed --- Allow non-ascii characters in username/password field (Case 51434, 52406)
- fixed --- ClearQuest activity review titles should default to headline (Case 49180)
- added --- Option to ignore integration history when calculating previous versions in Perforce (Case 51845, 45704)

## 5.0.5034 — July 9, 2010

- fixed --- Parse diff files that start with Unicode Byte Order Mark(Case 52660)
- fixed --- Reduce number of queries required to load diffs (Case 52392)
- fixed --- New branched TFS files missing from shelveset review (Case 48530)
- added --- Server plug-in[717] for Rational Team Concert, Beta

## 5.0.5033 — March 23, 2010

Upgrade to ExtJS 3.1. This fixes many UI issues including:

- fixed --- Diff Viewer resize issues in Internet Explorer (Case 52189)
- fixed --- Collaborator defect form entry display issue (Case 52096)
- fixed --- Participants list boxes when creating/editing review are cut off in ie6 (Case 52032)
- fixed --- Missing scrollbars on initial diff view load (Case 38247)
- fixed --- Image diffs in IE 6 do not work (Case 46236)

- added --- Support adding MKS change packages from other authors (case 52317)
- added --- Option to specify how frequently server checks for stalled reviews (performance) (Case 52392)

- fixed --- P4 integration ignores unmodified 'move/add' files when configured to arevertUnchanged' (Case 52319)
- fixed --- Updated backup/migration docco to cover how to point collab to a new database (Case 52411)
- fixed --- Performance - Improve custom field queries
- fixed --- Must provide value for sandbox (case 49641)
- fixed --- Fix 'Value requires 2 integer parts' exception (case 51225)
- fixed --- Support uploads of change packages from other authors (case 52317)

## 5.0.5032 — June 14, 2010

- fixed --- do not include group user members in configuration-only dump
- fixed --- Performance - Remove some wasteful queries.
- fixed --- ClassCastException in Oracle data loading code.
- fixed --- TFS Date parsing error in New Zealand (Case 51476)

### 5.0.5031 — 2010/06/10

- added --- More debug logging for "add diffs" commands (Case 52124)
- fixed --- Wrap long error messages (Case 51915)
- fixed --- Not finding CMVC track and find binaries on Solaris (Case 51930)
- fixed --- Error adding TFS 738 shelveset to review (Case 51386)
- fixed --- Performance fixes for SQL Server activity queries (Case 52086)
- fixed --- Not all defect activity runs defect activity trigger (Case 51898)
- fixed --- Dynamic reports now properly display MSSQL/HSQL dates (Case 49695)

## 5.0.5030 — May 22, 2010

- added --- Support for 'addactivity all' (Case 50444)
- added --- Better documentation for addversions command.
- added --- option to specify local database path in Rational Synergy (Case 51587)

- added --- Forward compatibility; improve error messaging when connecting to a 6.0 server.
- fixed --- Addactivity hangs reading activity (Case 51276)
- fixed --- ClearCase performance improvements (Case 50942)
- fixed --- Incorrect timezone offset for submitted Perforce changes (Case 50679)
- fixed --- Create new User Account subtext (Case 51328)
- fixed --- Handle deleted files in ClearCase activities (Case 51275)
- fixed --- Reports CSV output should not consume a license (Case 51457)
- fixed --- Auto-refresh clears unsubmitted comments in the text area (Case 51369)
- fixed --- Reviews created with addversions include files from other reviews (Case 50682)
- fixed --- Notification emails not being sent. (Case 51204)
- fixed --- Command line client missing edit file list prompt message.
- fixed --- NPE when content unavailable; blank screen loading user home page.

## 5.0.5029 — April 20, 2010

- fixed --- Add per-file lines metrics to review-xml (Case 51010)
- fixed --- Some PDFs do not show up in reviews (Case 50826)
- fixed --- Performance improvement - fewer queries to load action items.
- fixed --- Performance improvement - fewer queries to perform comment promotion; render review overview
- fixed --- Performance improvement - reduce computation when rendering review overview
- fixed --- Performance improvement - reduce computation required to map from SQL result sets to datamodel objects.

## 5.0.5028 — April 13, 2010

- added --- Technology Preview of IBM Rational Team Concert support
- fixed --- Limit syntax highlighted file size to prevent CPU spike when processing large files (Case 51015)
- fixed --- 'not a ClearCase object' error on Linux (case 50938)

- fixed --- Synergy passwords in clear text of debugging logs (Case 50925)
- fixed --- Fix NPE in logging (case 50799)

## v5.0.5027 — March 26, 2010

- added --- 'ccollab admin review defect delete'
- added --- 'ccollab admin review defect mark-fixed' and 'ccollab admin review defect mark-opened' commands
- added --- "ccollab admin review edit" "--deadline" option to change Review deadline (Case 50739)
- fixed --- Bad data on 'ccollab admin review-xml' in some cases using Sun JRE 6. (Case 50372)
- fixed --- Setting multi-valued fields from 'ccollab admin review defect edit'
- fixed --- Cannot Accept or comment after a file has been deleted (Case 50466)
- fixed --- Manual did not reflect changes to review deadline implementation (Case 47876)
- fixed --- Bad error message when command-line scm autodetect fails
- fixed --- complex histories cause poor performance adding files with Synergy (Case 50594)
- fixed --- error creating new custom field with same name as deleted custom field (Case 50457)
- fixed --- Improved performance of 'ccollab addchangelist' with multiple changelists
- fixed --- strip xml1.0 unfriendly chars from arguments list of xmlrpc calls
- fixed --- addactivity diffintegration option diffs wrong integration stream version (case 49870)
- fixed --- Set default review deadline when creating reviews from command-line (Case 50739)
- fixed --- better debug logging for "ccollab admin wget"
- fixed --- 'ccollab admin wget' fails with servers not at root of host (Case 50101)
- fixed --- NPE in logging when log message is null (Case 50799)

## v5.0.5026 — March 3, 2010

- added --- Command-line support for creating Defects and Comments at page coordinates
- added --- Documentation for configurable cache settings. (Case 50104)

- added --- Support Perforce Eclipse plugin[560] v2009.2.234487
- added --- Allow setting of a review's 'display changelists as' setting from the commandline (Case 50327)
- fixed --- Toolbar titles misaligned (Cases 48690, 48846, 49976)
- fixed --- Preserve state of Accept and Mark Read buttons after canceled defect edit. (Case 49512)
- fixed --- HTML encoding interacting with auto linking. (Cases 49794, 50010)
- fixed --- StarTeam add-diffs not capturing path info (Case 49875)
- fixed --- Cannot cancel, externalize, or edit Defect on Review overview (Case 47734)
- fixed --- NullPointerException in DocumentPageServlet when engine is not available (Case 50134)
- fixed --- Updated manual with current MySQL installation instructions[60] using the supported tools (Case 50002)
- fixed --- Updated docs with information on integrated authentication with SQL Server[64] (Case 49853)
- fixed --- NPE when version server path empty (Case 49944)
- fixed --- Race condition between content archiver and review creation (Case 30131)
- fixed --- Invalid cached differences result in no diffs being displayed (Case 30131)
- fixed --- Paths in reports are double encoded
- fixed --- Poor error behavior when "p4 set" and "p4 info" have no output (Case 50331)
- fixed --- Wrong ClearCase[666] branch predecessor for zero versions (Case 49102)
- fixed --- MKS[752] change packages should always be pending (Case 48754)
- fixed --- Updated manual to clarify overdue deadline status messages

## v5.0.5025 — February 10, 2010

- fixed --- Background services not started if database was unavailable at startup
- fixed --- Bad error message when no assigned tasks are found (Case 49423)
- fixed --- Can not add yourself as a participant to an existing review (Case 49419)

- fixed --- Default review title not set for addactivity[688] (Case 49180)
- added --- Getting started links on the last page of the client installer[487]
- added --- 'ccollab admin wget' command
- fixed --- Real paths for Rational Synergy[691]
- fixed -- The "Link" link in dynamic reports now works again. (Case 49199)
- fixed --- MKS[752] files incorrectly marked as new (case 48775)
- fixed --- Canonicalize MKS[752] host names (case 49466)
- added --- Keyboard shortcuts in the GUI client[496]
- fixed --- Verify connection to server when setting up Subversion[804] (Case 49773)
- fixed --- Updated installation documentation[64] for SQL Server 2008, Java 6, and JDBC drivers (Case 49238)
- fixed --- Users duplicated in drop-downs when using Groups (Case 49594)
- added --- Update browser launcher to support GNOME open, KDE open, and XDG open (Case 49677)
- fixed --- Participant filter name in dynamic reports is once again "participantLogin". (Case 49584)
- added --- 'ccollab commit --dismiss-only' option

Fixes from 4.0 build 864[1153]:

- fixed --- List report results improperly cached (Case 49572)
- fixed --- LDAP lockout can occur if Collaborator is used infrequently. (Case 49526)
- fixed --- "Too many open files" error in Content-Cache Diagnostic (Case 49687)

## v5.0.5024 — January 14, 2010

- added --- External Diff Launcher variable "after.version.localFilePath" to enable launching editor (http://uservoice.com/a/7wS9r)
- added --- JMX Interface to Diagnostics
- added --- addversions support for Rational Synergy (Case 48922)
- added --- edit defects on the command line (Case 49142)

- fixed --- filtering a report based on users now works properly with multiple participants in each role
- fixed --- NPE getting content from deleted file using ccollab adddiffs with before and after directories
- fixed --- Visual Studio Addin does not size correctly (Case 48914)
- fixed --- Visual Studio Addin should show wait cursor while busy (case 48914)
- fixed --- option to start Synergy session with -rc (Case 48922)
- fixed --- Perforce local changelists not showing reverted files as reverted (Case 48321)
- fixed --- unrepresentable characters break RPC interface (Case 46761)
- fixed --- Sort action items consistently
- fixed --- VSS diff anomalies (case 47484)
- fixed --- Review too large for refresh algorithm is wrong
- fixed --- use Eclipse proxy settings UI, if available (Case 41725)
- fixed --- broken error message on custom fun facts.
- fixed --- Copy-paste not working in web diff viewer, selection is always cleared(Case 49193)
- fixed --- NPE when completing add changes wizard on review selection screen
- fixed --- email notifications stop being delivered (Case 45913)
- fixed --- NPE in Starteam diff support (Case 49263)
- fixed --- Clearcase '--diffbranch pre' option picks wrong predecessor (Cases 49102, 48802)

## v5.0.5023 — December 11, 2009

- added --- Better debug logging for add*diffs commands (Case 48205)
- added --- "ccollab addfiles" option --relative-to (Case 46457)
- added --- StarTeam '--stcmd-exe' option (Case 48622)
- added --- Diagnostics section to the manual (Case 48571)
- added --- High Availability Best Practices section to manual
- added --- Content store cleanup diagnostic

- fixed --- Display external diff launcher errors on MacOs
- fixed --- Custom fields values reset when non-participant adds self to a review (Case 48338)
- fixed --- Support p4 2009.1 move/add and move/delete actions (Case 48312)
- fixed --- Default external diff viewer does not launch on Mac OS (Case 48356)
- fixed --- Review dump not available from searching by review ID
- fixed --- Do not install AJP connector by default. (Case 48372)
- fixed --- Automatic session creation for Rational Synergy (Case 48065)
- fixed --- 'specify p4 command Charset' needs to be a boolean drop down in the web UI (Case 48071)
- fixed --- .smartbear directory too accessible (Case 48001)
- fixed --- Handle Synergy objects not associated with tasks (Case 48623)
- fixed --- Subsequent MKS sessions fail to connect (case 48642)
- fixed --- Action Items refresh should cache login for 20 minutes
- fixed --- Require non-blank label for "Group" (Case 47463)
- fixed --- Require non-blank external URL (Case 47351)
- fixed --- Require non-blank Group title (Case 47487)
- fixed --- Require non-blank Group title (Case 47487)
- fixed --- Ignore unmerged branches in history (Case 48623)
- fixed --- Error in creating Automatic Link removes creation data (Case 47616)
- fixed --- Notifications need not refer to existing reviews for Database Integrity Check to pass (Case 48688, 48727)
- fixed --- Determine perforce client is unknown when 'p4 info' output gives no client root (case 48525)
- fixed --- NullPointerException in shutdown if datamodel could not be initialized.
- fixed --- Security fix for XSS vulnerability (Case 48559)
- fixed --- Install4j not installing non-GUI symlinks(Case 48731)
- fixed --- Added a workaround for SWT/GTK bug (Case 48377)

- fixed --- Unparseable surround date format (case 48762)
- fixed --- Empty files being written to content cache (Case 48834)
- fixed --- Respond better to coming online before the database after reboots

## v5.0.5022 — November 12, 2009

- added — Accessibility improvements for diff view and review overview.
- added — Technology Preview of Rational Synergy support
- fixed — Error message on client when error uploading content to server (Case 48135)
- fixed — Log error message in server when content-cache is full (Case 48135)
- fixed — Older Perforce servers do not include "SubmitOptions" in client spec (Case 48100)
- fixed — Fix for MKS keyword expansion (Case 47784, 47251)
- fixed — Workaround for 'tf history' bug in Visual Studio 9.0 (Case 47438)
- fixed — Spaces in custom field names broke variable substitution (Case #48105)
- fixed — Log exceptions closing files in content store
- fixed — Make preferences directory as necessary to save "last" review and defect (Case 48194)
- fixed — Old p4 clients ignore -ztag with the where command (Case 48154)
- fixed — Auto-refresh disabled popup is to obtrusive. (Case 48177)
- fixed — Continue if missing MKS revision encountered in change package
- fixed — Rename "Subversion Server GUID" to "Subversion Repository UUID" in UI and Docs

## v5.0.5021 — October 28, 2009

- added — log author prep time while in planning phase (Case 47593)
- fixed — Check for compatible product version before trying to restore from a dump file (Case 47869)
- fixed — Upgrade to JavaMail for defect causing email notifications to stop (Case 45913)
- fixed — Report filters display incomprehensible error messages (Case 47768)

- fixed — server should install license file with the EULA (Case 47946)
- fixed — Handle Perforce SubmitOptions: revertunchanged (Case 47488)
- fixed — Remove "Restore Defaults" button from Server Connection preferences page (Case 47950)
- fixed — Error adding a Subversion revision with Eclipse plugin
- fixed — Typo in the mandatory subscriptions error message (Case 47469)
- fixed — do not specify "-Q utf8" when the customer specifies "p4commandcharset none" (Case 47032)
- fixed — Unrecognized TFS date format for zh_TW locale (case 47813)
- fixed — Spurious P4V addons message at end of the install process on when Perforce is not installed
- fixed — Updated jdpdfimages binaries to fix null pointer when uploading PDF and subsequent PDF rendering problem (case 47879)

## v5.0.5020 — October 15, 2009

- added — log author prep time while in planning phase (Case 47593)
- added — Added information about how groups are stored in the database for custom reporting
- added — Support for MKS renames (Case 47037)
- fixed — next/prev file in diff viewer algorithm wrong (Case 47406)
- fixed — review custom field visibility by phase not editable (Case 47283)
- fixed — Diff download format broken when change occurs at end of file.
- fixed — Diff download fails when insertions at end of file. (Case 47748)
- fixed — improve explanation of managing Groups manually vs with sync
- fixed — LDAP access broken when server returns absolute names (Case 47391)
- fixed — Verify Database Schema diagnostic shows bogus errors on Oracle (Case 47732)
- fixed — LicenseDecodingException experienced when system attempts to retrieve stored license (Case 47690)
- fixed — do not let user install server on Linux if port is in use

- fixed — Auto-refresh of review overview screen locks the browser UI thread (Case 47740)
- fixed — External Diff Launcher fails for certain files on Windows (Case 47649)
- fixed — Edit defect causes tab to be inaccessible to screen reader
- fixed — Error deleting symlink in perforce (Case 47327)
- fixed — Make Add SCM Configuration resize more nicely (Case 47290)

## v5.0.5019 — October 1, 2009

- added — DateTime control in Subversion update changes by date GUI dialog
- added — Database schema diagnostic
- added — Support commit info on changelist current versions
- fixed — Layout of error popup on Eclipse 3.5 and GUI client
- fixed — GUI Client and Tray Notifier will not launch on 64-bit MacOS (Cases 47336, 46337, 47149)
- fixed — Poor behavior in notifications with anonymous defects (Case 47405)
- fixed — Duplicated output on adddiffs (Case 47409)
- fixed — Missing arg in versionCreate: support commit info for current versions in changelists (cases 44540, 45620)
- fixed — Performance fixes. Reduce total database load when displaying reviews
- fixed — sitting on the review overview page can result in consuming a license (case 46356)

## v5.0.5018 — September 17, 2009

- added — Tray Notifier on MacOS (Case 38167)
- added — GUI client and Tray Notifier support for Solaris on X86 (Case 46627)
- added — GUI client support for MacOS 64 bit (Case 46337, Case 47149)
- added — Add a general comment when a changelist is removed from a review (Case 46562)
- added — Make cache performance information available in System screen (Case 47315)
- fixed — 'ccollab admin group create' errors with "Group GUID '...' must exist before you can edit it." (Case 47166)

- fixed — 'ccollab admin group' commands error with "Group GUID '...' must exist before you can edit it." (Case 47166)
- fixed — Added documentation for Review Deadline to the Creating a review section and edited the Creating a review section (Case 47184)
- fixed — The link inside of the GIT integration had a typo that was making it appear malformed (Case 47191)
- fixed — Need to fix the AccuRev casing in the standalone GUI (Case 47247)
- fixed — Surround missing from scm config list for CLI
- fixed — Typo on the triggers page in the command for passing a review title (Case 47241)
- fixed — Better explanation of "Active Users" chart (Case 47316)
- fixed — Client does not open browser on MacOS (Case 47051)
- fixed — Improve performance of Fun Fact total review time query.
- fixed — Minimum required Surround client is 2009.1.0
- fixed — Better description of review deadline meaning (case 47239)
- fixed — GUI client does not close properly on MacOS X (Case 46061)
- fixed — Tray Notifier not resizable on Linux (Case 36320)
- fixed — Diff cache misses almost always (Case 47315)

## v5.0.5017 — September 1, 2009

- added — Support Perforce eclipse plugin 560 2009.1.209672 (Case 47023)
- added — Accessibility improvements to chat Comment and Defect tabs 429
- added — Allow expansion of changelist title 357 to the full list of included changelists (Case 46928)
- fixed — Subscription 327 authors not displayed when subscriptions are not editable (Case 46423)
- fixed — Diff viewer 377 screen blank for some files - ArrayOutOfBoundsException on server (Cases 46500, 46737)
- fixed — Review report idle times negative on HSQL (Case 46159)
- fixed — Binary files opened from chat links 286 do the wrong thing (Case 46808)

- fixed — Remove incorrect references to "side-by-side" from diff viewer preferences 379
- fixed — Truncate length of large changelist comments before uploading
- fixed — Users are not sorted in participants selection dropdowns 338 (Case 46594)
- fixed — Review Detail Report includes redundant comments (Case 46183)
- fixed — "null" showing up in version selection (Case 46876)
- fixed — SVN executable "svn" is reset when the GUI client 496 restarts
- fixed — Subversion Eclipse 563 plugin support recognize status "Incomplete" (Case 46930)
- fixed — Prevent Tomcat from filling up the output.log file
- fixed — More consistent layout for binary files (non-PDF)
- fixed — Modified Vault files not found (Case 46754)
- fixed — Modified MKS 752 files in pending change packages compare to wrong predecessor (Case 46511)
- fixed — Miscellaneous Surround fixes (Cases 46348, 45669)
- fixed — Some modified Vault files not being found (Case 46754)

## v5.0.5016 — August 13, 2009

- fixed — Handle errors in Subversion 799 working copy (Cases 46307, 35884)

## v5.0.5015 — August 11, 2009

- added — Links for further information in the installer 70
- added — Vault 5.0 and Fortress 2.0 support (Case 46586)
- added — Support configurable image file types 197 (Case 46632)
- added — Perforce changelist renumbering 765 script to manual (Case 41154)
- fixed — Use configured label for Group 184 in participants error message
- fixed — Scrub invalid dropdown values during upgrade
- fixed — Subversion integration 799 should be case-insensitive on Windows (Case 46561)
- fixed — Installer 70 crash due to malformed xml (Case 35977)

- fixed — do not count sysadmin on license screen with LDAP 119 (Case 45222)
- fixed — Ignore CVS 635 'no longer in the repository' errors (case 46519)
- fixed — Handle empty Subversion 799 comments (Case 46603)
- fixed — Deleted PDF 414 in changelist causing upload failure (Case 46574)
- fixed — Upgrade fails if no diffs cached in database
- fixed — Better XPath 927 output, including line feed between multiple results (Case 43498)
- fixed — Incorrect timezone application in customizable reports on MySQL (Case 45687)
- fixed — Reviews by Changelist and Unreviewed Changes reports broken if filtered on Changelist ID (Case 46534)
- fixed — Use configured SCM options in "Add Diffs" button on GUI Client (Case 45436)
- fixed — Log more info when sending email fails (Case 46728)
- fixed — Surround fixes for added files, parsing 'sscm ls' output, and embedded lines in comments (Cases 46480, 45669)
- fixed — Inactive users whose logins are duplicates of active users could block database upgrade (Case 46745)

