AccuRev

AccuRev®

On-Line Help Guide

Version 6.2

Revised 19-January-2015

Copyright and Trademarks

Table of Contents

##

##

Preface

This book documents the AccuRev Graphical User Interface (GUI). It provides the content for the AccuRev context-sensitive help.

Audience

This document is intended for end users and administrators who use the AccuRev GUI in the performance of their jobs. It is not meant to be a tutorial—we assume that you have had some form of AccuRev training.

See the *AccuRev™ Installation and Release Notes* for installation procedures and any late-breaking information about the current release.

Using This Book

The contents of this book are intended to be used as on-line help within the GUI. This book format is provided only as a convenience for users who wish to browse through the help outside of the AccuRev client. Because the content was developed as on-line help, the book is not designed to be read in a linear fashion. You will find that discussions tend to jump from topic to topic with little transition material easing the way between them.

The chapters in this book are divided by functional areas:

Chapter	Description
Overview of the On-Line Help	Describes the AccuRev on-line help mechanism and its organization and features.
Using the AccuRev GUI	Provides a high-level view of the AccuRev GUI.
The File Browser	Describes in detail how to use AccuRev's features to work with your files.
The StreamBrowser	Describes in detail AccuRev's unique stream architecture for organizing your work.
The History Browser	Describes how to use AccuRev to view the past operations that have been performed on your files.
The Version Browser	Describes AccuRev's graphical browser for viewing versions, the streams and workspaces where they reside, and the relations between them.
Diff, Merge, Patch, and the Change Palette	Explains how to view the differences between versions of your files, how to merge those differences, and how to use the AccuRev's unique Change Palette to cross-promote elements to specific streams.
Security	Describes features such as file locking, stream locking, access control lists (ACLs), and other ways of controlling accessing to your data.

Chapter	Description
AccuWork	Describes AccuRev's tightly integrated issue tracking feature.
The Schema Editor (Administrators Only)	Explains the GUI utility that administrators use to design and customize AccuWork forms.
Support and Troubleshooting	Explains how to contact AccuRev Customer Support, provides discussions of some commonly-encountered issues, and also describes how to configure various web browsers to work with AccuRev on-line help.

Typographical Conventions

This book uses the following typographical conventions:

Convention	Description
`blue sanserif`	Used for sample code or output.
`red monospace`	Used for examples.
bold	Used for command names, and button names in the AccuRev GUI
light italic	Used for emphasis, book titles, and for first use of important terms
blue italic	Identifies a hyperlink (to a page or Web URL, for example)

Contacting Technical Support

AccuRev offers a variety of options to meet your technical support needs as summarized in the following table.

For	Visit
Information about technical support services	*http://supportline.microfocus.com/*
Information about platforms support	*http://supportline.microfocus.com/prodavail.aspx*
Product downloads and installations	*http://supportline.microfocus.com/websync/productupdatessearch.aspx*
Product documentation	*http://supportline.microfocus.com/productdoc.aspx*
SupportLine phone numbers, listed by country	*http://www.microfocus.com/about/contact/support/assistance.aspx*

When you contact AccuRev technical support, please include the following information:

- The version of AccuRev and any other AccuRev products you are using (AccuSync or GitCentric, for example).
- Your operating system.
- The version of relevant third-party software (if you are using AccuSync, for example, the version of your ITS).

- A brief description of the problem you are experiencing. Be sure to include which AccuRev interface you were using (Web user interface, Java GUI, or CLI), any error messages you received, what you were doing when the error occurred, whether the problem is reproducible, and so on.
- A description of any attempts you have made to resolve the issue.
- A simple assessment of how the issue affects your organization.