## v5.0.5014 — July 31, 2009

- added — Documentation of "--scm none" option (Case 46456)
- added — Button to restore default notification templates and add missing ones
- added — Button to restore defaults on External Diff preference page (Case 46512)
- fixed — Comments on line 0 cause exceptions (Case 46332, 46379)
- fixed — Expose last activity time in review-xml. This is the review completion time for complete reviews (Case 46467)
- fixed — Participants warning appears when group security is not enabled (Case 46157)
- fixed — Old clients added System Adminstrator to review when adding a changelist whose author did not match a Collaborator user (Case 46187) (server-side fix)
- fixed — Surround SCM committed changelists support and misc fixes (Cases 45669, 45823, 45771)
- fixed — Better logging for login errors (Case 46463)

- fixed — More lenient check for loopback address in p4port
- fixed — do not autodetect SCM system when printing command-line help usage (Case 46353)
- fixed — do not create new review for add*diffs commands if no modified files found in diff
- fixed — Cleaner error message for add*diffs command when no diffs found (Case 46135)
- fixed — Display errors returned from 'ccollab addp4diffs' command (Case 46096)
- fixed — Handle odd userId cases correctly (Case 46523)

## v5.0.5013 — July 28, 2009

- added — "Test Connection" button for server-side version control server entry
- added — Create server-side version control server entries automatically from client uploads
- fixed — Error logging status of Subversion connection on server (Case 46249)
- fixed — update-changelist scm trigger finds changelists from a different server

## v5.0.5012 — July 24, 2009

- added — Ability to disable "attach changelist" in browser per server-side version control configuration
- added — Support Objective-C syntax highlighting in .m/.mm files (case 46057)
- fixed — Multiple concurrent database upgrade attempts when the Upgrade button is pressed repeatedly
- fixed — potential temp value collision in upgrade could cause all of one dropdown value to be assigned to an other.
- fixed — Error with "ccollab adddiffs new <before dir> <after dir>" with deleted directory (Case 46239)
- fixed — do not query for database version all the time. (Case 46241)
- fixed — "adddiffs" does not roll up with "addchanges" when no SCM system (Case 45964)
- fixed — If using 'p4-protects-script' do not require connection to Perforce Server
- fixed — Support MKS variants in change package files (case 46212)
- fixed — handle anonymous commits in Subversion (Case 46393)

- fixed — User prefs review subscription form field title "Review Creator" is misleading (Case 46310)

## v5.0.5011 — July 20, 2009

- added — supply P4PORT to 'p4 -s protects' script
- fixed — Upgrades of SSL Connectors
- fixed — require P4PORT to be an external server name, not "localhost"
- fixed — Improve behavior of activity update queries.
- fixed — Change diff caching mechanism to take load off database.
- fixed — put on schema blinders for oracle indices, triggers, and sequences

## v5.0.5010 — July 16, 2009

- added — Option to update ClearCase snapshot views (Case 45296)
- added — Button to accept self-signed Subversion server certificates (Case 46133)
- added — Server-side version control templates
- fixed — working of diff download tooltip on Review overview screen (Case 45985)
- fixed — tray notifier tooltip does not refresh (Case 46038)
- fixed — improve line wrapping algorithm (Case 45959)
- fixed — search for filenames fails to return results. (Case 46067)
- fixed — Command line client throws exception if user cancels (Ctrl-C) when prompted for password.
- fixed — scm triggers use server-side version control mapping to find correct server (Case 45841)
- fixed — scm triggers print extra output (Case 45963)
- fixed — groups created in Collaborator enforced after switching to CodeReviewer license (Case 46040)
- fixed — recent Participants link does not work for users not in the group (Case 46055)
- fixed — Improve file headings; especially for diffs.

- fixed — Updatechangelist trigger should not touch changelists descriptions that do not involve the trigger
- fixed — Update perforcetrigger usage message to indicate that it is deprecated.
- fixed — do not munge whitespace in p4 specfiles when we modify them (Case 46088)
- fixed — Error in Eclipse plugin "fix configuration" action item (Case 46164)
- fixed — SQL Server date format not specified. (Case 44329)
- fixed — new trigger algorithm backwards-compatible with old clients when there is only one Perforce server
- fixed — Improve performance of some queries under Oracle

## v5.0.5009 — July 7, 2009

- updated — jPDFImages library to version 2.13.
- updated — Tomcat to version 6.0.20
- added — Perforce 'Server Address' for version control identification/matching.
- fixed — support for http.nonProxyHost (proxy exceptions) (Case 45855)
- fixed — Check that database supports views before attempting to create views. (Case 45616)
- fixed — Interpretation of some svn output is broken (Case 45775)
- fixed — Wrong Vault predecessor for modified files in VSS mode (Case 45839)
- fixed — Enforce CodeReviewer database restrictions (Case 41184)
- fixed — Enforce CodeReviewer LDAP restrictions
- fixed — Cannot open .zip file from file download (Case 45903)
- fixed — Remove administration screen for configuring Legacy GUI Client versions; it is not supported in 5.0.
- fixed — Support Perforce multi-file diff format in ccollab adddiffs (Case 45420)
- fixed — Suppress 'p4 info': Client unknown warning (Case 45963)
- fixed — Idle users being counted as logged in (Case 46015)
- fixed — Support Perforce client specs with exclusionary mapping rules (Case 45986)

## v5.0.5008 — June 29, 2009

- added — Reminder for administrators to migrate from the trial database
- added — Link defects in the defect reports
- added — Syntax highlighting for Ruby
- added — Configure P4V/P4Win integrations in the installer
- fixed — Support for SVN checkouts at drive roots
- fixed — Cannot find Accurev workspace (Case 45645)
- fixed — Periodically checkpoint HSQLDB
- fixed — "Update From Smart Bear" sends the user-provided company key
- fixed — NPE in comment promotion
- fixed — External diff does not work with too many files (Case 45667)
- fixed — Improve UI for removing a disabled user from a review (Case 45755)
- fixed — Error uploading TFS shelvesets with new files (case 45709)
- fixed — Handle MKS change packages in submitted state (case 45631)
- fixed — No scroll bars in chat window for binary files (Case 45519)
- fixed — AccuRev NPE in ccollab addstream (Case 45764)
- fixed — Better error message for Perforce authentication problem (Case 45778)
- fixed — Incorrect activity start dates (Case 45759)
- fixed — Selecting a marker should scroll it into view (Case 45265)
- fixed — Upgrade 500 to 501 bug on MySQL
- fixed — Upgrade speed fix
- fixed — License check causing spurious log messages.
- fixed — Once disabled, role cannot be reenabled (NPE) (Case 45845)
- fixed — NPE in "Moving On" section (Case 45696)

## v5.0.5007 — June 15, 2009

* FIRST GA RELEASE OF v5.0! *

- added — Reporting of slow queries in the server log to help diagnose slow operations
- fixed — Slow browser response on some reviews (Case 44872)
- fixed — NPE and divide by zero in client commit (Cases 45480, 45497)
- fixed — NPE in comment promotion
- fixed — Updated HSQLDB to 1.8.0.10 to fix possible data corruption issues with embedded database.
- fixed — Changelists from different SCMs must be kept separate everywhere. (Case 45234)
- fixed — Files not showing up in reviews; TFS versions from early beta clients causing issues (Case 45504)
- fixed — Changelist rollup should be based on SCM config, where available (Case 45338)
- fixed — group not assigned when review created by client (Case 45511)
- fixed — j_security_check login bug (Case 42864, Case 45102, probably more)
- fixed — Long version names (from ClearCase) cause display issues in diff viewer (Case 45261)
- fixed — Misleading error message on ccollab admin trigger create-review (Case 45540)
- fixed — Verify for P4 2009.1: "p4 info" now reports the server address (Case 45635)
- fixed — create-review trigger does not set "last" review variable (Case 45437)
- fixed — Database dump fails when review contains URL versions (Case 45575)
- fixed — NPE when trying to archive files.

## v5.0.5005 (beta) — June 4, 2009

- (fixes from release 4.0.860 1153)
- (fixes from release 4.0.859 1153)
- updated — Manual updated with various new features.

- added — Filters for users on the user administration screens
- added — New variable substitutions: defect.isexternal, defect.externalname, review.group.title, and review.group.guid.
- added — Side by side document review/compare menu working (Case 44443)
- added — MKS change package and addversions support (Cases 43812, 44416, 44540)
- added — User preference for tab width (Case 40604)
- fixed — Company contact information not correct in license files.
- fixed — Improved page load times for review overview page.
- fixed — Improved performance of database query builder (Case 45029)
- fixed — NPE when changing general settings in reviewer.
- fixed — Review list report displays bogus warning when number of rows exactly equals capped number (Case 44525)
- fixed — URL's with embedded credentials not properly linked (Case 44958)
- fixed — create-review trigger argument --review-id-regex <value> should be optional (Case 45091)
- fixed — Compare list/diff viewer headings need more accurate information when using diffs (Case 44672)
- fixed — Next file order does not match compressed tree or tree views (Case 45098)
- fixed — Misleading label for new chat area when reviewing documents or images.
- fixed — Cannot jump back to current conversation when on a different page of a document (Case 44229)
- fixed — NPE in GUI client (Case 44914)
- fixed — Line numbers lost for comments on unchanged deleted files (Case 44819)
- fixed — NPE when reverting an unsupported file (Case 45092)
- fixed — Separate multiple changelist description text for readability (Case 44677)
- fixed — Error messages not cleared in attach changelist/url/file dialogs (Case 44726)
- fixed — Firefox sometimes incorrectly guesses RSS encoding (Case 45129)
- fixed — addgitdiffs command now gets more information from git diffs

- fixed — Unable to remove last item from a multiselect custom field (Case 44379)
- fixed — Added command to support Perforce changelist renumbering (Case 44978)
- fixed — Restrict access to fix defect global option does not allow defect creator to fix (Case 45202)
- fixed — Chat box opens for wrong line of code (Case 45149)
- fixed — Support ClearCase 7.1 and int'l date formats (Cases 44914, 45122)
- fixed — Added user state tooltip in the participants section of review overview (Case 43258)
- fixed — addchangelist with Perforce not uploading correct base content (Case 44577)
- fixed — Add configuration for server logging.
- fixed — Workaround for pool users not available.
- fixed — Home page tab counters not updated when content changes (Case 45238)
- fixed — Changed default poke notification text (Case 45258)
- fixed — Corrected timezone for date displays in tables such as the User/Admin screen
- fixed — String replacements fail with $ (Case 45353)
- fixed — Support multi-line environment variables (Case 43002)
- fixed — User subscription should not (appear to) be pre-populated with the first entry (Case 45317)
- fixed — NPE in content cache diagnostic (Case 45376)
- fixed — multi-select two panel filter does not display correctly on IE (Case 44352)
- fixed — respect notification limit in RSS on embedded database (Case 45129)
- fixed — Text from previous review can appear on new review screen (Case 45324)
- fixed — AccuRev getting incorrect previous version ( Case 45257)
- removed — Old-style defect reports -- use new defect reports instead. (Case 45215)

## v5.0.5004 (beta) — May 14, 2009

- added — Support addversions for Perforce SCM
- (fixes from release 4.0.858 1153)

- fixed — Cannot save General Settings in 5.0 (Case 44759)
- fixed — Update documentation for HTTPS configuration
- fixed — duplicate "Adding file" messages with "ccollab addchanges"
- fixed — Remove extraneous "Checking if file exists" messages printed to console with "ccollab"
- fixed — Defects can be added more than once (Case 42403)
- fixed — TFS autodetect causes error when TFS is installed and a file is unmanaged
- fixed — Upgrades fail against SQL Server 2000 (Case 44527, 44730)
- fixed — Next unread comment and next defect buttons do not always work (Cases 41695, 44339)
- fixed — FileMetrics for documents should include page count, but not line counts.
- fixed — Document extension matching (.pdf) is case sensitive (Case 44732)
- fixed — Document review should be unavailable in CodeReviewer
- fixed — Turkish locale issue with capital i (Case 44763)
- fixed — Large changed regions in over under have strange scrolling behavior on line click (Case 46432)
- fixed — Can create defect without supplying required custom fields (Case 40500)
- fixed — Status in review materials is wrong for added files (Case 44839)
- fixed — Command line SCM tokens were case sensitive (Case 44618)
- fixed — Bad VSS login argument on ss.exe command line (Case 44745)
- fixed — Line number validation fails in binary file and URL review (Case 44441)
- fixed — Support for non-ascii filenames in Subversion (Case 43430)
- fixed — Java syntax highlighting rules do not include byte and short
- fixed — Change MIME type of external diffs to be vendor specific (application/vnd.smartbear.cc-diff)
- fixed — URL and document review pane include unnecessary line difference pane (Case 44348)
- fixed — Post-commit review show rework when there was no rework

- fixed — Update documentation on trigger command line syntax (Case 43226)
- fixed — Email addresses not being properly converted to links
- fixed — Bogus error messages in admin screens when inputs are unsaved (Case 40999)
- fixed — Surround password showing up in cleartext (Case 43226, 44954)
- fixed — Custom report filters for meta-data drop-downs need to show disabled items (Case 44504)
- fixed — Diff viewer keyboard shortcuts dialog fails in IE (Case 44338)
- fixed — ArrayIndexOutOfBoundsException when creating svn client config (Case 44961)

## v5.0.5003 (beta) — April 20, 2009

- added — refactor review-xml to handle line number/location (Case 44613)
- (fixes from release 4.0.857 1153)
- fixed — Make the timezone drop-down items findable and readable.
- fixed — reverting on group edit page displays "updated" message (Case 44353)
- fixed — Transparency slider and image toggle not disabled when only one image and preference for newer content on right (Case 44475)
- fixed — Post-commit reviews show files as reworked, even when they are not. (Case 44493)
- fixed — Stack overflow when uploading large files. (Case 44497)
- fixed — Local file status is displayed incorrectly with Subversion (Case 44448)
- fixed — Server fails to shut down running on Windows with Java 5 (Case 44633)
- fixed — Tutorial mode preference is ignored in some places (Case 43229)
- fixed — Improve memory footprint and performance of PDF rendering (Case 44519)
- fixed — Uploading URL message is hidden because the dialog is not tall enough. (Case 44661)
- fixed — Fix addversions command
- fixed — Reordering of custom fields can fail (Case 44692)
- fixed — review-xml includes duplicate conversations (Case 44702)

- fixed — Stack overflow in Eclipse plugin with svn:externals (Case 42824)
- removed — Attach URL feature not supported in CodeReviewer (Case 44555)
- removed — Perforce ACL feature not supported in CodeReviewer (Case 44542)
- removed — Groups feature not supported in CodeReviewer. (Case 44553)
- removed — System wide message feature not supported in CodeReviewer. (Case 44557)

### v5.0.5002 (beta) — April 6, 2009

* FIRST RELEASE OF v5.0 BETA! *

## 11.2.10 Version 4

### v4.0.864 — February 4, 2010

- fixed --- List report results improperly cached (Case 49572)
- fixed --- LDAP lockout can occur if Collaborator is used infrequently. (Case 49526)
- fixed --- "Too many open files" error in Content-Cache Diagnostic (Case 49687)

### v4.0.863 — September 1, 2009

- added — Support Perforce Eclipse plugin 2009.1.209672 (cmdline wrapper only)
- fixed — Subversion Eclipse plugin support recognize status "Incomplete" (Case 46930)
- fixed — Some modified Vault files not being found (Case 46754)

### v4.0.862 — August 19, 2009

- added — Support MKS variants in change package files (case 46212)
- added — SQL Server JDBC 2.0 driver documentation
- fixed — Update documentation for new name of Vault 4.1.x client .jar download
- fixed — Handle empty SVN comments

- fixed — Multiple button upgrade bug wherein multiple users click the "Upgrade Database" button and
- multiple upgrades are attempted simultaneously
- fixed — Error in Eclipse plugin "fix configuration" action item (case 46164)
- fixed — .vmoptions file extension

## v4.0.861 — July 8, 2009

- fixed — Handle MKS change packages in submitted state (case 45631)
- fixed — p4 -Q does not work before *client* version 2005.2
- fixed — Support multi-line environment variables (Case 43002)
- fixed — Crash in ccollab addchangelist new (Case 45684)
- fixed — Support for svn checkouts at drive roots (Case 45707, Case 45614, Case 45717)
- fixed — Error uploading TFS shelvesets with new files (case 45709)
- fixed — Better error message for Perforce authentication problem (Case 45778)
- fixed — Wrong Vault predecessor for modified files in VSS mode (Case 45839)
- fixed — SVN output interpretation fix (Case 45775)

## v4.0.860 — June 3, 2009

- fixed — NPE in ClearCase 666 hashcode (Case 44189)
- fixed — Error parsing Team Foundation 738 fr-CA date format (Case 45373)
- fixed — TFS 738 usernames with leading backslash (Case 45373)

## v4.0.859 — June 1, 2009

- added — Better support for higher Perforce 764 security levels (Case 44899)
- added — Support ClearCase 666 7.1 dates (Case 44914)
- added — Support for wildcards in ccollab addfiles (Case 45201)
- fixed — Remove benign log warnings about not able to contact license server (Case 45003)

- fixed — Multiple Vault configurations not working (Case 44642)
- fixed — Custom field views 914 skipped deleted items (Case 44504)
- fixed — Multiple Authors allowed even when Maximum # of Authors 285 is 1 (Case 45107)
- fixed — Clicking 'next' does not do anything for Submitted Perforce Changelists 562 on Eclipse 3.4
- fixed — Modified files not found if username case differs from Vault (Case 44642)
- fixed — Handle Perforce plugin 560 authentication when password not saved (Case 44899)

## v4.0.858 — May 6, 2009

- added — Support for subversion 1.6
- added — UI to generate new Node ID (Case 44480)
- fixed — In some cases, reordering of custom fields can fail (Case 44692)
- fixed — Stack overflow in eclipse plugin with svn:externals (Case 42824)
- fixed — TFS autodetect causes error when TFS is installed and a file is completely unmanaged
- fixed — Locale sensitive bug in embedded database initialization (Case 44763)
- fixed — Bad VSS login argument on ss.exe command line (Case 44745)
- fixed — Custom report filters for meta-data drop-downs need to show disabled items (Case 44504)
- fixed — Exception when using subversion at a drive root (Case 44961)

## v4.0.857 — April 15, 2009

- added — Support for Perforce eclipse plugin 560 2008.2.195317 (Case 43854)
- fixed — Incorrectly reporting "num-defects" as "num-comments"
- fixed — MKS 752 trunk version limit error (Case 44275)
- fixed — MKS 752 history limited to 200 versions (Case 44275)
- fixed — MKS 752 password appears in debug log (Case 44322)
- fixed — Improper error handling when database fails to initialize (Case 44462)

- fixed — Eclipse plugin support for ClearCase 553 (Case 44561)
- fixed — Only list shelvesets 738 for configured user
- fixed — Broken manual link (Case 44506)

## v4.0.856 — March 30, 2009

- added — Ability to upload diffs from a file by name
- fixed — Update documentation on Tomcat auto-deploy for ROOT.xml changes (Case 43973)
- fixed — Updated 'ccollab admin batch' help text
- fixed — Fixed lack of support for multi-lined strings in PHP
- fixed — Exported reports should use UTF-8 character set so as to not garble any characters (Case 41853)
- fixed — Cannot parse certain CVS diffs (Case 44090)
- fixed — Fix broken admin custom fields icon
- fixed — Missing files in MKS changelist (Case 43812)
- fixed — NPE on uncontrolled CVS file (Case 43962)
- fixed — ClearCase host name incorrect (Case 44007)
- fixed — Cannot create roles or templates (Cases 44172, 44255)
- fixed — Engine.NotificationCreate() method should be public (Case 44276)

## v4.0.855 — March 5, 2009

- added — updates to Examples.java
- fixed — Linux client connection problems to MKS Integrity Server (Documentation update) (Case 42389)
- fixed — Split Role setting 'Can change defects' into 'Can change own defects' and 'Can change other user's defects' (Case 43765)
- fixed — Extra space at end of url causes problems in Eclipse plugin (Case 43798)
- fixed — Downloading files from a review with duplicate file names causes a corrupted ZIP file (Case 43796)

- fixed — SourceGear Fortress support in Vault (Cases 42962, 43783)
- fixed — NPE in AccuRev due to "file does not exist" (Case 43841)
- fixed — Long changelist description with Unicode characters and Oracle back-end fails to upload (Case 43535)
- fixed — Run each task in a separate thread (Case 43790)
- fixed — ClearCase 'Operation requires a view' exception with supporting documents in review (Cases 43898, 43920)

## v4.0.854 — February 23, 2009

- fixed — Perforce ccollabupdatechangelist trigger slow for many thousand files (Case 43688)
- fixed — Search results can return reviews that user cannot access (Case 43722)
- fixed — Unable to parse file extension in ClearCase extended paths, prevents syntax highlighting (Case 43746)

## v4.0.853 — February 11, 2009

- fixed — Encode server logs in UTF-8
- fixed — File permission error writing MKS log file under Linux (Case 43241)
- fixed — Javascript bug in classic diff view
- fixed — Bug id markup and HTML encoding conflict (Case 43479)
- fixed — Problem parsing incomplete CVS diffs (Case 42803)
- fixed — Support unmanaged files in MKS directories (Case 43333)
- fixed — Support unmanaged files in TFS directories (Case 43375)
- fixed — Improve chat performance with large numbers of conversations
- fixed — Exception on renamed TFS files when uploading rework (Case 43509)
- fixed — Cannot access view column names in Oracle (Case 43514)

## v4.0.852 — January 16, 2009

- fixed — Encode server logs in UTF-8.

- fixed — Do not create duplicate users that vary only base case (effects Oracle and embedded database users) (Case 42741, 42420)
- fixed — Do not add authors to a review if their user account is inactive (Case 43056)
- fixed — If Subversion username is specified, always specify a password (Case 43180)
- fixed — If an XML file includes a charset marker for an unsupported character set, try to recover using autodetection (Case 43182)
- fixed — Add more context to CVS diffs (Case 42803)
- fixed — Oracle limited to 127 switches in a CASE statement
- fixed — Update Vault documentation
- fixed — Handle no-longer-scheduled-to-be-added and -deleted directories in Subversion (Case 43204)
- fixed — Perforce '-Q' option only supported after 2005.2.
- fixed — New file in ClearCase activities appear as modified (Cases 39166, 43227)

## v4.0.851 — December 16, 2008

- fixed — Login should accept 64 characters to match the username database field. (Case 42836)

## v4.0.850 — December 12, 2008

- fixed — Perforce changelists with high Unicode characters in filenames or changelist descriptions get text corruption. (Case 41819)
- fixed — Add p4charset global option for interacting with Unicode Perforce servers.
- fixed — Build with Vault 4.1.4 API (Case 42625)
- fixed — Launch browser in daemon thread (Case 42826)

## v4.0.849 — December 3, 2008

- fixed — Peak usage by day still broken (Case 42748)
- fixed — Support TFS versions with deletion ids (Case 42331)
- fixed — ScmRevertedLocalCheckout cast error (Case 42450)

- fixed — StarTeam uploads file with multiple files in different subdirectories (Case 42523)
- fixed — Argument replacement bug (Case 42525)
- fixed — Multiple-file external diff on IE 6; cache headers cause file not found (Case 42601)
- fixed — Error when loading transaction from AccuRev without workspace (Case 42604)
- fixed — Unified diffs from Mercurial break diff parser (Case 42500)
- fixed — Allow changing case of login name (Case 42675)
- fixed — P4Win integration pause on error (Case 42677)
- fixed — No way to resolve text when review content is decoded in wrong character set (Case 41819, 42536, 42743)
- fixed — Support Perforce sandboxes on UNC paths (Case 42726)
- fixed — Phone numbers not displayed in tooltips (Case 42707)

## v4.0.847 — November 13, 2008

- added — Associate .inl files with C++ syntax highlighting (Case 42188)
- added — Documentation of Active Directory configuration with security groups.
- fixed — Show/hide previously uploaded changelists toggle does not remember state (Case 41246)
- fixed — Better handling of corrupted paswords in config files (Case 42283)
- fixed — Make client commit action get all most recent versions instead of active changelists.
- fixed — Do not display SCM passwords in debug log
- fixed — Duplicate user accounts with leading/trailing spaces (Case 42420)
- fixed — MKS IllegalArgumentException (Case 42252)
- fixed — Add changelist id to review-xml (Case 42520)

## v4.0.846 — October 28, 2008

- added — Support for Visual SourceSafe diffs (Case 41339)
- fixed — Database connections dying "randomly" causing intermittent web site failure and stopping the activity-update thread

- fixed — ClearCase server paths are canonicalized
- fixed — Wrong ClearCase predecessor after version list edit (Case 41561)

## v4.0.845 — October 13, 2008

- fixed — restrict ccollab admin review-xml according to access rules (Case 42080)
- fixed — consistent behavior for ignored and/or unmanaged files in addchanges and addfiles
- fixed — Make RPM installer cleanup of old files optional (Case 41786)
- fixed — Addfiles fails on ucontrolled files in CVS directory (Case 41590)
- fixed — Disabled users should not receive notifications (Case 41607)
- fixed — Use correct system admin login name in "license exceeded" error message
- fixed — "Total Person Time" not the same on review summary and review detail (Case 41118)
- fixed — Open home pages will not count towards licensing (Case 41804)
- fixed — Count chat refreshes against licensing only if the user is active.
- fixed — Prepend External Diff path disambiguation instead of appending so file extension is not changed.
- fixed — Config-only dump skipped review templates' custom review and defect field associations (Case 41826)
- fixed — Add Office 2007 file extensions to default binary formats.
- fixed — Error running ccollab set collab command with no argument. (Case 42028)
- fixed — Fix NullPointerException adding uncontrolled files (Case 41988)

## v4.0.843 — September 23, 2008

- added — Support for Vault proxy settings (Case 40815)
- added — Scripting 926 section to manual
- added — Mirror Defects to external issue-tracker 928 scripting tutorial to manual
- added — Add checklist to new Review scripting tutorial to manual
- added — Sync Perforce Users 933 scripting tutorial to manual

- added — Time to run report to report information display
- added — Jump next/prev should not wrap without confirming. (Cases 41431, 41594)
- added — Upload comment option for Starteam (and other) add*diffs commands (Case 41745)
- added — MKS changes for subsandboxes
- fixed — NPE in getRecentlyUsedServers with Team Foundation 2008 (Case 41618)
- fixed — Single-click flicker when double-clicking tray notifier
- fixed — Text variable typo
- fixed — Missing TFS files when server and local paths differ in case (Case 41468)
- fixed — Gap in usage graphs (Case 41396)
- fixed — Chat on line 1000000 should be disallowed if possible and promoted back into reasonable range. (Case 41399)
- fixed — Oracle "table or view does not exist" bug (Case 41184)
- fixed — Detection of SourceForge ClearCase Eclipse plugin (Case 41646)
- fixed — External Diff launcher error for Perforce Pending Changelists (Case 41515)
- fixed — Compare drop-down needs clarification (Case 41644)
- fixed — Vault takes too long to find modified files (Case 41558)

## v4.0.842 — August 29, 2008

- added — ccollab commit prompt for upload comment (Case 41055)
- added — support for GUI Client on Solaris (Case 41144)
- added — allow setting multiple-line custom field values on the command line by getting the field's value from a file. (Case 41176)
- added — 'ccollab admin batch'
- added — Support alternate Team Foundation user names
- added — "Open GUI" menu item to tray notifier (Case 41359)
- added — "Support" email links to GUI Client and Tray Notifier (Case 41343)

- added — "--xpath" and "--xsl" options to 'ccollab admin review-xml'
- fixed — Support alternate Team Foundation user name (Cases 41164, 41196)
- fixed — Fix unified diffs for Lua (Case 41195)
- fixed — role configuration admin screen where drop-down boxes were too wide for the form, causing the page to be extremely wide
- fixed — Clearcase error with Unix paths (Case 41100)
- fixed — make next/prev buttons only stop on differences in current comparison (Case 41001)
- fixed — ability to set custom field drop-down items from command-line (Case 39720)
- fixed — ccollabgui client not installed properly by Linux RPM (Case 41115)
- fixed — multi-file external diff (Case 41172)
- fixed — Unicode names get abbreviated incorrectly (Case 41150, 41300)
- fixed — Admin users should be allowed to add changelists to any review (Case 41330)
- fixed — ClearCase eclipse plugin integration cannot find view (Case 41302)
- fixed — User comboboxes fail with internationalized names.(Case 41205)

## v4.0.841 — August 20, 2008

- added — Command to set custom field dropdowns from the command line (Case 39720)
- added — Support for GUI client on Solaris (Case 41144)
- fixed — Change contact email address for licensing issues to the appropriate Smart Bear email address. (Case 41037)
- fixed — Role configuration admin screen select boxes too wide.
- fixed — Buttons on Linux displayed even though no SCM selected. (Case 41050)
- fixed — Clear Case error with Unix paths. (Case 41100).
- fixed — Server can fail to start if installation path includes a space on Unix.
- fixed — After changing revisions, Next and Prev buttons stop on differences from prior revision (Case 41001)
- fixed — GUI client not installed correctly by Linux RPM (Case 41115)

## v4.0.840 — August 5, 2008

- added — Faster Vault integration
- added — More support for MKS integration
- added — Eclipse v3.4 support for automatic upgrade site (Case 40495)
- added — Auto-detect tf.exe location for TFS 2008 (Case 40852)
- fixed — Enterprise organization field width too small (Case 40263)
- fixed — Inconsistent letter casing when uploading mixed-case paths from Windows (Case 40170)
- fixed — Notification emails encoded in system default character set can be garbled (Case 40693)
- fixed — Added more keywords to TCL syntax coloring (Case 40620)
- fixed — Vault was prompting for each file instead of for all files together (Case 40619)
- fixed — Proper handling for file renaming in TFS integration (Cases 39946, 40018)
- fixed — Erroneous error for "tf properties" (Case 40613)
- fixed — Support for AccuRev integration on OpenJDK (Case 40699)
- fixed — Sundry ClearCase integration issues (Cases 41026, 40198, 40132, 40437, 40619, 39766)
- fixed — ClearCase "unable to access" error when loading from root of Windows dynamic view (Case 41026)
- fixed — Filling output.log with error messages when a session is created and invalidated after response bytes begin (Case 41043)

## v4.0.839 — July 14, 2008

- added — Support for Subclipse 1.4 (Case 40457)
- added — ccollab browse --review option accepts "last" and "ask"
- fixed — ccollab adddiffs should put source info in changelist comment (Case 40550)
- fixed — do not error if there are zero review custom fields (Case 40566)
- fixed — Clarify options for ccollab login (Case 40240)

- fixed — do not print Subversion password in GUI client (Case 40649)

## v4.0.838 — July 7, 2008

- added — Defect permission configuration setting: do not allow edit/delete defects even if the user was the creator.
- fixed — Upgrade from 2.1.x fail with SQL Server 2005

## v4.0.837 — July 2, 2008

- added — Support for Subversion 1.5
- added — Subcommand 'admin review copy-participants'
- fixed — Added indexes for better query performance (Case 40290)
- fixed — User cannot delete a defect they created (Case 40399)

## v4.0.836 — June 24, 2008

- added — File commit support for TFS and ClearCase
- fixed — Better error messages in Perforce triggers
- fixed — Report errors in p4 print
- fixed — Incorrect metrics for "loc changed" in detail report
- fixed — Sort user list based on activity (Case 39030)
- fixed — Some metrics not working when using internal database (Case 40006)
- fixed — When large numbers of users in system, new review screen fails to load (Case 40339)
- fixed — Upgrades from early 4.0 versions fail (Case 40332)

## v4.0.835 — June 17, 2008

- added — Add Subversion diffs UI in GUI client 804
- added — Add CVS diffs UI in GUI client 636
- added — Add Perforce diffs UI in GUI client 769

- added — Add StarTeam diffs UI in GUI client
- added — Add AccuRev diffs UI in GUI client 623
- added — Role permission 281 to allow userto modify, but not delete, defects (Case 39630)
- added — Vault 4.1 support
- added — Syntax highlighting for TCL
- added — JMX monitoring 155 of licensing and users
- added — Peak usage chart 212 of license usage with accurate data
- added — Browse... button to GUI client 496 SCM dialog
- added — Track number of rejected logins due to licensing issues (Cases 39775, 39767)
- added — Integrated support for checking in reviewed materials
- fixed — improved performance of adding ClearCase versions with addversions and addactivity (Case 38732)
- fixed — improved performance of review overview screen (Case 39662)
- fixed — browser integration on OS X (CAse 39690)
- fixed — partial fix for external diff config manifest issues (Case 39752)
- fixed — autodetect tf.exe location (Case 39732)
- fixed — attach materials page missing image (Case 39526)
- fixed — Names with apostrophes and hyphens are not properly abbreviated (Case 39750)
- fixed — Optimize/fix adding versions by name (Case 39281)
- fixed — Do not change the case of mixed-case names when abbreviating
- fixed — No vertical scrollbar in classic view in IE7 (Case 39830)
- fixed — Include external diff button for binary files (Case 39754)
- fixed — User list should not be case sensitive (Case 39869)
- fixed — Performance improvement; less frequent access to assignments table (Case 39590)
- fixed — ccollab addchanges fails on Subversion unmodified file after branch (Case 39880)
- fixed — Bad data causes dump to not restore properly (Case 39623)

- fixed — Team Foundation script output appearing in version content (Case 39887)
- fixed — Links in external defects doubly encoded (Case 39921)
- fixed — Make "private" review field available in reports (Case 39900)
- fixed — Improve performance of default review reports
- fixed — Files missing from diff (Case 40002)
- fixed — Diff view does not reload entire page when options change
- fixed — Diff view preference for wrap lines not honored in single version view (Case 39885)
- fixed — Cannot cancel "Track Externally" by choosing "Edit Defect" instead (Case 39073)
- fixed — Review-only dumps corrupted (Cases 39975, 40060)
- fixed — Do not log passwords in cleartext on Windows (Case 40010)
- fixed — Firefox 3 popup calendar sized incorrectly (Case 39723)
- fixed — Delay review creation so users do not create spurious reviews (Case 40031)
- fixed — Tomcat logging should be enabled by default at INFO level.
- fixed — Handle "No Data Given" error parsing spurious TFS properties (Case 39946)
- fixed — Exception in Perforce trigger when changelist description is empty (Case 40085)
- fixed — GUI client should explain that it does not support default Perforce changelist (Case 39948)
- fixed — Filenames with characters outside default character set get garbled (Case 40124)
- fixed — 'ccollab addchanges' on deleted filenames (Case 39483)
- fixed — Error parsing Team Foundation dates with DBCS chars (Case 40145)
- fixed — Performance imrpovement when loading review phases (Case 39860)

## v4.0.834 — May 21, 2008

- added — Custom report field for number of open defects (Case 39234)
- added — Added 'force-new-browser' option to force new browser window
- added — More validation of Browser and Server URL values in GUI Client

- added — Added 'cvs-exe' global option in case CVS executable is not in PATH
- added — Performance improvements to the diff viewer
- added — Added 'accurev-exe' global option in case AccuRev executable is not in PATH
- added — Added 'svn-exe' global option in case Subversion executable is not in PATH
- added — Eclipse plugin Subclipse integration 563 prompt to automatically switch to SVNKit if necessary
- added — Ability to disable reporting in the Web UI (Case 39550)
- added — Better defect usage description
- fixed — Exception in Database Diagnostic (Case 39323)
- fixed — AccuRev NPE on added file (Case 39195)
- fixed — Prevent users from deleting/canceling reviews (Case 38989)
- fixed — Consistent ordering of changelists by SCM in review materials (Case 39159)
- fixed — Eclipse plugin 551 should prompt to save modified files before uploading them (Case 34752)
- fixed — Scm systems not being mapped correctly when uploading changelists
- fixed — Subversion moves (add with history) should show the metrics from the diff (Case 39512)
- fixed — Error when changing default value (Case 38091)
- fixed — Deleted files not included in external diff package (Case 39529)
- fixed — Handle error getting modified files in Add to Review wizard (Case 36685)
- fixed — Perforce (P4V) integration 786 does not report errors (Case 39532)
- fixed — Perforce (P4V) integration 786 does not work when Workspace not specified (Case 39532)
- fixed — Last line of code removed from review if blank (Case 38957)
- fixed — "AccuRev not in working directory" message comes out as "unexpected error"
- fixed — Update user last activity much less frequently (Case 39590)
- fixed — Deadline status should not display when deadlines disabled

## v4.0.833 — April 28, 2008

- added — GUI client remembers last four (4) server URL's 502 (Case 38271)
- added — Support for locale-specific TFS dates.
- added — Flag to disallow non-author uploads 188 (Administrator Setting) (Case 38619)
- added — Added "browse" subcommand to clients
- fixed — Subscriptions should not be applied for users whose accounts are disabled (Case 39270)
- fixed — Fix addchangelist help for TFS shelvesets
- fixed — Include summary metrics for all versions in the review detail report. (Case 39228)
- fixed — Add indicator to the "Compare" menu to indicate what is being shown (Case 39219)
- fixed — External diff launcher arguments should be quoted
- fixed — Diff viewer should have a minimum on the number of skipped lines (Case 38119)
- fixed — Add a meaningful error message for unsupported TFS files
- fixed — Fix some bad error messages in client. (Case 39426)
- fixed — Give a better approximation of disk usage in archiving (Case 39333)
- fixed — Diffs not showing with addsvndiffs
- fixed — ccollab --debug option tells user where log is being saved
- fixed — Subclipse integration does not support anonymous repositories (Case 39203)
- fixed — Database dumps are incomplete when dumping a single review ( Case 39420)

## v4.0.832 — April 10, 2008

- added — Support for AccuRev Eclipse plugin
- fixed — Solaris does not understand test -e (Case 39079)
- fixed — Diff uploads appear to pick incorrect version (Case 38974)
- fixed — Defect description on review overview omits anything that looks like a tag (Case 38708)
- fixed — Notification emails stop working after server restart (Case 39102, 39066, 39173)

- fixed — Case error in custom field documentation (Case 39197)
- fixed — Performance improvement generating diffs (Case 39079)

## v4.0.831 — April 7, 2008

- added — User creation trigger 206 (Case 38762)
- added — Allow users to set skip unchanged 372 option in diff viewer (Case 39051)
- added — Option to disallow non-author uploads 188 (Case 38619)
- added — ccollab set prompts if no option value specified (Case 36979)
- added — ccollab set displays value for all options if no option specified (Case 36724)
- added — Support GIT diff variant with ccollab adddiffs (Case 39048)
- added — External diff preset 615 for Beyond Compare (Case 38988, 37588)
- added — Collaborator GUI Client support for Linux x86_64
- added — Syntax 381 highlighting for SQL (Case 38300)
- added — Tooltip showing when reworked files 357 were last updated
- fixed — Correctly set windowing system parameter in Linux RPM client installer
- fixed — Collaborator GUI Client should persist size and location (Case 38906)
- fixed — Improve progress messages for ccollab addchanges (Case 36423)
- fixed — IE "remember me" cookies (Case 38947)
- fixed — syntax highlight "sbyte" as a keyword in C#
- fixed — Update P4V Tools import file for P4V 2007.3 (Case 38994)
- fixed — Syntax highlighting for PHP (Case 36489)
- fixed — When there are too many users, review creation page is slow (Case 38688)
- fixed — Better support for nested Subversion working copies (Case 38979)
- fixed — Only administrators can run ccollab admin syncusers 785
- fixed — Update SQL Server drivers; drop support for SQL Server 7.0
- fixed — ccollab adddiffs fails if files have different names (Case 37940)

- fixed — Minor performance improvements in diff emitting code
- fixed — do not display time in review deadline (Case 38948)
- fixed — Bad links in server upgrade section
- fixed — ccollab addfiles does not pick up SCM configuration (Case 36538)
- fixed — Review not displaying files (Case 39093)

## v4.0.829 — March 26, 2008

- added — Support for GIT-style diffs in ccollab adddiffs
- added — Detect XML "encoding" header attribute and use to parse XML files with correct encoding (Case 38933)
- added — Indicate the number of reworks a file has gone through in the "Status" column of the review summary page 357 instead of just saying "Reworked". (Case 38875)
- fixed — Error trying to upload or view files not in UTF-8 or certain binary files (Cases 38925, 38913, 38933, 38941)
- fixed — "Log user off" link should be present only if the account is logged in according to floating-seat rules (for example, auto-log-off after 1 hour), not by whether the user has logged off manually.
- fixed — "New comment" form should be disabled when no line is currently selected
- fixed — ClearCase Windows version paths garbled in viewer drop-down

## v4.0.828 — March 25, 2008

- added — Separate licensing button "update from smart bear" 96 to reduce confusion
- added — System property to specify the cache sizes for line parse and syntax coloring cache
- added — JMX beans for cache sizes; framework for other JMX beans
- added — More review information in notification emails (Case 38618)
- added — Include server configuration files in debugging dump 174
- added — Debug and migration dumps 174 use form to select options instead of list of links
- added — Option to include server logs in system data dumps 174
- added — Incorporate review dumps 175 into the new data dump form

- added — Team Foundation Server 738 support (beta)
- added — Add "log this user off" 223 to administrative user list (Case 38845)
- fixed — improved performance of caches for syntax and diff highlighting.
- fixed — ccollab addchanges should not create review when no files are selected (Case 36649)
- fixed — Error on some JVMs running addcvsdiffs 643 (Case 38723)
- fixed — Getting blank screen when trying to input new license code (Case 38830)
- fixed — Binary characters blocking chat from working (Case 38825)
- fixed — User Login Prompt field and others should highlight hyperlinks automatically (Case 38841)
- fixed — "Log off" from one browser should log out of all browsers
- fixed — Password reset does not reset password for admin user (Case 38825)
- fixed — Improve performance of scm output processing (Case 38864)
- fixed — Force clear of Collaborator 2.1 cookies

## v4.0.825 — March 18, 2008

- added — Fun facts 181 help
- added — Client 'admin review finish' command
- added — Users can select browser to launch (Case 36294)
- fixed — Button text in planning phase of Review Overview page (Case 36946)
- fixed — Restore links to metrics definitions topic in manual
- fixed — Refreshing action items from clients counts you as logged in (Case 38546)
- fixed — Roles without participants should not be empty in table (Case 37001)
- fixed — Confusing instructions in the defect log box (Case 37002)
- fixed — AutoDetectingReader not working properly (Case 38239)
- fixed — Initialize database button style (Cased 37080)
- fixed — Rework cookie handling to comply more completely with Tomcat's specifications

- fixed — Add more information on Perforce trigger errors (Case 38624)
- fixed — Defects showing up with comment icons in the diff view gutter. (Case 38502, Case 38651)
- fixed — Client 'addsvndiffs' command fails with Cygwin svn. (Case 38285, Case 37943)
- fixed — Client installer reports server is too old when the server is unavailable. (Case 38682)
- fixed — Client prints stack trace when adddiffs finds no diffs (Case 38547)
- fixed — Update Tomcat jar to fix LDAP authentication issues (Case 38795)

## v4.0.824 — March 6, 2008

- added — User configurable Fun Facts 181 (Case 37885)
- added — Ability to get more than 10 search results 458 (Case 38454)
- fixed — Action items need to update immediately when the tray notifier is clicked.
- fixed — Trying to auto-detect AccuRev and throwing exception instead of just skipping it. (Case 38197)
- fixed — Defects by User metrics differ from defects list report (partial fix). (Case 38443)
- fixed — Reports throw exception when filtered on invalid date (Case 38500)
- fixed — Use a different browser launching utility.
- fixed — Auto-login cookies are not cleared in IE after an upgrade from 821 (or earlier) to 823 (Case 38592)
- fixed — Various text updates for UI consistency (Case 37060)
- fixed — GUI clients should not log passwords in plain text.