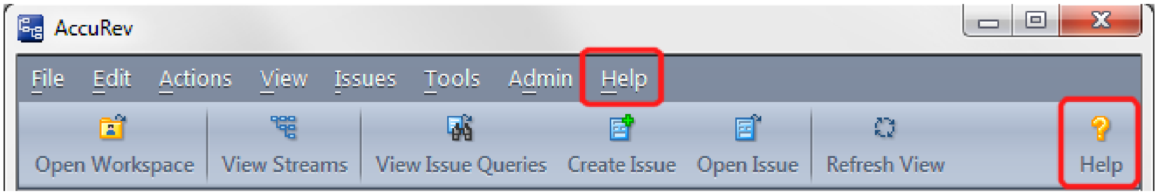
1. Overview of the On-Line Help

This document contains all of the information available in the AccuRev Graphical User Interface (GUI) on-line help. When you click on a **Help** button (or press the **F1** key) in AccuRev, a browser opens and you will be taken to a page somewhere within this book.

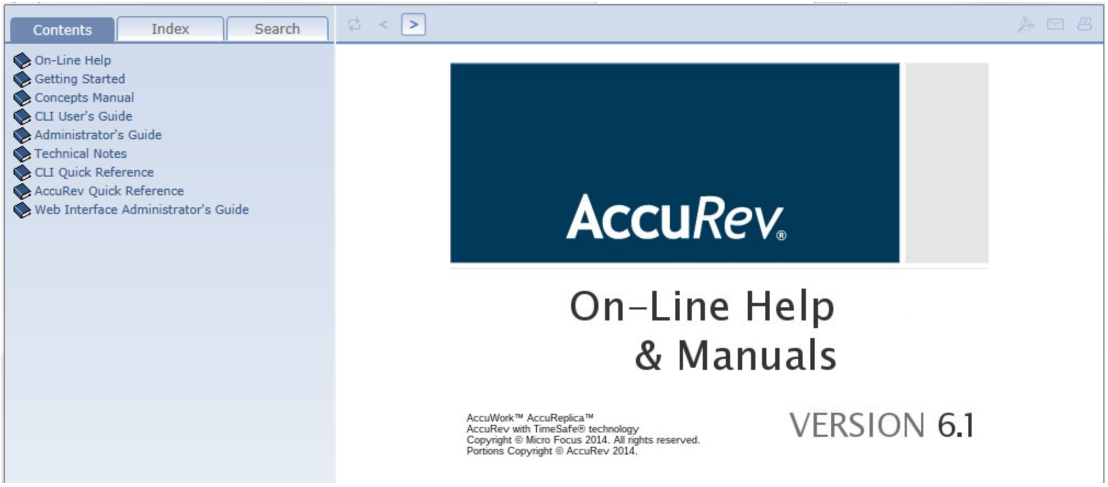
This chapter provides a quick overview of the AccuRev Help mechanism.

If You Are a New AccuRev User

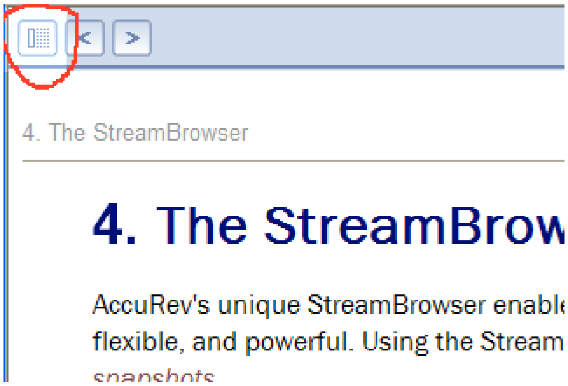
The AccuRev GUI provides extensive documentation in the form of a context-sensitive help system. This help not only covers the AccuRev GUI, but also provides access to the complete AccuRev documentation set, in both HTML and PDF format. The help can be invoked from the AccuRev **Help** menu, or from the **Help** ? button on the toolbar, for from the **F1** key, anywhere in the AccuRev GUI.



If you click on **Help > Help Contents & Manuals**, AccuRev opens a browser that includes a navigation pane, containing Contents (TOC), Index, and Search tabs.



If you click on a **Help** button or the **F1** key, or click on the **Help** menu option for "This Screen", AccuRev displays context-sensitive help focused on the topic of interest. In this case, the navigation pane is collapsed to save screen space, but it can be displayed by clicking the **Show Navigation** button in the upper-left corner of the display.