## v4.0.823 — March 3, 2008

- added — Select suggested new review title 545 (Case 37152)
- added — Cache action items 333 refresh connection for 20 mins
- added — Ability to capture debug log 505 in GUI Client
- added — Upgrade Tomcat to 5.5.26
- fixed — Handling of system-administrator in LDAP configuration

- fixed — File upload description is not unicode clean (Case 36875)
- fixed — Clients ignoring proxy settings
- fixed — Change default action items refresh interval to 5 minutes
- fixed — Widget is disposed in tray notifier (Case 38262)
- fixed — Double quotes in chat turn in to quadruple quotes (Case 38408)
- fixed — Large text field does not resize automatically on "edit review" screen (Case 38405)
- fixed — Strike-through on defect text should not include "tracked external as" portion (Case 38208)
- fixed — Recognize CVS for individual files (Case 38417)
- fixed — Inserts or deletions at the end of file prevent classic view from loading (Case 37984)
- fixed — Custom fields editor shows wrong inputs when validation fails (Case 38214)
- fixed — Metrics discrepancies (Case 38443)
- fixed — Embedded database search results are case-sensitive (Case 38473)
- fixed — Defect report label is "Severity" when it should be "Type" (Case 38486)
- fixed — Reinstate user preference for disabling syntax highlighting
- fixed — Safari & Opera render <wbr> tag incorrectly (Case 38488)
- fixed — GUI Client and Tray Notifier should write to different log files (Case 38378)
- fixed — Keyword highlighting within intra-line diffs

## v4.0.821 — February 21, 2008

- added — When word-breaking long continuous words, try to break on camelCase boundaries if possible (Case 38289)
- added — SCM server information now in the <artifacts> section of "ccollab admin review-xml" (Case 38261)
- fixed — Links in chat are corrupted, both hyperlinks and bug-links (Case 38319, 38337, 38330)
- fixed — Debug review dump cannot be unzipped (Case 38036)
- fixed — Custom report fails to run when "Idle Time" column is selected under SQL Server (Case 38240)

- fixed — Script error on review overview when review is in the completed phase
- fixed — Erroneous error "version content already sent" when using diff-shim application with Subversion under Windows (Case 38225)
- fixed — Tray notifier displays erroneous error (Case 36402, 37744, 36237)
- fixed — Erroneous "<wbr>" text displayed inside URL's (Case 38195, 38215)
- fixed — Make label for "Local path" more clear in the cross-platform GUI client (Case 38066)
- fixed — Fixed incorrect comment label of "DEDT" on upgraded databases (Case 38336)
- fixed — Custom review filter is partially ignored when looking at printable format (Case 38269)
- fixed — Incorrect Subversion check-in comment parsing (Case 38285)
- fixed — Hide and encrypt password configurations in the cross-platform GUI client (Case 37830)
- fixed — Large files were sometimes not showing up in the diff viewer (Case 38235)

## v4.0.820 — February 12, 2008

- added — Support for sym-links 801 in Subversion (Case 38038 and Case 38039)
- added — Reports and views now count "number of comments" as the number of comments made by users in the context of chatting, not including system messages like "created defect" or "file uploaded". (Case 37645)
- fixed — Bug upgrading older databases to build 819 (Case 38189)

## v4.0.819 — February 7, 2008

- added — Log off link 315 on the menu bar. (Case 37923)
- added — Update button 96 on licensing screen to force update of license codes from SmartBear servers.
- added — Support for Perforce Eclipse Plugin (P4WSAD) version 2007.3.601 (Case 37916)
- added — Installer option to preserve existing database settings. 82
- added — New icons for various operations
- added — Chat icons in diff window 377 gutter show unread conversations 431 with unread conversation icon.

- fixed — Oracle strings now support more than 1023 characters
- fixed — Vault integration case sensitivity.
- fixed — After completing a review to move it to rework, a refresh moves the review back to inspecting (Case 37927)
- fixed — Review cancel action should take user back to home page. (Case 37938)
- fixed — Adding files by diff does not handle rework status correctly (Case 37902)
- fixed — Support version name /main/0 as a valid predecessor
- fixed — Reduce chat load times.
- fixed — Template selection alignment is inconsistent (Case 38004)
- fixed — Support Subversion in languages other than English (Case 37537)
- fixed — Use "last changed revision" as Subversion local version
- fixed — AccuRev failing on Linux with capital letter in path (Case 37932)
- fixed — Asynchronously load file content for better browser performance.
- fixed — Trailing context lines has one too few lines. (Case 38020)
- fixed — Do not send multiple copies of emails to the administrator (Case 37954)
- fixed — Long text on review overview makes some information flow offscreen.(Case 37129)
- fixed — Links for dumping system data are confusing (Case 37922)
- fixed — AccuRev cannot auto-detect from path (Case 38052)
- fixed — When rolled up and changelist comments repeat, save space by indicating the number of repetitions
- fixed — Misaligned diffs (Case 38026)
- fixed — Restore ability to change the login prompt
- dropped — Eclipse plugin no longer supports Eclipse 3.0 (Eclipse 3.1 or better is required)

## v4.0.818 — January 24, 2008

- added — Added usage statistics 212 sub-page to the Admin section
- added — Added documentation 496 for the GUI Client

- added — Added documentation[613] for the Tray Notifier
- added — Added documentation for ccollab login command
- added — Downloadable Eclipse plugin as zipped-up eclipse update site (Case 36063)
- fixed — Log version number in GUI Client log
- fixed — Moving the attach file dialog on the review edit screen causes the dialog to disappear. (Case 37658)
- fixed — Restore ability to edit title after review completion (Case 34178)
- fixed — Database unavailable at server startup makes Collaborator require a restart. (Case 37848)
- fixed — Add*Diffs[510] commands use of GUID's for changelist id is confusing to users (Case 37640)
- fixed — IE 7 does not like RSS feeds with DTD's in them (Case 37890)
- fixed — Overdue notice showing up on complete reviews on the action items lists (Case 37587)
- fixed — Invalid Subclipse info for svn servers with no path (hostname only) (Case 37665)

## v4.0.817 — January 18, 2008

- fixed — Incorrect diffs from Subversion (Case 37133)
- fixed — Print product version number in Eclipse log (Case 37648)
- fixed — Script errors in classic view
- fixed — User short names are duplicated if names are too similar (Case 37704)
- fixed — Client installer no longer kills running client taskbar app (Case 37314)
- fixed — Predecessor version not found if file has many version; ClearCase only (Case 37666)
- fixed — Cannot configure multiple Perforce servers (Case 37736)
- fixed — ContentViewer must havea content provider when input is set (Case 37755)
- fixed — ASCII control characters in files cause "More Lines" operation to hang (Case 37718)
- fixed — GUI client needs to allow SCM specification and local path
- fixed — addactivity subcommand should show local synced version numbers (Case 37364)

- fixed — IE 6 fails to download custom reports
- fixed — Added ccollab logiin subcommand (Cases 36731, 36729, 36730)
- fixed — Comments misaligned at the point of a code insertion (Case 37747)
- fixed — Mandatory subscriptions not uniformly enforced on the server
- fixed — Clear Case fixes for named local versions
- fixed — CVS uploads from GUI client do not work (Case 37800)
- fixed — Make action item url's work in non-root contexts
- fixed — User account creation fails from administration screens (Case 37823)
- fixed — Grace seats are not handled properly

## v4.0.814 — January 8, 2008

- added — Added commands "ccollab admin review participant assign", "ccollab admin review participant remove", and "ccollab admin review comment create"
- added — Added command "ccollab admin review defect create"
- added — Global option for SMARTBEAR_PROCESS_USER_WAIT
- added — Support clients working in ClearCase view directory (Case 37082)
- added — --creator option to "ccollab admin review create"
- added — add "ccollab admin review defect mark-external" and support defect id "last"
- added — trigger talkback[206] to set review access restriction (Case 36783)
- fixed — New user registration on login page fails (Case 37448)
- fixed — AccuRev integration predecessor algorithm should be "previous occupant" (Case 36970)
- fixed — Make skip unchanged preference work with new diff viewer (Case 36422, Case 37397)
- fixed — Intraline diff highlight expands to include entire SGML tags, making it hard to understand what actually changed. (Case 35430)
- fixed — 'ccollab set username' has poor error messaging (Case 37225)
- fixed — ' ccollab info' messaging has bad formatting (Case 37226)

- fixed — adddiffs with no third argument causes NPE (Case 37303)
- fixed — Default Perforce p4port to 1666 if not specified anywhere at all
- fixed — auto-detect ClearCase in view root directory
- fixed — Custom reports creates links too long for IE (Case 37551)
- fixed — Diff viewer title should lead with file name (but not full path), so it will show up nicely in the task bar. (Case 37547)
- fixed — Underscores occluded when intraline diffs wrap (Case 37436)
- fixed — Error in toolbar application when launching application twice (Case 37550)
- fixed — Empty Boolean type options cause NullPointerException (Case 37508)
- fixed — Diff viewer should scroll new selections somewhere toward the middle of the page. (Case 36634)
- fixed — Completed reviews should not use current date to determine if review is "overdue". (Case 37587)
- fixed — NullPointerException in CvsClientConfiguration. (Case 37574)
- fixed — read config files and ccollabgui scm config settings on a best-effort basis

## v4.0.812 — December 21, 2007

- added — Added commands "ccollab admin review create", "ccollab admin review edit", " ccollab admin review delete"
- added — Ability for Administrators to change access rules 186 on reviews in progress (Case 37190)
- added — Friendlier stalled review notifications 329 (Case 37391)
- fixed — Vestigial new review remains after client error (Cases 37393, 23390)
- fixed — ccollab addchanges 629 for AccuRev uploading files in lowercase
- fixed — Full line differences not highlighted (Case 37413)
- fixed — Track externally fails for defects in overall section of files (Case 37319)
- fixed — Side by side view has issues displaying deleted files correctly (Case 37412)
- fixed — Error installing server with Oracle or SQL Server databases

## v4.0.811 — December 14, 2007

- added — Email notifications 329 contain a footer indicating that the email was automatically generated.
- fixed — Improved performance of diff viewer for large files, especially with Unix or Macintosh line endings.
- fixed — Command line client erroneously reports ccollab addchanges works with CMVC.
- fixed — Emails going out from the system administrator rather than the default address (Case 37177)
- fixed — Perforce integration ignoring p4client or p4user if only one is set
- fixed — Better help for the command line client
- fixed — Triggers should not be required to respond with a well-formed document if they are not talking back (Case 37307)
- fixed — AccuRev integration throws NullPointerException in ccollab addstream.
- fixed — AccuRev integration displays wrong error message when not authenticated.
- fixed — Subscription fixes and access restrictions (Case 36793)
- fixed — Email notification option to disable for administrators; restoring migration files restores with notifications disabled. (Case 37285)
- fixed — Highlighted text does not honor user preference fonts (Case 37369)
- fixed — After changing the review deadline, the "Apply" button reverts the deadline to the default (Case 37382)

## v4.0.810 — December 7, 2007

- fixed — StarTeam differences support in ccollab addstdiffs command (Case 37034)
- fixed — ClearCase ccollab addactivity 688 should give empty predecessor for files new to activity (Case 37142)
- fixed — Support for ClearCase \0 versions (Case 36675)
- fixed — Error when getting SQL or CSV from custom reports in Internet Explorer (Case 37185)
- fixed — Mark external (defect) fail on review overview page (Case 37274)

- fixed — Tray notifier throws NullPointerException when double clicked upon startup (Case 36879)
- fixed — Perforce integration not including changelist number in review title
- fixed — Perforce integration not copying changelist description into overview field.
- fixed — NullPointerException in addp4diffs 779.
- fixed — Additional logging in the email notification processor.

## v4.0.809 — December 4, 2007

- added — ClearCase performance changes (Case 36819).
- added — Chat pane 379 is masked when loading
- added — Add "review completion date" to review_activity_summary 914 view (Case 37196)
- fixed — External diff launcher fails because of Windows absolute file names (Case 37175)
- fixed — Improved remember me cookie system; works with container managed role-based security.
- fixed — Custom fields copied from previous reviews gets wrong previous reviews (Case 37183)
- fixed — Installer fails to migrate from 2.1 if application is running at a non-root context path.
- fixed — Fix addversions argument parsing.
- fixed — Defect tracking integration not showing up for custom fields (Case 37147)
- fixed — AccuRev integration addstream 634 was picking incorrect previous versions in some cases (Case 36970)
- fixed — New diff viewer ignores user font preference (Case 37159)
- removed — Undocumented report command; would not work with improved security model.

## v4.0.808 — November 28, 2007

- added — Added SCM configuration options to cross platform GUI.
- fixed — ClassCastException in addstdiffs command (Case 37034)
- fixed — Manual pages were not loading properly in Eclipse

- fixed — Attach changelist fails in IE6 (Case 36932)
- fixed — Clean up font sizes in cross platform GUI for GTK systems.
- fixed — Removed spurious logging when database is known to be invalid.
- fixed — Floating license timeout is now one hour instead of four.
- fixed — Installer fails to remove old jars when updating the Eclipse client (Case 37009)
- fixed — Under certain circumstances, servers running with the embedded database could not be upgraded to 806 or 807 builds.
- fixed — Improve performance of the chat portion of the diff viewer (Case 37127)

## v4.0.807 — November 19, 2007

- added — Do not allow reviews to move to inspection phase without materials 344
- added — (AccuRev only) Added support for ccollab addardiffs 632 for uploading arbitrary AccuRev differences with full context.
- fixed — Super-search box did not handle characters outside the Latin-1 range
- fixed — Action item "Response to Comments" not posted when new changelist contains only updated files (Case 36847)
- fixed — "Compare" menu in diff viewer lists all uploads as "1st" (Case 36931)
- fixed — ccollab addsvndiffs now includes full diff context instead of just 3 lines surrounding each change
- fixed — ccollab addsvndiffs now supports binary files correctly regardless of local line ending settings
- fixed — Inconsistent table cell alignment on review overview page (Case 36945)
- fixed — Custom fields cannot be reordered (one-off special database export error case)

## v4.0.806 — November 15, 2007

- added — Defect state icons in the diff viewer 377
- added — Client commands for administering author subscriptions
- added — Client commands for administering file-based subscriptions
- added — Home Page menu option in the tray icon context menu 614 (Case 36856)

- added — Web user interface accepts and displays Unicode input (Case 36697)
- added — Add submitted changelists support to cross-platform GUI[496]
- fixed — Diff viewer should start with the first diff if user clicks on file name on review overview page (Case 36637)
- fixed — Participant list expansion does not happen on first apply (Case 36632)
- fixed — The ccollab addstream command requires being in a local workspace directory (Case 36350)
- fixed — Unable to edit defects in the Overall section (Case 36806)
- fixed — Editing defects does not set custom field drop-downs in Firefox (Case 36807)
- fixed — Under certain scenarios, comments needed to be marked read in all versions of a file before closing a review (Cases 36845, 36572)
- fixed — Printable report double-html-encoded text (Case 36877, 35540)
- fixed — Custom report "Recently Completed Reviews" did not properly filter against "completed" status
- fixed — Review title not set by default when uploading changelists with descriptive text (Case 36437)
- fixed — NullPointerException in tray notifier
- fixed — Enterprise organization field is too small (Case 36904)
- fixed — File subscriptions not working for Perforce paths (Case 36870)
- fixed — Anonymous Subversion has log entries with no author
- fixed — Archiving fails with blank screen. (Case 36973)
- fixed — Improved performance of archive queries.
- fixed — Trivial Reviews report should not contain canceled reviews (Case 36873)
- fixed — File subscription input fields too short; expanded to 255 characters (Case 36939)
- fixed — Explain to the user why the recent participants list is empty (Case 36941)
- fixed — Password overrides and ccollab set collab had confusing behavior.

## v4.0.805 — November 5, 2007

- added — AccuRev 622 only: New ccollab addstream 634 command uploads differences pending promotion given an AccuRev stream name (Case 36530)
- added — User RSS feeds 330 are now identified by a guid rather than by user name.
- added — Review completion date in review overview 348 (Case 36679)
- added — Eclipse review wizard 545 suggests a new review title.
- fixed — Eclipse synchronize view actions missing icons
- fixed — Text   appearing in file overview display (Case 36704)
- fixed — Eclipse "add to new review" and "add to existing review" should be consolidated.
- fixed — Variables not being substituted in stalled review emails (Case 36703)
- fixed — No password (unset) causes issues in Eclipse plugin and system try notifier (Case 36689)
- fixed — ccollab set collab "" does not overwrite server-url properly (Case 36725)
- fixed — Some database dump files could not be reloaded because of improperly encoded special characters (Case 36773)
- fixed — Installer does not properly add strictAuthOnly attribute to realm definition.
- fixed — 4.0 clients fail to authenticate when server is LDAP authenticated. (Case 36739)
- fixed — Do not specify password argument to P4 commands unless specifically configured to do so.
- fixed — Support spaces in subversion committed file names.

## v4.0.804 — October 29, 2007

- added — Performance improvements on Review Overview screen
- added — Handle Perforce ticket mode, automatically login for a new ticket if necessary
- added — Use Add to Review Wizard 545 in GUI Client
- fixed — Add view for custom field drop-down values (Case 36607)
- fixed — Bogus scrollbar when window shrinks in non-wrapped side by side diff (Firefox only) (Case 36209)
- fixed — Clicking on skipped lines section puts non-number in the new chat line number (Case 36561)

- fixed — With short filenames in the same directory, icon column is too wide.
- fixed — Binary file type field should be longer than 255 chars (Case 36640)
- fixed — Remove mandatory dependency on Eclipse CVS plug-in (should be an optional dependency)
- fixed — New subscription fails (Case 36656)
- fixed — Improved logging and error messaging when send email fails (Case 36641)
- fixed — Next/Previous highlights the wrong line for deletes (Case 36635)
- fixed — Support spaces in file/path names in Subversion (Case 36360)

## v4.0.803 — October 18, 2007

- added — Action items 333 on homepage refresh without a complete page refresh.
- added — Documentation on configuring LDAPS 119
- fixed — Custom reports broken when using embedded database (Case 36470)
- fixed — addactivity command fails when file paths contain spaces (Case 36463)
- fixed — NullPointerException in ccollab addchanges (Case 36507)
- fixed — Metrics by Review report only showing reviews with defects (Case 36481)
- fixed — Eclipse plugin stores passwords in plain text (Case 35694)
- fixed — actionitems command does not list action items (Case 36505)
- fixed — addchanges command should not be available when SCM is Perforce
- fixed — Oracle backend should store integers as Number(10) so external tools will recognize as Number.
- fixed — Possible fix for tray notifier crashing
- fixed — Comment promotion prevents bad comments from being marked read
- fixed — External diff viewer link should not be displayed when showing only one version (Case 36451)
- fixed — Jump to next line sets the line to -1 if no more defects
- fixed — Installer fails to preserve database credentials on upgrade (Case 36540)

## v4.0.802 — October 12, 2007

- added — New field in review_activity_summary view, author_rework_hours (Case 35473)
- fixed — Minor UI improvements on review overview page.
- fixed — Optimized syntax coloring for very long lines and large files.
- fixed — Search box now jumps directly to review if the search text matches a review id. (Case 36390)
- fixed — Selection causes text to move around in some parts of diff viewer. (Case 36210)
- fixed — Firefox refreshes the review overview page twice instead of the once required. (Case 35882)
- fixed — Tray notifier should to to "normal" mode if no more urgent action items.
- fixed — Version ordering on review overview screen was incorrect if user preference set to alphabetic.
- fixed — System tray notifier was not shutting down on upgrade.
- fixed — Upgrading a server with existing LDAP configuration results in 403 errors from the server.
- fixed — In diff viewer, selecting text results in a script error (IE only)
- fixed — When license code is invalid, the node id is not displayed, but is required in order to resolve the issue.
- fixed — Minor changes to documentation.

## v4.0.801 — October 8, 2007

* FIRST RELEASE OF v4.0 BETA *

- BIG NEW FEATURES
- added — Revamped diff viewer 377 with over-under view 377, much faster loading times, more jump/search features 376, and hide-able chat pane 429
- added — Subscriptions 327 allow users to get on a review with author- or file-based rules
- added — "Recently Completed Reviews" 333 list on home page

- added — Customized Review Report 467
- added — "New Review" 336 page is now a single page rather than a wizard, cutting down drastically on the time it takes to create and start a new review
- added — Open differences in local diff viewer 377
- added — Review deadlines 192
- added — Proactive notification system 329 alerts you to reviews that are stalled
- added — RSS feed 330 for Action Items 314
- added — Command-line structure reorganized and many new options 523 added
- added — Extensive Command-line help
- added — Tray notifier 613 for Windows and Linux
- added — Enterprise organization 226 specification for reviews
- added — Annotate files 356 in overview screen with one-line comments that are visible without opening files
- added — SCM system can usually be detected automatically

- LITTLE NEW FEATURES
- added — "Fun Facts 181" feature on home page
- added — User list drop-down 336 now support substring-based searching for names
- added — Custom reporting view defects_by_file 914
- added — Command-line client now connects to server much more quickly
- added — Addchanges from multiple SCM configurations in one command
- added — All global options can be overridden 508 on the command-line
- added — Special last keyword can be used on the command-line to refer to the review that was last created
- added — Version control system is automatically detected by the command-line client in most cases
- added — Ability to disable all metrics 186 displays (Case 25888)

- added — Ability to edit list of files before they get uploaded for ccollab addactivity 688 and ccollab addtrack (Case 35009, 35383)
- added — Home page action items 333 have more information and are split by incoming, outgoing, and more
- added — Many UI elements can now expand and contract
- added — Date input boxes now use a proper calendar widget
- added — Database diagnostics 212 help you and tech support diagnose problems
- added — Cleaned up user activity statistics 219 on the Admin/User page so it is easier to see how many licenses you really need
- added — User initials algorithm now supports "Last, First" semantics
- added — Button on New Review page lets you pick up custom field settings from the previous review
- added — Added on-wire compression for more web page elements for faster page-load times
- added — Speed improvements for the Review Overview page 345, eliminating 60% of the SQL queries
- added — Speed improvements for the Chat pane 429, making the common no-op case fast
- added — Review columns 357 are hard to differentiate with many reviewers (Case 35450)
- added — Improved defect icons 436 for red-green colorblind users (Case 35023)

- FIXES
- fixed — Syntax highlighting got confused with certain sequences of escape characters in C-style strings
- fixed — Subversion checkout at root of local disk fails to upload files properly (Case 34992, 35056)
- fixed — New files/directories caused Eclipse plug-in to fail to find other modifications (Case 35664)
- fixed — Users could re-open review by continuing to comment; now must explicitly re-open 348 the review with a button.
- fixed — When server is awaiting database upgrade, clients were reporting "incompatible server" instead of "needs database upgrade" as the error message (Case 32710)

- fixed — Incorrect handling of non-UTF-8 characters in communications with the Perforce server (Case 36367)
- fixed — Reporting database view 914 review_activity now includes all participants, not just those with non-zero activity time

## 11.2.11 Version 2

### v2.1.731 — November 29, 2007

- fixed — Fix activity changelist editing of LOCAL versions. (Case 36839)

### v2.1.730 — October 30, 2007

- added — Add loc_unversioned field to the review_version_summary containing the line count from files not under version control (Case 36613)
- fixed — Improper handling of Unicode characters in Perforce changelist text. (Case 36737)
- fixed — Subversion integration fails when spaces in file names

### v2.1.729 — October 16, 2007

- fixed — Installer can throw NullPointerException when installing server under Java 1.4 (Case 36080, 36093, 36247, 36242, 36243)
- fixed — Metrics by review report only shows reviews with defects (Case 36481)

### v2.1.728 — September 14, 2007

- added — Support for editing the list of files being uploaded for ccollab addactivity 688 (Case 35009, 35383)
- fixed — Last activity date was being updated from Windows GUI Client even if the user was not logged in, only when a non-floating license code was installed
- fixed — Perforce trigger was blocking check-in on integrate even with --ignoreintegrate is set, if additional files were added as part of the integration and therefore tagged as "branch" by Perforce (Case 35897)
- fixed — Now impossible to change login or password if LDAP/Active Directory is being used (Case 36052)

- fixed — User permissions on views were being destroyed when views were recreated in Oracle and MySQL

## v2.1.727 — August 24, 2007

- added — First release of AccuRev support 622
- added — Customizable text 183 on the login screen instructing the user which account to use (Case 35653)
- added — Recent participant list 336 should go back only 30 days so older users disappear (Case 35685)
- added — ccollab addchangelist now supports multiple changelists on the same command-line
- fixed — URL encoding error externalizing a defect when a quote character appeared in the defect text (Case 35638)
- fixed — Comment promotion error with comments on deleted files in rolled-up view (Case 35698)
- fixed — Some custom reports views were not available under Oracle when custom field titles were longer than 30 characters (Case 35420)
- fixed — User preference for "start with latest" versus "start with base" not being honored with mixture of local and committed changelists (Case 35597)
- fixed — Not properly encoding control characters (below 0x20 ASCII) in database dump XML files
- fixed — Installing server component multiple times switching databases between embedded and SQL Server causes invalid configuration (Case 35488)

## v2.1.725 — August 1, 2007

- added — Variable substitutions 161 for review and defect custom fields 256 (Cases 35552, 35556)
- added — Now only administrators can un-cancel a canceled review (Case 35502)
- fixed — Next/Prev buttons broken in the New Review Wizard 336 (regression from build 724) (Case 35491)
- fixed — Default values for defect custom fields not showing up under IE 6 in the side-by-side view chat 429 area (Case 35551)
- fixed — User could download file content from the /data servlet without login credentials (Case 35580)

- fixed — User preference 317 for whether to display "previous" or "base" version by default in side-by-side view was selecting the wrong thing with files not under version control (Case 35476)
- fixed — ccollab addchangelist 814 with Subversion 799 puts the wrong text in the title of the review if the changelist description starts with the letter "r" (Case 35452)
- fixed — ccollab addchanges 681 with ClearCase 666 invoked from different subdirectories can insert "." path components thereby making comments not promote across versions properly (Case 35575)
- fixed — Server installer was not setting the LDAP system administrator property (Case 35477)

## v2.1.724 — July 23, 2007

- added — Support for Unicode text files in the diff viewer 377
- added — Multiple conversations for binary files in side-by-side view
- added — Support for Python and Visual Basic syntax coloring in the side-by-side view (Case 35427)
- added — Support for the UltraCompare textual diff file format 512 (Case 35343)
- added — Subversion trigger for creating a review 819 should allow for setting the review title and overview text (Case 35252)
- fixed — Double-clicking the "Accept" button when there is text in the comment box can result in duplication of the comment and "Accept" text in the conversation history (Case 35384)
- fixed — ccollab addcvsdiffs 643 does not use absolute RCS server paths when a file has been added (Case 35345)
- fixed — Review title limited to 128 characters when editing review information after the review has started, whereas the limit is 255 when creating a new review (Case 35399)
- fixed — External bug URL 194 link was not honoring BUGSUBJECT or BUGID special fields (Case 35419)
- fixed — Files now sorted case-insensitive (Case 34511)

## v2.1.723 — July 13, 2007

- added — New command-line command ccollab addfiles for uploading files not under version control (Case 35268)

- added — Display the official, permanent link to a review at the top of the Review Summary 345 page (Case 35277)
- added -- Now supports Eclipse v3.3; changes include: Removed hidden popup that was making help content invisible, Removed byte order mark in Eclipse help
- fixed — When role disallows editing/deleting defects, participant cannot even edit/delete his own defects (Case 35272)
- fixed — Able to make a new conversation 429 on line 999999 (Case 35317)
- fixed — Spurious error message when uploading files using v2.1 client 506 against v2.0 server (Case 35263)
- fixed — Create-Review trigger 206 was not being executed when the command-line client created the review (Case 35128)
- fixed — Removed hidden popup in Eclipse v3.3 that was making help content invisible

## v2.1.721 — July 5, 2007

- added — ClearCase ccollab addactivity 688 should diff re-bases and integration branch with --diffintegration (Case 34382)
- added — Configurable minimum timeout for AJAX chat refresh rate 184, plus restrict per-user refresh rate so as to not swamp the server or the browser with chat-refresh requests
- fixed — Cannot continue the review due to "unread comments" when in fact all comments are read and the "unread" ones are just rolled up (Case 35163, 35104)
- fixed — Some trigger variables 161 were not being replaced correctly (Case 35128)
- fixed — Client installer 486 does not check for an existing trailing semi-colon when updating the Windows PATH environment variable (Case 34989)
- fixed — Stop logging spurious error messages when the database has no tables (Case 34890)
- fixed — ClearCase version of ccollab addversions not properly handling files with predecessor version main/0 (Case 35144)
- fixed — Cannot get past "database needs updating" screen when MySQL 60 auto-increment settings do not start at 1 (Case 35141)
- fixed — Report data broken when reporting user-time when using the embedded database (Case 35104)
- fixed — Migrating data 100 into Oracle failed when custom field title was blank

- fixed — Migrating file data from review dump put the content in the wrong place when the user overrides the location of the content cache
- fixed — Leaking database connections when "download files" command was used. Connections were reclaimed eventually but wasted resources.
- fixed — Perforce trigger truncated changelist description lines at '#' (Case 35271)

## v2.1.719 — June 18, 2007

- fixed — ccollab addchanges 813 fails for Subversion checkout at file system root (Case 35056)
- fixed — Windows command-line invocations not being parsed properly in certain special cases
- fixed — Database cleanup for exceptional case on multi-core servers where a bug in a previous build could cause a comment/defect to not be associated with a visible conversation, thereby making a review impossible to complete (Case 35074)
- fixed — Quoted strings in C-style languages could cause a line of code to be incorrectly colored in the side-by-side view if it contained escaped quotes (Case 35048)

## v2.1.717 — June 15, 2007

- added — Option to override the maximum file size for a review. (Case 32313)
- fixed — Perforce trigger 788 broken when files have been truncated due to excessive length. (Case 32313)
- fixed — Reverted files show up as "Reworked". (Case 34592)
- fixed — Files not contained in directories display incorrectly in review overview 357 screen.
- fixed — Remove spurious log messages about uploaded files not having previous version as this is the expected condition.
- fixed — Race condition prevents some pages from loading and causes others to only be partially loaded. (Case 35051)
- fixed — Admin license page 94 displays the wrong number of licenses.

## v2.1.714 — June 8, 2007

- fixed — Web page can take forever to load (or new comment takes forever to appear) when emails are being sent out due to the action just submitted (Case 34986)

- fixed — URL validation in input fields should check format but not actually attempt to connect (Case 34997)
- fixed — Perforce trigger 789 was blocking non-Collab-related changes to the changelist spec, specifically in the case of creating a new changelist from scratch (Cases 34436, 34812, 34915)
- fixed — ccollab addchanges 813 fails for Subversion checkout at file system root (Case 34992)
- fixed — Variable substitution 161 system could skip a variable if you strung together many variables in a row (Case 35002)

## v2.1.713 — June 6, 2007

- fixed — Drop-down items not displaying current values (Case 34963)

## v2.1.712 — June 5, 2007

- added — Drop-down typed custom fields 256 can now be given a default value
- added — ClearCase only: ccollab addactivity 688 now supports diffs against latest version from integration branch (Case 34382)
- fixed — Cannot add yourself 352 to a review if you are not already a participant or an administrator (Case 34834)
- fixed — Custom field description text not being displayed in the "create defect 436" form (Cases 34671, 34861)
- fixed — Binary file identification pattern 197 should not be case-sensitive (Case 34895)

## v2.1.711 — May 30, 2007

- added — Subversion server-side hook for automatically creating a review 819 whenever a change gets submitted
- added — Subversion server-side hook for uploading revision data 819 to the server after a change gets submitted
- added — Manual page describing when notifications 329 are sent to users
- added — More documentation and troubleshooting information about Perforce server-side triggers 788
- fixed — Database connection errors or NullPointerException errors in a race condition when sending email notifications

- fixed — Error running ccollab addsvndiffs 816 with Subversion 799 clients prior to v1.4.0.
- fixed — Scrollbar missing in chat 429 window for binary files (Case 34623)
- fixed — Do not display "0 changelists" under "local changes" when all changelists are in fact checked in (Case 34798)
- fixed — Broken links to manual pages on client tools when attaching materials 340 to a review
- fixed — Supplying a blank name for a role configuration causes problems in the administration GUI

## v2.1.709 — May 21, 2007

- added — Speed enhancements for the diff viewer 377 web view.
- added — ClearCase ccollab addactivity 688 support for diffs against integration branch (Case 34382)
- added — Subversion server-side hook 819 for ensuring that files are reviewed before they are checked in.
- added — User's Guide chapter for recommended hardware 158 and how to increase server speed 158.
- fixed — CVS variant of ccollab addversions uploaded correct file content but incorrectly marked the versions as "added" in the web GUI (Case 34617)
- fixed — Subversion command-line can get too long under Windows with large numbers of files.
- fixed — SQL error trying to archive files using Oracle, SQL Server, or embedded database back ends.
- fixed — Subversion Eclipse plug-in for Eclipse v3.1 would fail to upload files if uncontrolled files were left in a project and the Subclipse preference "Select unversioned resources on commit" was enabled.

## v2.1.708 — May 16, 2007

- added — New Perforce trigger 789 for automatically updating changelist description with information about the associated review.
- added — Support for using wildcards with paths when using ccollab addchanges with SCM system type none under the Windows cmd shell

- added — Support for Subversion's[799] svn:externals feature wherein a repository contains a soft link to another repository.
- added — Speed enhancements for the Review Summary[345] page when viewing very large reviews.
- added — Speed enhancements for common operations in the Subversion integration.
- added — "Obfuscated Dump[174]" for both review-specific and whole-system database dumps to allow customers to send SmartBear tech support data dumps without sending any sensitive information.
- fixed — (Eclipse/Subversion plug-in[563]) Changes from different resources but in the same repository were showing up in different changelists (Case 34544)
- fixed — The command-line client[506] could get confused when svn switch was used on a local check-out
- fixed — Eclipse Subversion plug-in can pick up the wrong repository configuration (Case 34399)
- fixed — Suppressed spurious (and harmless, but annoying) error messages we were printing to the Subclipse console
- fixed — URL's inside review comments can get mangled if they are very long (Case 34461)

## v2.1.707 — May 7, 2007

- added — Optimized subversion integration.
- fixed — Side by side not resizable when wrap lines turned off.
- fixed — Subversion integration did not properly handle added file in added directory.
- fixed — Recent participants list should only include participants from the selected workflow.
- fixed — System administrator account matched incorrectly when LDAP[119] configured.
- fixed — Content archive fails when using embedded database.

## v2.1.706 — May 1, 2007

* FIRST NON-BETA RELEASE OF v2.1 *

- added — Documentation for SSL configuration[111].

- added — Author should get email notification when (a) Author is required to finish the review and (b) everyone else is finished, even if those other users are not themselves required to finish the review.
- added — Passwords entered on the command-line now echo asterisks (Java v1.6 only) (Case 34477)
- added — Eclipse Subversion integration 563 can now add files to a review from the Synchronized view (Case 34442)
- added — Tutorial box about how reports 467 are cached and how to update them.
- fixed — ccollab adddiffs failing to recognize a custom diff format (Case 34465)
- fixed — ccollab addactivity 688 adds wrong predecessor when multiple versions in activity are not predecessors of each other (Case 32382)
- fixed — Subversion integration can upload HEAD version instead of HAVE in certain cases (Case 33456)
- fixed — Subversion integration threw exception when an uncontrolled file was added to a review (Case 34427)
- fixed — Subversion integration uploading incorrect file content when multiple parent directories, some not direct parents, are involved in the same changelist (Case 34328)
- fixed — Perforce integration throws exception if changelist state changes (Case 34308)
- fixed — Keyboard shortcuts 376 in the web user interface stopped working when the content view changed
- fixed — Switching roles on two participants in a review did not reset the states of those users, which could result in reviews that cannot be closed and confusing Action Item messages
- fixed — Search box 458 should trim leading and trailing whitespace from text -- handles common case when pasting text from a web browser
- fixed — In brand new installs, if the license server cannot be reached users get a blank page after login.

## v2.1.705 — April 24, 2007

- added — Command line client needs to pick up Perforce changelist comments as title for new reviews.
- added — Command line client uses Subversion changelist comment as title for new reviews.
- added — Command line client uses Clear Case activity titles as review title for new reviews.

- fixed — Subversion exception adding an uncontrolled file to a review (Case 34427)
- fixed — Eclipse; Subversion arrows not appearing on select files page. (Case 34451)
- fixed — Reverted files do not show up properly if using different clients; Eclipse and command line.
- fixed — Invoking Windows commands with embedded spaces fails. (Case 34445)
- fixed — Keyboard shortcuts mask well known Windows shortcuts. (Case 34460)
- fixed — Custom fields can get into a state where changing a single custom field changes multiple fields (Case 33700)

## v2.1.704 — April 19, 2007

- added — Perforce trigger 788 to check that a review has been created, but not care if that review has been finished.
- added — Reinvite preference expanded to mean "works independently".
- fixed — Perforce GUI integrations 786 fail to find p4 after upgrading to 2.1.
- fixed — When migrating 99 from SQL Server 64 or Oracle 67 to MySQL 60, installer fails to properly configure user authentication. (Case 34353)
- fixed — Time-in-review database view updated to ignore time spent during "rework" phase.
- fixed — Specifying multiple files to addchanges fails for Subversion (Case 34354)
- fixed — Subversion added files showing up as modified (Case 34337)
- fixed — Search results should not return reviews that the current user does not have access to.
- fixed — Case sensitivity problem in Perforce trigger (Case 34288)
- fixed — Side by side version titles not appropriate for checked in changelists (Case 33422)
- fixed — Eclipse plugin subversion integration 563 does not properly set the base revision (Case 34108)
- fixed — Eclipse plugin subversion integration 563 does not handle spaces in filenames correctly (Case 34008)
- fixed — Subversion integration not finding reverted files (Case 34400)
- fixed — Uploading new changes to a review changes the state, but not the phase of the review.