4. The StreamBrowser

4. The StreamBrow

AccuRev's unique StreamBrowser enabl
flexible, and powerful. Using the Stream
snapshots

> **Note:** Once the Navigation pane has been opened, it cannot be closed, but you can click and drag its right border toward the left and make it almost zero width.

All documentation is provided in both HTML and PDF. Typically you search or browse to the information you need in the HTML. If you decide you want to view the PDF version of the current document, you click the PDF button in the upper-right of the Help window.

Also, Windows users have direct access to the PDFs through the Windows ***Start -> AccuRev*** menu.

Finally, you can access the on-line help system without starting the AccuRev client by using the following URL: `<accurev_install>/doc/WebHelp/AccuRevHelp.htm`

If You Are Upgrading From a Previous AccuRev Version

If you have used the AccuRev GUI on-line help in releases previous to 5.2, you will notice a number of improvements.

- We have brought back a PDF version of the GUI Help documentation, named the *AccuRev On-Line Help Guide*.
- The entire document set (except for AccuBridge, WebUI Help, and Release Note documents) are now packaged together with the on-line help, and can be searched and viewed with a single mechanism. If you use a browser that accepts cookies from a local file system, you will also see a ***Favorites*** tab in the navigation page, which allows you to store links to frequently-accessed topics.
- We can now provide cross-links between the on-line help and the doc set, so if a help discussion refers to a chapter in the *Administrator's Guide*, you can click on a link and be taken directly to that discussion.

How Do I Configure My Browser?

As browsers have matured, they have often added tighter security requirements with new releases. Since the AccuRev help is based on Javascript, and is accessed from a local file system rather than through a web server, you may need to configure the security settings on your preferred browser to display the help. If the AccuRev help fails to display properly the first time you try to use it, please see *Configuring a Web Browser to View Help Topics* in *Appendix B Support and Troubleshooting* for more details, and check the Release Notes for any late-breaking changes.

Help Topics for Multiple Features

Most **Help** buttons within AccuRev go directly to the section of the documentation that addresses the current feature in the GUI. However, there is one situation where the same **Help** button can appear in different contexts. In this case, you are presented with a choice of topics. Depending on what you are doing in AccuRev, you will want to select:

- General help on the Update command: see *Update: Keeping Your Workspace Current* on page 129
- Help on the Update progress box only: see *The Update Progress Box* on page 132

  Note: You may also be directed to these topics if you were was using the **include/exclude** facility or **Populate**. The File Browser's **include/exclude** operations complete their work by performing an **Update** of the workspace. **Populate** command does not perform an **Update**, but uses the same low-level routines to copy files from the repository to the workspace tree.

2. Using the AccuRev GUI

This chapter describes the basics of using the AccuRev GUI.

Starting the GUI

You can start the AccuRev GUI from the desktop (Windows only) or from a command shell:

- On the Windows desktop, double-click the AR *AccuRev* icon. There is also an *AccuRev* group in the *Start* > *Programs* menu.
- In a command shell, use the *acgui* command:

```
> acgui      (Windows)
> acgui &    (UNIX)
```

The first time you open the AccuRev GUI, it displays the Welcome tab, which summarizes the highlights of the current release. After that, each time you open the AccuRev GUI it displays the tabs you had open in the preceding GUI session. For example, you might see:

- File Browser tabs, each displaying the contents of an individual workspace or stream. (See *The File Browser* for more information.)
- StreamBrowser tabs, each displaying the hierarchy of streams, snapshots, and workspaces for a particular depot. (See *The StreamBrowser* for more information.)
- AccuWork tabs, each showing the set of AccuWork queries defined for a particular depot (and, if you have defined a *default query*, its results). (See *AccuWork* for more information.)

Using AccuRev with a Secure AccuRev Server

AccuRev supports the Secure Sockets Layer (SSL) protocol to provide encrypted communication between the AccuRev client and the AccuRev server. Setup and configuration of SSL is managed by your AccuRev administrator. This section describes the situations in which you, as the end-user, are likely to experience SSL if it has been enabled on your AccuRev server.