- fixed — Action Item message for authors is misleading if the author is required to complete the review.
- fixed — External defects should be drawn with strike through and italic style (Case 34023)
- fixed — Externalized defect system should link to configured create defect URL (Case 34021)
- fixed — Remove usernames from Subversion URL's (Case 33425)
- fixed — Where possible, derive the author from the Subversion URL (Case 33806)
- fixed — Remove usernames from Subversion server config (Case 33807)
- fixed — Action item text did not always make sense for users whose activity is required to initiate phase change.
- fixed — Authors who marked reviews finished had no action item at all while waiting for other participants.
- fixed — Hide-show previous changelists must be sticky (Case 33888)

## v2.1.703 — April 11, 2007

- added — New role-specific option to have participants not re-invited to a review when another user makes a comment
- added — New role-specific option to control whether this role is allowed to change review details 348 while the review is active (Case 34302)
- added — Allowing edit of the review details 348 after the review is complete, but only when you are a participant or admin (Case 31478)
- added — Special warning when a user is viewing a review but is not a participant in that review (Case 34026)
- added — Display the title of the current workflow in the Review Overview 348 section (Case 33972)
- added — Now supporting either ccollab set scm perforce or ccollab set scm p4 to configure 777 ccollab Perforce settings.
- fixed — Long comment text does not wrap in chat viewer. (Case 34289)
- fixed — CMVC integration fails when not all fields are populated
- fixed — Subversion was not properly handling file changes when parent directories were moved or deleted

- fixed — Subversion was not properly handling file changes when the parent path was the result of a directory-move, as with the first check-in on a new branch
- fixed — Some non-ASCII characters not displaying correctly when entered in the AJAX chat component 429 in side-by-side view (Case 34050)
- fixed — HTML characters in the review title cases Review Overview 345 page to display incorrectly (Case 34281)
- fixed — In ClearCase 666, ccollab addactivity 688 picks wrong latest version when not checked out (Case 33771)
- fixed — Email notifying a user that they have been added to a review did not include their role (Case 34165)
- fixed — Reinstalling 486 command-line client sets SCM configuration to Perforce when Perforce is installed, even if the user specifically switched to a different SCM system (Case 33741)
- fixed — Confusing error message (UnknownHostException) printed to the console instead of a useful message (Case 33858)

## v2.1.702 — April 3, 2007

- fixed — NullPointerException getting defect custom field causes chat to stop updating (Case 33947)
- fixed — Externalized defects 442 should be drawn with a line through the text (Case 34023)
- fixed — Externalized defects should not hide other menu items (Case 34024)
- fixed — In-chat defect log on Review Summary 356 page does not show custom field 256 values (Case 34019)
- fixed — Deleting a defect from the Review Summary page does not confirm with dialog (Case 34018)
- fixed — Pending Perforce 764 changelist was not considered pending when P4CLIENT not specified
- fixed — Removed tutorial boxes in create review wizard when user preferences demand it
- fixed — CMVC upload failed when a track had no parts

## v2.1.700 — April 02, 2007

- added — Eclipse 563 support for Subversion (both pre-commit and post-commit)
- added — Eclipse 553 support for ClearCase (pre-commit)
- added — SSL support 111 on web server and all clients
- added — Oracle 67 back-end database support
- added — Externalized defects 442 feature
- added — Keyboard shortcuts 376 in side-by-side view
- added — CMVC integration (Case 33617)
- added — Special alert when a pop-up blocker prevents one of our web-application windows from opening (Case 33041)
- added — Added filenames to the "Defect List" report 467 (Case 33228)
- added — Full Java example file to the Java Client Library documentation
- added — Support for importing diffs 510 with inconsistent line endings (Case 33427)
- added — Support Perforce branch-add state (Case 26437)
- added — Command-line 506 now prints the review ID when a new one is created (Case 33344)
- added — Option to disable "system dump 100" link for non-administrators
- added — Command-line 516 option --nobrowser
- added — New review wizard now displays the review title on all screens (Case 33750)
- added — Smarter algorithm for filenames with very long paths in side-by-side view (Case 33490)
- fixed — do not include deleted files in the ZIP file 358 of review files.
- fixed — Underscores being clipped in word-wrapped view on certain browsers on certain platforms (Case 33699)
- fixed — Web GUI not allowing more than 128 characters for review title, but 255 characters are actually allowed (Case 33900)
- fixed — Deleted files fail to show deleted content (Case 33468)
- fixed — Perforce integration 764 was not honoring $P4CONFIG or the p4 set -s variables under Windows
- fixed — Notification when removed from review should come from user, not admin (Case 33354)

- fixed — Newly created defect custom fields show up as review custom fields (Case 33966)
- fixed — Proper default mono-space fonts for platforms lacking the "Courier New" font
- fixed — Perforce files opened for integrate should use head version as previous, not the have version (Case 33554)
- fixed — Comment and Defect icons not showing up properly when changelists were rolled up (Case 33621, 33676)
- fixed — Improved search / error-handling for external command-line tools
- fixed — Subversion upload was failing sometimes when a deleted file was in the changelist (Case 33689)
- fixed — SQL Server was not escaping characters correctly in certain LIKE queries
- fixed — p4collab 786 now contains a usage statement if you run it from the command-line
- fixed — Subversion integration 799 can now find svn.bat and other non-standard command-line clients
- fixed — Perforce commit prevention trigger 788 did not honor the "canceled 348" state for reviews.
- fixed — Custom field reporting views 914 can take so long to complete that the query times out (Case 33446)
- fixed — When review is deleted, activity records should be deleted as well (Case 33496)
- fixed — When uploading files from Subversion, ignore both _svn and the more common .svn.
- fixed — Cancel AJAX web requests when window closes; was causing browser to run out of connections (only 2 are ever allocated) and then hang (Case 33557)
- fixed — Web GUI URL auto-formatter is too greedy (Case 33636, 33907)
- fixed — Perforce addchanges trigger 788 can fail with no error message (33584)
- fixed — Error parsing ClearCase output when backslashes were used (33224)
- fixed — Viewing XML files sometimes shows SGML character entities instead of proper characters (Case 33280)
- fixed — Could throw NPE when Java's mailcap file is corrupt
- fixed — Could get database error when inserting too much text into a field
- fixed — No notifications when you cancel the review from the review planning phase (Case 33369)

- fixed — When you have exceeded your license[92] the system now allows the existing users to continue using the system and shuts out only the additional users, rather than shutting out everyone all together.

## v2.0.621 — April 9, 2007

- fixed — Allow editing of review data in complete phase. (Case 31478)

## v2.0.620 — April 2, 2007

- fixed — NullPointerException getting defect custom field causes chat to stop updating (Case 33947)

## v2.0.619 — March 26, 2007

- fixed — Web pages refer to external resources (akamai.com, yimg.com) (Case 33192, 33876, 33885)

## v2.0.618 — March 13, 2007

- fixed — ccollab addchanges new .[813] fails for Subversion[799] local changes (Case 33784)

## v2.0.617 — March 9, 2007

- fixed — Allow install as non root user on Unix systems. (Case 33622)
- fixed — Running ccollab addp4diffs ...[779] hangs on binary files. (Case 33581)
- fixed — Perforce trigger[788] never lets ktext files be checked in. (Case 33696)
- fixed — Running ccollab addchangelist ...[814] fails when there is a deleted file in the Subversion revision (Case 33689)
- fixed — Overall chat was missing unread flags. (Case 33633)

## v2.0.612 — March 1, 2007

- fixed — Perforce addchanges[788] trigger fails with no error message (Case 33584)

## v2.0.611 — February 28, 2007

- fixed — Running ccollab set scmconfig p4 ... had no effect on Perforce integration.
- fixed — P4V integration 786 invoked Windows executable.
- fixed — Do not include deleted files in download (Case 33537)
- fixed — Null Pointer Exception when uploading file diffs (Case 33553)
- fixed — CVS diff parsing (addcvsdiffs 643) can fail if files have mixed line endings (Case 33427)

## v2.0.609 — February 21, 2007

- fixed — Uploading raw diffs 510 of binary files from Subversion could result in line-ending transformations
- fixed — Uploading raw diffs 510 with mixed line endings styles now works even if the diff generator program was not aware of the line-ending problem (Case 29801)
- fixed — Can get "Illegal Operation" exception trying to upload file content when only line-endings have changed (Case 33533, 33306)
- fixed — Subversion integration 799 fails to upload changes, either local or checked-in, when directory operations were involved (Case 33097)
- fixed — Subversion integration 799 fails to upload atomic changelists when one of the files is deleted
- fixed — In ClearCase 666, ccollab addactivity 688 was failing on directories (Case 33511, 33520)
- fixed — Perforce 764 branched adds not being added to the review (Case 26437, 32517)
- fixed — ClearCase 666 parsing of \main\LATEST can fail, and the selection of which version is "latest" was wrong in certain cases
- fixed — The reviewcustom and defectcustom database view 914 definitions were so inefficient that with a complex custom field 256 configuration the database can time out waiting for the query to complete (Case 33446)
- fixed — When a review is deleted, activity records 908 in the database are not deleted as well, leading to orphaned rows. This did not affect the operation of the software but it could be confusing when making custom reports (Case 33496).

## v2.0.608 — February 9, 2007

- fixed — Perforce trigger 788 fails to ignore canceled reviews (Case 33402)
- fixed — On Windows, Subversion integration can fail to find svn client

## v2.0.606 — January 19, 2007

- added — Many more database views 914 for external reports 908
- added — Review overview 345 screen should not refresh automatically when the user is adding a comment or defect. The auto-refresh can cause the form fields to be erased (Case 25572)
- fixed — Server no longer requires external license server for user login 315
- fixed — Removing 352 unavailable reviewers should close the review (Case 32302)
- fixed — Overall chat 356 needs unread comment flags (Case 31491)
- fixed — Uncontrolled files prevent Review Display 377 from being shown
- fixed — Javascript error on Review Display 377 (Case 32202)
- fixed — User font preferences 317 ignored under certain browsers/platforms (Case 27991)
- fixed — When you submit Subversion change-set by ID with ccollab addchangelist 814, files that were deleted in the changelist are not uploaded at all. This does work when uploading files not yet checked in (Cases 29961, 30500, 33097)

## v2.0.602 — January 11, 2007

- added — Full support for ccollab adddiffs 510 and addcvsdiffs/addsvndiffs/addp4diffs
- added — Full support for ccollab addversions for all SCM systems, not just ClearCase
- added — (Eclipse Plug-in) Pre-submit file uploading for Subversion now supported
- added — Option to disable changelist roll-up (Case 31485)
- fixed — Hide/show changelists not working when multiple SCM's in the same review
- fixed — Subversion executable not found causing command-line to fail (Case 32520)

- fixed — Subversion directory-add and directory-delete was causing the command-line client to fail to upload files
- fixed — Subversion protocol on svn://localhost URL's is now supported properly
- fixed — Author chat is automatically marked "read" if they upload a new file version (Case 32702)
- fixed — Reports failing to execute under Microsoft SQL Server for lack of proper identifier escaping
- fixed — Installer incorrectly configuring context.xml for internal-based user authentication when running against Microsoft SQL Server
- fixed — Diff-caching algorithm was showing correct but inconsistent line-diff metrics depending on the order the files were uploaded
- fixed — Context-sensitive help was not able to jump down to a sub-section of a single page of the manual
- fixed — Horizontal scrollbar in side-by-side view under IE 7 with "no-wrap" enabled was sometimes not long enough to view the entire line (Case 30481)
- fixed — If server starts up before the database, the server will never recover (Case 13626)

## v2.0.601 — December 29, 2006

- added — new AJAX-based chat system really works like instant messaging (Case 24895)
- added — Microsoft SQL Server support (with migration path from MySQL)
- added — overall-review comment/defect chat area (Case 13622)
- added — server-side event-based trigger system for running custom scripts (Case 26886)
- added — link to download all review files to the local workstation (Case 24891)
- added — ability to cancel a review rather than deleting completely (Case 24892)
- added — ability to change participants while review is going on (Case 24894)
- added — database view for person-hours per review, user, and role
- added — command-line client options for overriding server URL, username, and password
- added — emails now include user's full name both in "to" and "from" fields (Case 23212)

- added — date/time of comments now displayed as a tool-tip (Case 10644)
- added — support for "local-mode" CVS servers (Case 25020)
- added — links to server/review debugging data from "System" screen
- added — user option for whether default side-by-side diff is "current vs. last upload" or "current vs. base version" (Case 25812)
- added — P4V/P4Win plug-in should list current reviews when attaching changelists to existing reviews (Case 28458)
- added — embedded database installer option for easy test servers (with migration path to MySQL or SQL Server)
- added — installer allows admin to set web server port number
- added — can create users with name and email address from the user administration page
- added — user administration page now shows which users logged in, inactive, disabled, and administrators
- added — deleted file content now viewable in side-by-side
- added — for SCM systems without atomic changelists, consolidate all changelists into a single list view
- added — for SCM systems without atomic changelists, rework uploads include reverted files
- fixed — insertion/deletion markers cause confusion; tooltips added to explain (Case 30393)
- fixed — filename different depending on which browser attached the external file (Case 25329)
- fixed — "accept" markers are now cleared on files that have been newly uploaded (Case 20534)
- fixed — with review in inspection phase, one reviewer is done but others are not, no action item for the first reviewer (Case 25823)
- fixed — recent participant list included inactive users (Case 24141)
- fixed — user initials can be ambiguous (Case 24274)
- fixed — speed optimizations for very large reviews (Case 29003)

## 11.2.12 Version 1

### v1.2.516 — November 2, 2006

- fixed — Perforce trigger option for specifying a profile directory.
- fixed — Improve Perforce integration and trigger verbose logging.
- fixed — Installer removing attributes from realm declaration.

## v1.2.515 — October 26, 2006

- fixed — diff widget broken in Firefox 2.0 (Case 27312).
- fixed — not all files are rolled up into single changelist (Case 26821).
- fixed — preferences can fail to load client configuration (Case 27311).

## v1.2.512 — October 23, 2006

- fixed — attach uploaded changelists from the web ui (Case 26811).
- fixed — consolidated changelist view does not show all files (Case 26821).
- fixed — content archiving administration screen "hangs" with very large data caches (Case 26228).

## v1.2.510 — October 4, 2006

- fixed — Unable to mark conversations "read" when over 2000 comments present in the review (Case 25865).
- fixed — database connection leak.

## v1.2.508 — September 15, 2006

- added — options to only allow participants to view reviews/review content. (Case 24501)
- fixed — space used to mark insertion point was confusing. (Case 24246)
- fixed — whitespace inserted/removed at end of line causes rendered code to differ from real code. (Case 24767)
- fixed — syntax highlighting sometimes dropped spaces. (Case 25040)

## v1.2.507 — September 11, 2006

- added — automatically separates changed files from separate SCM systems
- added — administrative option to hide/show the option to display multiple changelists as a single "changelist"
- added — hide/show previous changes is sticky.
- fixed — clients were not honoring proxy settings taken from preferences
- fixed — increased maximum length of email message to 64k. (Case 24832)
- fixed — confusion of unique changelist ID's when using more than one Perforce server against a single Collaborator database
- fixed — when displaying changelists in "single" mode, balls up files from different SCM systems instead of making one package for each SCM system

## v1.2.506 — August 24, 2006

- fixed — Changing mark reviews as fixed workflow setting had no effect.

## v1.2.505 — August 23, 2006

- fixed — Perforce changelists picking up wrong predecessor version (Case 24157).

## v1.2.504 — August 22, 2006

- added — Verbose logging for Perforce integration.
- added — Support for Subversion username/password supplied on command-line. (Case 24124)
- fixed — Downloading binary files in Internet Explorer broken. (Case 23671)
- fixed — Client installer failed to authenticate when changing server and user password. (Case 23535)
- fixed — Removed unnecessary AJP connector from server.xml file due to small known memory leak. (Case 23736)

## v1.2.503 — August 16, 2006

- fixed — More graceful handling of license server errors using cached license codes.
- fixed — Include documentation of how to configure non-proxied hosts.
- fixed — Increased number of characters allowed in certain meta-data fields.

## v1.2.502 — August 8, 2006

- added — URL's, email addresses, and issue ID's are now hyper-linked when displaying custom review/defect fields
- fixed — uploading local changes from CVS or ClearCase could pick up the wrong previous version
- fixed — Diff highlight expansion caused code to be not printed. (Case 23379)

## v1.2.501 — August 1, 2006

- fixed — Database error causes trigger to fail.

## v1.2.500 — July 28, 2006

- added — Administration screen for archiving old review contents.
- added — Option to add/remove/edit review and defect custom fields.
- added — Reviewers have the option of annotating the review materials before the review begins.
- added — Option to change notification email subject prefix
- added — Quick links on review creation wizard to add the current user to a review
- added — clients pick up default global configuration from $HOME/.smartbear and $CWD/.smartbear, and do not create a .smartbear directory until preferences are actually saved.
- added — Perforce configuration value of "[none]" instructs the command-line utility to ignore the value completely
- added — ability to upload arbitrary diffs from CVS server
- added — line numbers in defect log
- added — selecting diff preferences causes immediate page refresh; do not have to click "Submit"

- added — file names now bold in changelist summary display
- added — option to show multiple changelists as a single unit
- added — user configurable email notification levels
- added — optional "create new user" form on login page
- added — three database views holding custom fields for reviews and defects, and extra user preference data
- fixed — First leading space is not displayed in side-by-side view (Case 10032)
- fixed — Normalize CVS paths from command-line and Windows client uploads (Case 11009)
- fixed — Status icons on review summary should link to the file (Case 10742)
- fixed — Accepted status cleared for newly uploaded versions (Case 10111)
- fixed — Chat notification icons do not update until side-by-side reloads
- fixed — Chat notification icons do not display when displaying single files
- fixed — Version selection and headings scroll off screen in side-by-side (Case 11213)
- fixed — Perforce GUI plug-in allows empty changelists to be uploaded to new reviews
- fixed — Only administrators should be allowed to change user logins (Case 8270)
- fixed — Complete line changes should be displayed as a delete followed by an add (Cases 10913, 10915)
- fixed — Review List report links linked to the wrong location (Case 11832)
- fixed — Word, Excel, and PDF documents show up as garbled text in side-by-side.
- fixed — Metrics by Review report failed to report defects per person-hour on small sample sizes.
- fixed — Metrics by Defect Type report failed to report opened per person-hour on small sample sizes.
- fixed — Next/Prev change buttons are disabled when there are no next/prev diffs.
- fixed — Clicking line of code jumps chat to correct area (Case 13280)
- fixed — Support filenames with adddiffs option (Case 13528, 14949, 16619)
- fixed — Diffs too greedy (Case 14152)
- fixed — No content uploaded for uncontrolled files (Case 14959)

- fixed — Perforce trigger has more explicit error messages when rejecting checkins. (Case 18398)
- fixed — Administrators always need access to review creation wizard. (Case 18399)
- fixed — Exported reports fail to open directly in Internet Explorer. (Case 13524)

## v1.1.442 — July 17, 2006

- fixed — Perforce trigger option for ignoring integration changelists (Case 14745).
- fixed — Perforce trigger added --verbose option for verbose logging (Case 16732).
- fixed — Perforce trigger sometimes compares wrong content (ensurecontentreviewed) (Case 16696).
- fixed — Perforce GUI integrations now include --verbose option for verbose logging.
- fixed — Perforce integration runs out of ports when large changelist present.
- added — Support bug system hyperlink in review Title
- fixed — Administrator account password update failed (Case 18606).

## v1.1.436 — May 22, 2006

- fixed — XML-RPC encoding broken on z/OS (Case 9367).
- fixed — NPE executing Clearcase cleartool (or any command line client executable) on Linux (Case 12781)

## v1.1.435 — May 17, 2006

- fixed — Perforce trigger should not run as submitting user (Case 13161).

## v1.1.433 — May 11, 2006

- fixed — Perforce usernames should not be considered case-sensitive.
- fixed — When using LDAP authentication, clients could not log in.
- fixed — Perforce trigger ensure content reviewed incorrectly rejects very large files.

- fixed — P4V/P4Win integration picks up environment variable for P4CLIENT.
- fixed — P4V/P4Win integration does not reject invalid changelists when creating new review.

## v1.1.429 — May 1, 2006

- fixed — Improve Perforce trigger error messages.

## v1.1.428 — April 26, 2006

- fixed — Perforce trigger ensure content reviewed misses additions/deletions from end of file (Case 11304).

## v1.1.427 — April 25, 2006

- fixed — Clear Case command line should only gather files from current view (Case 11799).

## v1.1.426 — April 21, 2006

- fixed — System administrator account lost administrator privileges under LDAP authentication (Case 11468).
- fixed — Command-line client not parsing Unix paths -- Clear Case integration (Case 11501).

## v1.1.425 — April 20, 2006

- fixed — Restored legacy ReviewList report.

## v1.1.424 — April 10, 2006

- fixed — Command line picked up wrong predecessor version when using ClearCase SCM (Cases 9040, 10738).
- fixed — Entering license code fails to update license.
- fixed — Client unable to locate cleartool on Windows.
- fixed — Reports not working on some headless Unix servers.

## v1.1.423 — April 5, 2006

- fixed — unattended installations preserve all configuration information.
- fixed — notifications include product name.
- changed — enabled unattended installations.

## v1.1.422 — April 4, 2006

- changed — disabled unattended installations until issues can be resolved.

## v1.1.421 — April 3, 2006

- fixed — author comments and trivial reviewer activity (for example, accepting) were triggering erroneous emails to the author to come into the review
- fixed — when multiple already-submitted Perforce changelists with common files were added to a single review, the older ones were being hidden in "Previous Uploads" when they should be displayed along with the rest.
- fixed — user passwords no longer stored in the clear
- fixed — deleting defect severities or types causes UI glitches when old defects are displayed (Case 10268)
- fixed — side-by-side font wrong in some places (Case 10607)
- fixed — participants are not notified or re-invited to reviews when the author uploads a new set of changelists
- fixed — side-by-side windows should have the filename in their title
- added — support for stronger Perforce commit-trigger that checks whether the file list and file contents match between changelist and the review to make sure the developer did not change anything between review-time and commit-time.
- added — user option for disabling syntax coloring in side-by-side view for faster content downloads
- added — option to create new review from command line client when running 'addchanges' or 'addchangelist'
- added — ability to diff previous file uploads in side-by-side view even when those uploads belong to different, discontinuous, already-committed changelists

- added — fixed defects are now indicated with a "green bug" to distinguish from still-open defects in the Review Overview screen
- added — in Review Overview screen, defects are now shown in the column of the user that first reported it rather than in a separate column
- added — added audit messages when a defect is marked fixed or open
- added — verbose option for command line clients to create debug logs
- added — more specific error message when attempting to upload an empty Perforce changelist
- added — display entire file path in defect log when the defect is linked to a particular file
- added — defect icon tooltips should include defect ID's for quick-reference
- added — command line clients prompt for password if not specified on command line so password is not in command history
- added — command line 'adddiffs' subcommand for reviewing differences between two locally accessible directories
- added — command line quiet option to suppress opening the text editor to edit file list
- added — cache control headers to allow caching of images, stylesheets, and JavaScript.