The first time you start AccuRev, whether through the GUI or the command line, AccuRev will prompt you to enable SSL for that client. This process allows you to inspect the SSL certificate, and either accept or reject it. If you accept it, AccuRev downloads the certificate to the client machine, enabling you to connect to the AccuRev server. If you reject it, you cannot connect to the AccuRev server. This will prevent you from starting the AccuRev GUI or executing any AccuRev commands.

Once you accept the SSL certificate for your AccuRev client, you are prompted to take action in the future only if:

- **The SSL certificate expires or changes.** If the original certificates on the AccuRev server and client expire, AccuRev will prompt you to get a new certificate once the certificate on the AccuRev server is updated. In this case, you again have the opportunity to inspect the new certificate. If you accept it, AccuRev replaces the current certificate with a new one and you can continue using AccuRev. If you reject it, you will not be able to connect to the AccuRev server.

- **SSL is disabled on the AccuRev server.** If the administrator disables SSL on your AccuRev server after you have already accepted a certificate, the next time you start AccuRev or execute an AccuRev command you are informed of this situation and given the choice to disable SSL on your AccuRev client. If you choose to disable SSL, you will be able to continue using your AccuRev client, but the client-server communication will no longer be encrypted.

  Note: AccuRev recommends that before disabling SSL on your client you bring this situation to the attention of your AccuRev administrator.

The Login Command

The ***login*** command establishes your AccuRev user identity. In the "User Name" field, you must enter the name of a registered AccuRev user. If your organization has more than one AccuRev repository (each managed by a separate AccuRev server process), you can choose the server to which you want to connect.

> **Note:** *Active and inactive users.* A user name can be Inactive (***Remove User*** command) or Active (when first created; or when the ***Reactivate User*** command is used for an Inactive user). When you log in, the username must be currently **Active.**

See also: *Security* on page 263.

Invoking the Login Command

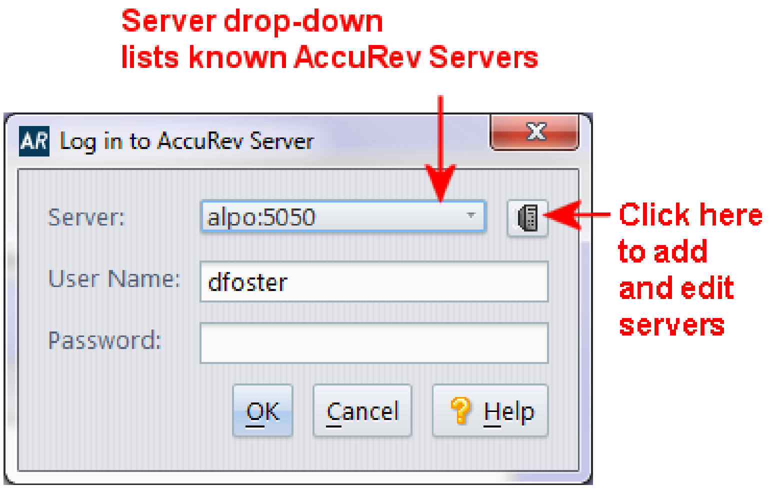
Choose ***Tools*** > ***Login*** from the GUI main menu. This command is invoked automatically:

- At the beginning of an AccuRev GUI session if there is no valid session file in the **.accurev** subdirectory of your home directory
- When you invoke a command that requires user authentication, but your session has expired

Using the Login Dialog

To log in:

Type in your user name -- and if you have one, your password -- then click **OK**. Passwords are case-sensitive.


Server drop-down lists known AccuRev Servers

Click here to add and edit servers

Server: alpo:5050

User Name: dfoster

Password:

To switch to another server:

Open the listbox by clicking the down-arrow, then select another AccuRev Server (hostname/port-number combination).

Then, complete the login by typing your username (and password).

To add an AccuRev Server to the listbox

1. Click the ***Edit Available Servers*** button ( ).

   The Available AccuRev Servers dialog appears. See *The Available AccuRev Servers Dialog (and the New Server and Edit Server dialogs)* on page 284 for more information.