## v1.0.410 — March 20, 2006

- fixed — cannot close review with multiple reviewers when defects are entered directly into the defect log (Case 10116)
- fixed — proper error message inside P4V on changelists when uploading changelists that are not part of the current workspace
- fixed — Javascript error loading Defect Severities or Defect Types page

## v1.0.409 — March 16, 2006

- fixed — `ccollab syncusers` was failing when the Perforce user login differed from the Collaborator login only by case

## v1.0.408 — March 14, 2006

- fixed — notifications were no longer prefixed with Collaborator

## v1.0.407 — March 14, 2006

- fixed — incompatibility with some versions of MySQL
- fixed — duplicate user logins possible when changing user login
- fixed — email notifications stop being sent after some time

## v1.0.406 — March 9, 2006

- fixed — memory leak with GUI client connections
- fixed — installer not setting VM heap size
- fixed — Case 9744 - NullPointerException in P4Win/P4V integration
- fixed — Case 9710 - All action items were showing urgent icon
- added — additioal logging in database code
- added — System Dump includes more VM information

## v1.0.402— March 2, 2006

- fixed — logout broken for some versions of Internet Explorer
- fixed — some clients unable to login
- fixed — logging now enabled by default at INFO level

## v1.0.400 — February 28, 2006

- added — chat conversations are now marked as "has chatted" and/or "as opened defect" and/or "has accepted" rather than the "undecided" and other confusing concepts
- added — new reporting subsystem with additional filters and export options
- added — added option to show "only uploaded version" even when other diffs are available
- added — better support for binary file uploads
- added — ability to make a user "inactive" (you can never delete users because they are needed for reports and to display old reviews)

- added — Action Items list now more specific about the exact state of the review as it relates to the viewing user
- added — e-mail notifications now more specific about the exact state of the review as it relates to the viewing user
- added — ability to jump from a defect in the review summary defect log directly into that point in the latest source code upload
- added — support for local temporary license code if the network is unavailable for on-line licensing
- added — new customer-accessible Java library to read/write everything in the Collaborator server remotely
- fixed — cannot "Complete Review" when there are unread comments on previously-uploaded changelists that have already been marked read in the currently-uploaded changelists
- fixed — when chat is carried forward from older uploads, chat icons on review summary are not being displayed
- fixed — cvs rlog command reports error with certain versions of the CVS client
- fixed — cannot delete a review when database is in a certain rare state
- fixed — "out of memory" errors for certain operations
- fixed — database connection leak with certain types of authentication
- fixed — correct diff but erroneous intra-line highlighting when unchanged text is symmetrical on either side of change
- fixed — selection highligher highlights wrong line of code when window is narrow

## v1.0.361 — February 9, 2006

- added — (Case 8439) support for more types of CVS repository specifications.

## v1.0.360 — February 8, 2006

- added — (Case 8941) support for trial license codes which do not contact license server.

## v1.0.359 — January 27, 2006

- added — (Case 8486) when licensing server cannot be reached, retry on backup port.

### v1.0.358 — January 24, 2006

- added — (Case 8486) support for HTTP proxy authentication when validating license codes.

### v1.0.357 — January 20, 2006

- added — (Case 8420) command line client support for uploading Subversion revisions.
- fixed — (Case 8439) in certain cases, command line client was failing to upload CVS changes.

### v1.0.356 — January 9, 2006

- fixed — (Case 8319) some client upgrades failing with authentication issues
- fixed — (Case 8344) improved error message when server failed to contact license server.

### v1.0.355 — January 3, 2006

- fixed — clients were not properly authenticating when using LDAP

### v1.0.354 — December 29, 2005

- added — option to allow system administrator to participate in reviews
- fixed — (Case 7560) unable to upload files after clean install with old database.
- fixed — LDAP role fields were confusing and not required for basic LDAP support.

### v1.0.353 — December 21, 2005

- added — progress indicator for when loading file contents in file viewer
- fixed — (Case 7530) server failed to get user initials in some cases

### v1.0.352 — December 19, 2005

- fixed — (Case 7501) Stale cookies were causing issues with login/logout.

## v1.0.351 — December 12, 2005

- added — (Case 6640) previously-uploaded changelists are now hidden by default to avoid confusion when verifying fixes
- added — (Case 6812) link to leap from defect into associated file/line
- added — (Case 6871) support more than one file open at a time, bringing forward side-by-side window if already open
- added — (Case 6824) automatically hyperlink URL's and e-mails in changelist comments, chat comments, defect text, and review overviews
- added — quick-report of recent changelists and unreviewed changelists when attaching materials to a review
- added — application level log system
- added — support for LDAP authentication
- added — new installer prompts for key system parameters; server configuration split between server.xml and context.xml
- fixed — (Case 7226) code displayed in word-wrapped mode now does not have extraneous whitespace inserting into long tokens
- fixed — (Case 6932) not able to upload any changelist if Perforce default changelist was not empty
- fixed — (Case 6949) downloaded reports were not sending a filename
- fixed — added special error message when uploading a pending Perforce changelist that belongs to the current P4USER but to a different P4CLIENT
- fixed — added java.library.path to the local application scan directory for systems without environment variables
- fixed — uploading additional Perforce changelist did not update the time-of-upload; was taking the time-in-changelist instead
- fixed — uploading older Perforce changelist was sometimes destroying metrics data for previous file versions
- fixed — not able to upload submitted Perforce changelists from the command-line when P4CLIENT environment variable is changed
- fixed — after following a link but not being authenticated, after authentication you go to the home page instead of the linked page

## v1.0.349 — October 25, 2005

- added — (Case 6595,6804) side-by-side difference viewer needs "ignore whitespace" and "ignore capitalization" options
- added — support for client-side HTTP proxies
- added — link to download server logs in "System" debugging page
- fixed — (Case 6780) Perforce server-side trigger should ignore changelists consisting of branched paths only for purposes of both review and automatic uploading
- fixed — (Case 6814) error installing client under Windows when user-level PATH variable does not already exist

## v1.0.347 — October 14, 2005

- added — "mark all comments read" quick-button in side-by-side view; administrators can disable this feature
- fixed — unable to "mark read" on new chat if already marked read in the same chat session
- fixed — issue regular expression was case-sensitive

## v1.0.346 — October 10, 2005

- added — new Action Item when e-mail or full name not currently set for a user
- added — prompt to download local client software from the server
- added — administrative contact information is now required

## v1.0.345 — October 3, 2005

- added — (Case 6664) ability to view files/metrics for uploaded changelists in "Attach Materials" section of the New Review Wizard
- added — (Case 6652) Action Items with no "new" comments should be marked as such on the "Action Items" list, and should be low priority
- added — (Case 6666) refresh button for chat window
- fixed — (Case 6635) URL's in e-mails give error when opened under Eudora

- fixed — (Case 6661,6643) JavaScript error with next/previous buttons in side-by-side view
- fixed — entering non-digits in the "jump to review" menubar field results in strange error message
- fixed — file paths with backslashes not word-wrapping on Review Overview screen

## 11.2.13 Version 0/Alpha

### v0.9.344 — September 26, 2005

- added — initial support for ClearCase
- added — initial support for the Windows GUI Client
- fixed — (Case 6621) "add defect" comment field still limited to 255 characters in form on Review Overview page
- fixed — (Case 6475) "edit defect" command should update GUI widgets to indicate "edit" rather than "create"

### v0.9.343 — September 21, 2005

- added — command-line support for uploading CVS changes by label
- fixed — (Case 6593) error uploading Perforce files: "unrecognized chunk"
- fixed — (Case 6540) bogus date in user admin screen for users who have never logged in
- fixed — (Case 6475) "edit defect" should have different form text than "create defect".

### v0.9.342 — September 15, 2005

- fixed — (Case 6544) script error viewing diffs in certain files, or when file content is not yet fully loaded
- fixed — action items for "waiting for comments" should appear after "respond to comments" because they are lower priority
- fixed — syntax highlighting split by intra-line difference causes incorrect colors in both difference and syntax display

### v0.9.341 — September 14, 2005

- added — (Case 6471) external base server URL (used with e-mail notifications) is now configurable by the administrator
- added — (Case 6408) participant-picker needs recently-used list to speed up the assignment process
- fixed — (Case 6508) 255 char limit in overview
- fixed — (Case 6529) error attempting to send email with a certain JVM
- fixed — (Case 6534,6538) typing comment then clicking "Accept" causes comment to be lost
- fixed — (Case 6522) ccollab addchanges cannot handle relative paths

## v0.9.339 — September 9, 2005

- added — (Case 6267) workflow revamp: faster to "accept," concepts of "reject" and "defect" have been fused, no more accept/unsure/reject determination for commentary, comments carry forward to code verification step, defects shown in side-by-side next to comments for editing and marking fixed
- added — (Case 6372) large text fields should expand as more text is entered into them
- added — (Case 6406) administrative option to set system-wide tab width for source file display
- fixed — (Case 6406) tabs not being displayed properly in side-by-side view
- fixed — (Case 6355) should not be allowed to add a changelist to a review that is completed or that you are not a participant in or a creator of
- fixed — (Case 6265) comment text should not be limited to 255 characters
- fixed — (Case 6382) client error message not helpful when incorrect server/port is specified, especially when a real service is present
- fixed — (Case 6354) clicking on a specific command-line does not open the file to that line

## v0.6.337 — August 24, 2005

- fixed — (Case 6275,6279,6281) database errors accessing user data
- fixed — (Case 5624) error not able to locate content-cache subdirectories on the server

## v0.6.336 — August 23, 2005

- added — initial support for CVS integration
- added — (Case 5628) user-preference to set number of lines of context displayed in side-by-side view
- fixed — error accessing reviews when review meta-data is missing from the database
- fixed — (Case 6252) exception adding comments to the comment-list

## v0.6.335 — August 19, 2005

- added — (Case 5551) launch browser window after P4Win plug-in attaches changelist
- fixed — (Case 5528) error uploading large file
- fixed — (Case 5615) incorrect LOC-changed metrics for file uploads
- fixed — (Case 5617) extra whitespace in side-by-side view of code when tabs are used in a certain way
- fixed — (Case 5618) file download should supply filename to the browser for better open/save handling on the browser end

## v0.6.334 — August 18, 2005

- added — (Case 5553) ability to alter review overview information during any phase of the review (without reverting back to "Planning" mode)
- added — (Case 5612) reviews that just entered the Inspection phase (either because just created or because fixes are being verified) should be displayed as "Waiting for comments" on the author side until a comment is actually made; the author should be notified by e-mail when this comment is made.
- fixed — not properly HTML-escaping certain characters in review titles
- fixed — (Case 5600) not always receiving e-mails as review in a new code review or when fixes have been uploaded
- fixed — (Case 5613) exception when accessing admin screen when upgrading from a certain database version

## v0.6.332 — August 17, 2005

- added — (Case 5590) ability to download file versions directly from the side-by-side view

- added — (Case 5576) optional feature to send e-mails from the review creator's e-mail address rather than the system default from address
- added — (Case 5387) gutter icons in side-by-side show which lines have associated chat comments
- fixed — (Case 5528) error uploading large file
- fixed — (Case 5577) uploaded "previous version" is actually the HEAD revision instead of the last-synched revision
- fixed — (Case 5584) erroneous error message while uploading files from a Perforce branch
- fixed — (Case 5386) skip-lines algorithm in side-by-side view should treat chat comments as significant lines, not just changes.
- fixed — (Case 5597,5598) incorrectly-encoded SGML character entities when intra-line diff splits on an encoded character

## v0.6.331 — August 14, 2005

- added — user preference for flipping the side-by-side view to older-on-right instead of the default older-on-left
- added — (Case 4388, 5279, 5424) ability to compare follow-up change against other uploaded changes in addition to the original SCM base version
- added — (Case 5335) participant input list in Review Creation Wizard automatically adds additional rows for large number of participants
- added — (Case 5544) better e-mail headers including easily-filterable text and the name and ID of the related review
- added — (Case 5468) drill-down from Review List report to actual review view
- added — (Case 5426) new report: Review List with Perforce Changelists
- added — (Case 5567) option to disable "issued fixed" throughout the application
- fixed — (Case 5550) lines with trailing whitespace are showing up with false-positive differences
- fixed — (Case 5484) errors on changelist-upload when no bug system regular expression is given by the administrator
- fixed — (Case 5515) new defects opened on completed reviews should change the review phase to Phase II: Review

- fixed — (Case 5557) erroneous exception when uploading file data under a special condition

## v0.6.330 — August 10, 2005

- added — global configuration options to ease per-user installation
- fixed — PATH variable not being set as environment-expand registry value under Windows
- fixed — (Case 5478,5482) uploading certain Perforce changelists takes exceedingly long time
- fixed — (Case 5481, 5516, ...) erroneous ccollab error requiring review-id and changelist-id when files are in the default changelist
- fixed — (Case 5524) better error messages when configuration files are not accessible
- fixed — (Case 5508, 5525) better error message when `http://` is missing from collaborator server specification

## v0.6.327 — August 8, 2005

- added — (Case 5426) ability to jump to review by ID from the menubar
- added — (Case 5471) command-line should pick up Perforce configuration from environment variables
- fixed — (Case 5453, 5454) exception working with changelist from uploaded file
- fixed — (Case 5460) exception viewing certain review detail reports
- fixed — (Case 5476) javascript error auto-refreshing review page under certain IE version
- fixed — (Case 5479) word-wrapping is breaking inside SGML character entities on side-by-side source code view

## v0.6.325 — August 7, 2005

- Case 5428 — added — support for PHP-style review logging data for older upgrade paths
- Case xxxx — added — report data can now be retrieved from the command-line in any format, with optional filters
- Case xxxx — fixed — PHP-upgrade path fixes

## v0.6.323 — August 5, 2005

- Case 5366 — added — now collecting time-spent-in-review metrics per user, per role, per review, per phase, per defect count/type/severity
- Case xxxx — added — report data can now be exported in XML or CSV format
- Case xxxx — added — report data can now be retrieved from the command-line in any format, with optional filters
- Case 5393 — fixed — administrator should not be allowed to change login name
- Case 5393 — fixed — user preference title has wrong username in title when administrator edits other users' information

## v0.6.322 — August 1, 2005

- Case 5381 — added — configurable review participant role behavior to support different workflows
- Case xxxx — added — ability to delete reviews when they are still in planning phase

## v0.6.321 — July 29, 2005

- Case 5278 — added — P4Win/P4V integration for creating a new review using a changelist
- Case xxxx — added — client installer now picks up Perforce configuration automatically (can still be overridden from the command-line)
- Case xxxx — added — client installer now prompts for and verifies server connectivity settings (can still be overridden from the command-line)
- Case xxxx — added — new "Licensing" administration page
- Case xxxx — added — administrative settings: minimum allowable build numbers for command-line and windows clients to force users to upgrade

## v0.6.320 — July 26, 2005

- Case 5278 — added — P4Win/P4V integration for uploading changelist data and associating it with an existing review
- Case 5342 — fixed — cannot switch participant roles in Review Planning

- Case 5344 — fixed — new review created; next/prev buttons work but links on the wizard pages list on left do not work
- Case 5345 — added — server system parameters on the System debugging link
- Case xxxx — fixed — spurious error when uploading already-committed changelists when the changelist has already been uploaded

## v0.6.319 — July 25, 2005

- Case xxxx — fixed — error sending e-mails with older Java mailer
- Case xxxx — fixed — associating changelist with review not properly encoding certain string before inserting into database
- Case xxxx — fixed — error-handling, validation, and help text for bug-tracking integration items
- Case xxxx — added — synchronizing Perforce userlist with Collaborator from the command-line

## v0.6.317 — July 25, 2005

- Case 5041 — added — uses value of REMOTE_USER for automatic log-in when password is blank
- Case xxxx — fixed — error accessing new user preference item

## v0.6.316 — July 21, 2005

- Case 4614 — fixed — new wizard prompts user better about pre-checkin changes
- Case xxxx — fixed — error where "mark as read" while also changing comment status does not get reflected on review summary page
- Case 5280 — added — user preferences for side-by-side word-wrapping, and font family and size
- Case xxxx — added — many more Web Service API's

## v0.6.314 — July 15, 2005

MAJOR UPGRADE! This new alpha release sees the unveiling of the new Collaborator platform. We have switched from our proof-of-concept PHP platform to an industrial-strength Java platform with greatly enhanced scalability, support for multiple databases, and an improved, simplified workflow.

## v0.4.215 — June 14, 2005

- Case 4614 — fixed — new wizard prompts user better about pre-checkin changes
- Case 4607 — added — test PHP configuration at install-time
- Case ???? — added — optimizations for database communication cutting page-load time in half

## v0.4.213 — June 01, 2005

- Case 4562 — fixed — resizing window causes JavaScript error (IE only)
- Case 4564 — fixed — maximizing window causes JavaScript error (IE only)
- Case 4567 — fixed — scroll position is no longer remembered on screen refresh (IE only)
- Case 4563 — fixed — word-wrap still wraps sometimes (IE only)

## v0.4.212 — May 30, 2005

- Case ???? — fixed — all HTML and CSS now validated HTML 4.0.1-Transitional
- Case 4144 — fixed — focus change in accept/reject/undecided combo-box does not move selection rectangle
- Case 4126 — fixed — user's file-diff preferences not persisted (that is, lines of context, ignoring case)
- Case ???? — fixed — line-selection rectangle obscures underscore characters in certain browsers
- Case ???? — fixed — word-wrapping algorithm in file-diff too pessimitic, especially under IE
- Case ???? — fixed — tightened up intra-line difference highlighting
- Case ???? — added — optional new line-comparison frame makes it easier to compare long lines
- Case 4126 — added — option to disable word-wrapping in file-diff file content

- Case 4126 — added — buttons to advance to the next/previous change
- Case 3300 — added — review-create phase is now a "wizard"
- Case 3644 — added — on-line licensing system
- Case 4316 — added — new report on number of comments made per review / per user
- Case ???? — added — ability to review whole files and file-differences uploaded independantly from version control

## v0.4.211 — April 21, 2005

- Case 3858 — fixed — sometimes vertical scrollbar not appearing in file-diff view
- Case 3858 — added — side-by-side divider between chat and file content is now move-able so you can easily "hide" comments while reading a file
- Case 3949 — added — ability to jump to a particular review by ID from the menubar

## v0.4.210 — April 20, 2005

- Case 3929 — fixed — disallow running the command-line client against the Code Collab demo server (the default configuration)
- Case 3930 — fixed — javascript error when planning a review and the review is not yet ready to proceed to the next stage
- Case 3920 — added — ability to delete changelist associations from reviews
- Case ???? — added — "revert" button on various forms

## v0.4.209 — April 19, 2005

- Case 3905 — fixed — pending changelists should not be available for adding from the website
- Case 3677 — added — "Save" buttons on forms remember scroll position for easier data-entry on long pages
- Case 3721 — added — administrative usage monitor available from bottom of admin "User" page
- Case 3838 — added — ability to set minimum build number for the command-line client from the Collab server admin settings page

- Case 3842 — added — issues mentioned in changelists should be associated with the review automatically
- Case ???? — added — reports can now "group by" certain things (see "Assignments" and "Reviews" for examples)

## v0.4.208 — April 12, 2005

- Case 3687 — fixed — files in "add" state in Perforce changelist not being picked up by the command-line client
- Case 3563 — added — user administration page: add users manually; view info and usage; admins can set "admin" flag and other user preference informatiion for other users
- Case 3702 — added — customization administration page: change text for defect severities, defect types, and phase titles and prompts
- Case 3694 — added — "Remember Me" should redirect the login page to the user's review home page.
- Case 3697 — added — report enhancements: review-created date on report and summary page; complete user comment table on review summary page
- Case 2623 — added — "Preview Changes" on changelist during Review Planning phase
- Case 3664 — added — margin icon for comments is now a pencil instead of a meaningless arrow
- Case 3681 — added — help documentation on the purpose and behavior of user roles
- Case 3699 — added — phase prompt bullets are hidden when tutorial mode is disabled

## v0.4.207 — April 11, 2005

- Case 3617 — fixed — local changes not showing up in side-by-side view
- Case 3606 — added — review phase data to getReports(), so scripts can check for review-completion, not just review-existance
- Case 3614 — added — "Action Items" list on the home page; daily e-mail action items reminder (if any are present)
- Case 3616 — fixed — observers should not prevent a review from proceeding to the next phase
- Case 3645 — added — removed "controller" role

- Case 3618 — added — ability to move from Phase II back to Phase I
- Case 3622 — fixed — prevent moving from "Planning" to "Reviewing" when there are unsaved changes to the Participant List
- Case 3623 — added — support "Remember Me" log-in option to make logging in easier
- Case 3660 — added — adding changelist to the review should automatically add the changelist author to the review in the "Author" role unless that user is already added in a different capacity
- Case 3662 — fixed — commenting on new line of code with "Accept" without additional commentary did not work
- Case 3665 — fixed — inconsistent wording: "neutral" vs "undecided"
- Case 3666 — fixed — review in Phase II should show comments from all participants, not "consensus". Cannot really show "undecided" or "unreviewed," only what comments are made
- Case 3676 — added — usernames always get initials; smarter algorithm for pulling out initials automatically

## v0.4.206 — April 8, 2005

- Case 3471 — added — all users imported from Perforce, not just those in recent version history

## v0.4.205 — April 8, 2005

- Case 3606 — added — ability to call arbitrary RPC methods from the command-line
- Case ???? — added — help documentation on XML/HTTP/RPC integration points

## v0.4.203 — April 5, 2005

- Case 3557 — fixed — author needs notification when review enters "Rework" or "Completed" phase
- Case 3558 — fixed — should not be able to close a review when "new" comments are pending
- Case 3559 — fixed — creating new users from the front page results in accounts that cannot be logged in to

- Case 3560 — fixed — install should check for writable /file-cache directory; instructions should say to check this also

## v0.4.202 — March 29, 2005

- Case 3474 — fixed — error message when incorrect file data is shown
- Case 3485 — fixed — add-defect form had incorrect maximum limit on number of lines in the file
- Case 3486 — fixed — Perforce not loading previous file versions (reload version control data using "Clear Version Control Data" link in install.php)
- Case 3470 — added — can now delete existing review from the bottom of the "Review Summary" tab
- Case 3472 — added — passwords in database are now encrypted

## v0.4.201 — March 28, 2005

- Case ???? — fixed — not compatible with MySQL v3.23

## v0.4.198 — March 24, 2005

- Case 3316 — fixed — ccollab synchscm documentation is confusing
- Case 3396 — fixed — deletions not handled properly in Perforce
- Case 3317 — added — option to disable "create new user" from front page
- Case 3367 — added — debugging support in server and command-line utility
- Case 3369 — added — added support for content-compression for browsers that support it

## v0.4.186

*First alpha release*

# 11.3 Appendix C: Java VM Options

## Files That Store Java Virtual Machine Options

You can find the options for the Java Virtual Machine in the `.vmoptions` files that are located in Collaborator's install directory. The file name corresponds to the name of the appropriate executable file.

For **server** the path will be: `<collab server install dir>/ccollab-server.vmoptions`

For **command-line client** the path will be: `<collab client install dir>/ccollab.vmoptions`

For **GUI client** the path will be: `<collab client install dir>/ccollabgui.vmoptions`

> **Note:** You must restart the Collaborator server or client before these settings take effect.

This topic describes some of the most frequently used options.

## File Format

To specify Java system properties that will be available to Collaborator, use this syntax:

```
-Dvariable_name=value
```

**Important:** the last line in the `.vmoptions` files must be empty, that is, you must enter a new line after the last line with data.

## Collaborator Server Properties

| Property Name | Purpose and Usage |
|---|---|
| `com.smartbear.ccollab.binary.converter.threads` | The maximum number of threads allocated to convert binary documents into images for review. This value must be a positive integer. Default is 4. |
| `com.smartbear.ccollab.license.noperiodicupdate` | If set to a non-blank value, this property prevents the Collaborator server from checking for license updates. |

| Property Name | Purpose and Usage |
|---|---|
| com.smartbear.collab.datamodel.activity.update.interval | The number of seconds to wait for user activity requests in order to update user's time in review. Collaborator server queries for review activity every 15 seconds and updates the time counter if any of these requests were successful during time interval specified by this VM option. Default value is 60 seconds. Minimal allowed value is 16 seconds (values lesser then 16 seconds will be increased to 16 seconds automatically). To disable time tracking of user activity, set this VM option to 0. |
| com.smartbear.ccollab.datamodel.drift.disable | A Boolean setting (values: true/false) to disable in-product chat 33 for trial users. |
| com.smartbear.ccollab.notification.max.retry.interval | The number of milliseconds to wait between attempts to connect to the SMTP server. |
| com.smartbear.server.email.max.num.per.execution | The maximum number of e-mails sent during a single check up for stalled reviews and reviews approaching deadline. Default is 100. |
| com.smartbear.server.sso.disable | A Boolean setting (values: true/false) to disable SAML or Crowd single sign-on authentication 130. |
| com.smartbear.server.disable.comments | A Boolean setting (values: true/false) to disable comments in review materials. If enabled, only defects can be added. Default is false. |
| com.smartbear.database.longquerythreshold | The time in milliseconds to allow database queries to run before logging a warning message. This is used to help debug bottlenecks and characterize the behavior of complex queries such as those used by the custom reports system. Default is 2000 milliseconds. |
| com.smartbear.collab.datamodel.remotesystem.cache.size | In order to decrease number of calls to repository hosting servers, Collaborator caches some of the most often retrieved entities from APIs (commits, pull request diffs, commit diffs). This setting specifies the number of entries to keep in cache for each of the remote systems. Default is 20000. To disable caching, set this option to 0. |

| Property Name | Purpose and Usage |
|---|---|
| com.smartbear.collab.datamodel.remotesystem.api.client.retry.seconds | The time in seconds to wait before trying to re-connect when the remote repository server is not accessible. Can have values from 1 to 30. Default is 1.<br><br>If the remote repository server is still not accessible, Collaborator will retry to connect in increasing time intervals until the remote server responds or until the maximal number of attempts is reached. Wait time is increased with each attempt: wait_time*1, wait_time*2, wait_time*3 and so on.<br><br>If the server did not respond after all attempts, Collaborator will mark the respective webhook as inactive and put an exception to remoteSystem.log 169 |
| com.smartbear.collab.datamodel.remotesystem.api.client.retry.attempts | The maximal number of attempts to re-connect when the remote repository server is not accessible. Can have values from 1 to 5. Default is 3. |
| com.smartbear.collab.datamodel.remotesystemitem.update.interval | The time in milliseconds to wait between updating remote system item statuses. For those remote systems which do not support webhooks. Default is 120000 milliseconds. |
| com.smartbear.collab.datamodel.remotesystem.issuemeta.update.interval | The time in milliseconds to wait for issue creation meta info from the remote issue-tracker. Default is 60000 milliseconds. |
| com.smartbear.collab.datamodel.remotesystem.bitbucketserver.api.delay | The time in milliseconds to wait between pull request update and a call to Bitbucket server API. Default is 5000 milliseconds. |
| com.smartbear.collab.datamodel.remotesystem.socket.timeout | Specifies the timeout (in milliseconds) of the remote system connection when updating remote system item statuses. Default is 3000 milliseconds. |
| com.smartbear.collab.database.search.review | Specifies the preselected search scopes for reviews in the Oracle database. Can be empty (no preselected scopes), or can contain any combination of the following comma-separated values: |

| Property Name | Purpose and Usage |
| --- | --- |
| | • `id` - Search for substring matches among review identifiers.<br>• `title` - Search for substring matches among review titles.<br>• `creator` - Search for substring matches among review creators.<br>• `reviewCustomFields` - Search for substring matches in review custom fields.<br>• `checklistCustomFields` - Search for substring matches in checklist custom fields.<br>• `participantCustomFields` - Search for substring matches in participant custom fields. |
| `com.smartbear.collab.database.search.changelist` | Specifies the preselected search scopes for changelists in the Oracle database. Can be empty (no preselected scopes), or can contain any combination of the following comma-separated values:<br>• `comment` - Search for substring matches in changelist comments.<br>• `author` - Search for substring matches among changelist authors.<br>• `scmIdentifier` - Search for substring matches among changelist identifiers. |
| `com.smartbear.collab.database.search.file` | Specifies the preselected search scopes for files in the Oracle database. Can be empty (no preselected scopes), or can contain the `fileName` value to preselect a search scope for substring matches in filenames. |
| `com.smartbear.collab.database.search.defect` | Specifies the preselected search scopes for defects in the Oracle database. Can be empty (no preselected scopes), or can contain any combination of the following comma-separated values: |

| Property Name | Purpose and Usage |
|---|---|
| | • `text` - Search for substring matches in defect descriptions.<br>• `customFields` - Search for substring matches in defect custom fields. |
| `com.smartbear.collab.database.search.comment` | Specifies the preselected search scopes for comments in the Oracle database. Can be empty (no preselected scopes), or can contain the `conversations` value to preselect a search scope for substring matches in review comments. |
| `com.smartbear.collab.enable_statistics` | Specifies whether Collaborator server should collect and send usage statistics to SmartBear. To learn about our privacy policy, visit https://smartbear.com/privacy/. |
| `com.smartbear.diff.cache.maxentries.memory` | The number of entries allowed in the diff cache. The size of any given entry can vary significantly, but this tuning parameter provides a rough mechanism for tuning the cache size. Default is 2000. |
| `com.smartbear.diff.image.cache` | A Boolean setting (values: true/false) to enable caching of previously opened documents in order to display them quickly when you reopen them. Default is true.<br><br>Applies to word processing documents, PDFs, presentations and vector graphics. |
| `com.smartbear.diff.image.conversion.framework` | Switches between Aspose.Total and Apache PDFBox frameworks for converting new documents into images. Possible values are: ASPOSE and APACHE. Default is APACHE.<br><br>Applies when converting word processing documents, PDFs, presentations and vector graphics. |
| `com.smartbear.diff.image.resolution.scale` | The resolution scale of converting documents into images displayed in DiffViewer. Default is 162. |

| Property Name | Purpose and Usage |
|---|---|
| | We do not recommend changing this option unless DiffViewer has troubles displaying small characters, diagrams or documents look blurry. In this case, to get clear images, you will need to increase the resolution scale, restart Collaborator server and clear browser cache.<br><br>Applies when converting word processing documents, PDFs, presentations and vector graphics. |
| `com.smartbear.lines.cache.size` | The maximum size of the parse lines cache. Values should be a memory size in kilobytes or megabytes indicated by the case-insensitive suffixes "k" and "m" respectively. For example, 20 megabytes could be indicated by "20m" or "20M". Default value is 5 megabytes. |
| `com.smartbear.web.debug.max.log.entries` | The maximum number of entries in the client log file which is enabled by "CAPTURE DEBUGGING LOG". The default value is 160000 which results in a log file that is approximately 50MB. |
| `smartbear.appstate.period` | The delay, in milliseconds, between runs of the application state recorder. The state recorder periodically writes information about the state of the application into the database for future trend analysis. Default is 15 minutes. |
| `smartbear.appstate.startup.delay` | The delay, in milliseconds, before the application state recorder starts. Delaying the application state recorder allows the application initialization process to complete before anything is written to the database. Default is 5 minutes. |
| `smartbear.chat.request.limit` | The maximum number of chat update requests allowed per minute. This value will be used for active reviews and the clients and servers will automatically scale back the request rate as activity on the review declines. |
| `smartbear.news.disable` | A Boolean setting (values: true/false) to hide the Collaborator News panel in the WebUI. |

| Property Name | Purpose and Usage |
|---|---|
| flash.enabled | Specifies whether Web Clients will use Adobe Flash component to upload files. By default, the option is disabled and Web Clients use more secure HTML5 component for uploads. |
| smartbear.internal.p4DisableAuthorCheck | A Boolean setting (values: true/false) that specifies whether to disable author check-up of Perforce changelists. If set to true, users could upload changelists of other authors. |
| smartbear.mail.smtp.connectiontimeout | The timeout, in milliseconds, for SMTP connections. The value must be a positive integer. |
| smartbear.mail.smtp.timeout | The timeout, in milliseconds, for SMTP requests. The value must be a positive integer. |
| smartbear.reports.ignore.accepted.comments | A Boolean setting (values: true/false) to indicate whether or not accepted comments should be ignored in the reporting of review statistics in the Review Detailed Report. |
| smartbear.trigger.timeout.seconds | Applies to the triggers on the Admin \| Triggers 206 page. Specifies the maximum allowable time for the trigger execution in seconds. If the execution time exceeds this value, Collaborator stops the trigger and logs an exception. 0 means there is no limit set. |
| smartbear.userauth.class | Specifies the authentication adapter used to authenticate users. Do not set this property unless directed by SmartBear technical support. |
| com.smartbear.review.workflow.phase.rework.notification | Specifies if the reviewers in "waiting" state will receive notifications of other participant/author activity during the Rework phase. If this option is not specified or is true, reviewers will get notifications, otherwise they will not. |
| com.smartbear.reviewpools.lock.regex | Prevents review pools from being deleted. If a review pool group name matches the specified regular expression, this group cannot be deleted. |
| com.smartbear.reviewpools.lock.allowrolechange | A Boolean setting to allow the roles of the locked review pools to be changed by anyone. |

| Property Name | Purpose and Usage |
|---|---|
| | This setting has effect only if the `com.smartbear.reviewpools.lock.regex` setting is specified. |
| `com.smartbear.reviewpools.lock.rolesallowedtomodify` | A comma separated list of system role names that are allowed to modify locked review pools and their roles. Administrators are always allowed to modify the locked review pools and their roles.<br><br>This setting has effect only if the `com.smartbear.reviewpools.lock.regex` setting is specified. |
| `javax.net.ssl.trustStore` | A generic Java machine option that specifies the location of the Java keystore file containing the collection of CA certificates trusted by this application process (trust store).<br><br>For example: `-Djavax.net.ssl.trustStore=${ProgramFiles(x86)}\collaborator server\tomcat\conf\collab.ks`<br><br>This option is NOT recommended. To set trust store location for Collaborator tomcat server it is better to use `truststoreFile` attribute the `Connector` element of the <Collaborator Server>\tomcat\conf\server.xml configuration file. |
| `com.smartbear.code.metrics.style` | Specifies whether to include comments and whitespaces when calculating line of code (LOC) metrics. If set to `SLOC`, comments and whitespaces will be ignored in metric calculations. |
| `com.smartbear.ccollab.datamodel.manager.chunk.for.32.jdk` | If set to a non-blank value, this property indicates that Collaborator server is running on 32-bit Java virtual machine. This indication is needed for accurate creation of dump file[100]. |
| `com.smartbear.collab.datamodel.remotesystem.webhooks.ssl.enable` | A Boolean setting (values: true/false) that specifies whether webhooks created by the Easy Add Repository would have SSL verification enabled or disabled. |

| Property Name | Purpose and Usage |
|---|---|
| | By default, SSL verification of webhooks is disabled to avoid handshake issues on Collaborator servers with self-signed or untrusted certificate authority (CA) root certificates. |
| `com.smartbear.web.diffviewer.zoom.scale.min` | Specifies the minimal zoom level (in percentage) of images, Word and PDF files displayed in Diff Viewer. Default is 10. |
| `com.smartbear.web.diffviewer.zoom.scale.max` | Specifies the maximal zoom level (in percentage) of images, Word and PDF files displayed in Diff Viewer. Default is 500. |
| `com.smartbear.web.diffviewer.zoom.scale.increments` | Specifies the increment of zoom level (in percentage) of images, Word and PDF files displayed in Diff Viewer. Default is 10. |
| `com.smartbear.database.disable_oracle_regexp_like` | A Boolean setting (values: true/false) that specifies which predicate (REGEXP_LIKE or LIKE) Collaborator will use to query the data from Oracle database.<br><br>By default, when this option is not specified or when the option is set to false, Collaborator uses REGEXP_LIKE predicate in queries.<br><br>When the option is set to true, Collaborator will use a faster LIKE predicate in queries. We recommend to use this option if you encounter performance issues on Oracle database. |

## Client Properties

The following properties are specific to Collaborator clients:

| Property Name | Purpose and Usage |
|---|---|
| `smartbear.ccollab.upload.truncate.size` | The size, in bytes, to truncate uploaded files in the changelist. This threshold prevents clients from inadvertently overloading the server with files too large to process. The minimum threshold value is 4MB. |

| Property Name | Purpose and Usage |
| --- | --- |
| | If any of your clients uses this option, avoid using the ensure-diffs-reviewed[795] and ensure-content-reviewed[794] triggers. In this case, these two triggers will compare only the remaining parts of the truncated files. Consider using the ensure-reviewed trigger instead. |
| smartbear.ccollab.upload.ignore.binary.file | A Boolean setting (values: true/false) to indicate whether or not certain binary files (specified by the Binary File Types[197] setting) are ignored when uploading changelists |
| com.smartbear.ptc.forceprojectpath | A Boolean setting (values: true/false) to use project path instead of configpath if your PTC projects[752] have variant sub-projects. |

## Java Memory Settings

There are two aspects that affect the Java Virtual Machine regarding memory management and that may affect application performance. One is the maximum heap size and the other is the configuration of the garbage collector (GC).

- The default maximum heap size that is available for Collaborator client applications depends on the Java Runtime Environment (JRE) version being used: 64 MB, if you have JRE version 5.0 and earlier, or 1GB or one quarter of physical memory (what is the smaller) for later versions unless your computer has only one processor or it has less than 2 GB of RAM (in which case 64 MB is used).

  By default, the maximum heap size for the Collaborator Server is set to 1 GB. You can increase the maximum heap size by using the `-Xmx` switch. For example, to set the maximum heap to 2 GB, use the following line:

  ```
  -Xmx2G
  ```

  You can use the letters `G`, `M` and `K` to specify gigabytes, megabytes and kilobytes. You can specify only whole numbers. So, for example, instead of using `-Xmx1.5G`, you have to use `-Xmx1536M`. Note that some Java virtual machines do not support the `-X` option.

  Increasing the maximum heap size can help you if you got the errors like `java.lang.OutOfMemoryError: Java heap space errors` in your logs.

  We would recommend that you increase the maximum heap size to 2 GB or even more, depending on the typical size of documents that your users upload to the server.

- In addition to increasing the heap size, you will also likely need to change the garbage collector's settings, especially if you see the "`java.lang.OutOfMemoryError: GC overhead limit`" error in in your server logs, or experience severe performance degradation.

  To change the garbage collector settings for the Collaborator Server, add the following line to the `ccollab-server.vmoptions` file:

  ```
  -XX:+UseConcMarkSweepGC
  ```

  If you are using Java 7 Update 45 or later, use the following line instead:

  ```
  -XX:+UseG1GC
  ```

  We do not recommend applying these settings, if your computer has 2 or fewer processors (cores). If you are running into performance issues when performing reviews, you should consider running the Collaborator Server in a system that has at least 4 processors (cores). Changing these garbage collector settings will improve your server performance, especially of you also use the `-Xmx` option to increase the maximum heap size.

  Note that the operating system can report that your java process consumes more memory than the maximum heap size you specified. This happens because other parts of the Java runtime, for example, PermGen, consume memory as well. You can try using the `-XX` parameters to tune these parts. See Oracle HotSpot VM documentation for complete information. Note that quite often the `-X/-XX` options are JVM-specific and Oracle can change them without notice. We recommend that you consult documentation of your Java Virtual Machine.

## Server Temporary Directory

By default, the server will use the *tomcat/temp* subdirectory under the Collaborator Server installation directory as the temporary directory. To use an alternate location, specify the Java temporary directory variable in `ccollab-server.vmoptions`, for example:

```
-Djava.io.tmpdir=/path/to/temp/directory
```

Similarly (though less likely to be needed), this can be done for the Collaborator clients in the corresponding `.vmoptions` files.

## Network Proxy Settings

To configure server proxies, use the `ccollab-server.vmoptions` file. See Network Configuration 89 for information on available settings.

## Network Connection Debugging

To debug LDAP or SSL connection problems as well as other network related issues, add the following line to the `ccollab-server.vmoptions` file:

```
-Djavax.net.debug=all
```

After restart, Collaborator will create an additional log file, `output.log`, on the server computer with trace information on the issue.

## 11.4 Appendix D: Java Compatibility Matrix

The following tables describe which versions of Java environment are supported by different versions of Collaborator.

### Collaborator Server

| | Oracle JDK 7 | Oracle JDK 8 | Oracle JDK 9 | Open JDK 11 / Oracle JDK 11 |
|---|---|---|---|---|
| Collaborator Server 12.0 and later | No | Yes (built with this version) | Yes | Yes |
| Collaborator Server 11.3-11.5.x | No | Yes (built with this version) | Yes | No |
| Collaborator Server 9.x-11.2 | Yes (built with this version) | Yes | No | No |

### Collaborator Client

| | Oracle JDK 7 | Oracle JDK 8 | Oracle JDK 9 | Open JDK 11 / Oracle JDK 11 |
|---|---|---|---|---|
| Collaborator Client 12.0 and later | No | Yes (built with this version) | Yes | Yes |
| Collaborator Client 11.3 - 11.5.x | No | Yes (built with this version) | Yes | No |
| Collaborator Client 9.x-11.2 | Yes (built with this version) | Yes | No | No |

## Eclipse Plug-in

| | Oracle JDK 7 | Oracle JDK 8 | Oracle JDK 9 | Open JDK 11 / Oracle JDK 11 | Eclipse running Java 7 | Eclipse running Java 8 |
|---|---|---|---|---|---|---|
| Collaborator Eclipse plug-in 12.0 and later | No | Yes (built with this version) | Yes | Yes | No | Yes |
| Collaborator Eclipse plug-in 11.3 - 11.5.x | Yes | Yes (built with this version) | Yes | Yes | Yes | Yes |
| Collaborator Eclipse plug-in 9.x-11.2 | Yes (built with this version) | Yes | No | No | Yes | No |

## Rational Team Concert

To learn about compatibility between Collaborator and IBM Rational Team Concert, see the respective topic 702.

## Notes

- In table above, the "Yes (built with this version)" means that the given version of Collaborator was built using the given version of Java environment.
- OpenJDK 11 is recommended, Oracle JRE/JDK 8, 9, or 11 will work as well.
- On systems with multiple JREs installed, it may be necessary to specify to the installer which JRE should be used for Collaborator. On Windows platforms, running the installer with the -manual argument will suppress the JRE search and cause the installer to prompt for the JRE location (specifically, java.exe). On *nix platforms, you can specify the JRE location by setting the INSTALL4J_JAVA_HOME_OVERRIDE environment variable to the JAVA_HOME value.

# Index

## - A -



## - B -



## - C -



## - D -

# - E -



# - F -



# - G -



# - H -



# - I -



# - J -



# - K -



# - L -



# - M -

# - N -



# - O -



# - P -



# - R -



# - S -

## - T -



## - U -



## - V -



## - W -