2. Click the ***Add Server*** button ( ) to specify:

   - The hostname (or IP address) where the additional AccuRev Server runs.
   - The IP port number on which it listens for AccuRev client requests.

Tip: The ACCUREV_SERVER environment variable influences which AccuRev Server the AccuRev GUI interacts with at startup time. See *ENV_VARS* on page 264 of the *AccuRev CLI User's Guide* for more information.

Logging Out

To log out of AccuRev, you must use the **accurev logout** command:

```
> accurev logout
```

Logging out cancels the session that is created when you log in to AccuRev. See *The "session" File* on page 7 for more information.

Whether or not the session is canceled when you exit the AccuRev GUI is controlled by the **Keep Session Active** user preference. This preference is on by default -- your session remains active even after you exit the AccuRev GUI. See *AccuRev Preferences (Tools > Preferences Command)* on page 39 for more information on setting preferences.

The "session" File

A successful ***login*** command creates an encrypted file in the **.accurev** subdirectory of your home directory. This file records your AccuRev username and password, along with the IP address of your client machine. Most AccuRev client commands can be executed only by an authorized user. Such commands send the information in your session file to the AccuRev Server process, so that you don't need to repeatedly "remind" the AccuRev Server who (and where) you are.

Tip: You can specify whether you want AccuRev to cancel the user session when you exit the AccuRev GUI. See *Keep Session Active* on page 41 for more information.

Session Expiration

Session files are automatically deleted by the AccuRev Server after an administrator-configurable interval, which defaults to 240 minutes. This implements a session-expiration feature. Administrators can also actively cancel all sessions associated with a user using the **accurev remove sessions** command. See *remove* on page 212 of the *CLI User's Guide* for more information.

Multiple Session Files for Different Servers

The name of the session file includes the hostname and port number of the AccuRev Server. If you ***Login*** to different AccuRev Server processes, you'll have multiple session files, one for each Server. Example:

```
session_VENUS_5050
session_MARS_5050
session_MARS_5999
```

These session files indicate that your are logged in to an AccuRev Server on host *venus*, listening on port 5050, and are also logged into two different AccuRev Server processes on host *mars*.

Multiple Session Files for the Same Server

What if you want to be testuser *john* in one GUI window, but testuser *mary* in another -- both using the same AccuRev Server? You can't have two session files with the same name (for example, *session_VENUS_5050*) in your **.accurev** subdirectory. But you can have two or more session files with the same name in different .**accurev** subdirectories. The AccuRev GUI uses the value of environment variable ACCUREV_HOME as the location of the **.accurev** subdirectory (and will create this subdirectory, if necessary). It also uses ACCUREV_HOME whenever it checking for the existence of a session file in order to authenticate you.

Example (Windows):

1. Set ACCUREV_HOME to *c:\myusers\john*.
2. Create directory *c:\myusers\john*.
3. Start the AccuRev GUI.
4. Login as *john*.
5. Set ACCUREV_HOME to *c:\myusers\mary*.
6. Create directory *c:\myusers\mary*.
7. Start another instance of the AccuRev GUI.
8. Login as *mary* in this second instance.

Hint: Setting up scripts to run the AccuRev GUI makes it easier to ensure that the GUI will be running with the environment variable set correctly.

The Change Password Command

The ***Change Password*** command changes (or creates) the password for your AccuRev user identity.

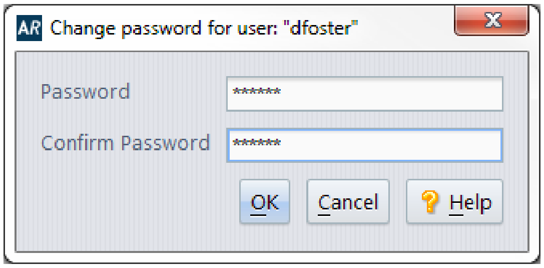
See also: *Security* on page 263.

Invoking the Change Password Command

Choose ***Tools*** > ***Change Password*** from the GUI main menu.

Using the Change Password Dialog

Type the new password twice, once in the Password field, and again in the Confirm Password field.



Change password for user: "dfoster"

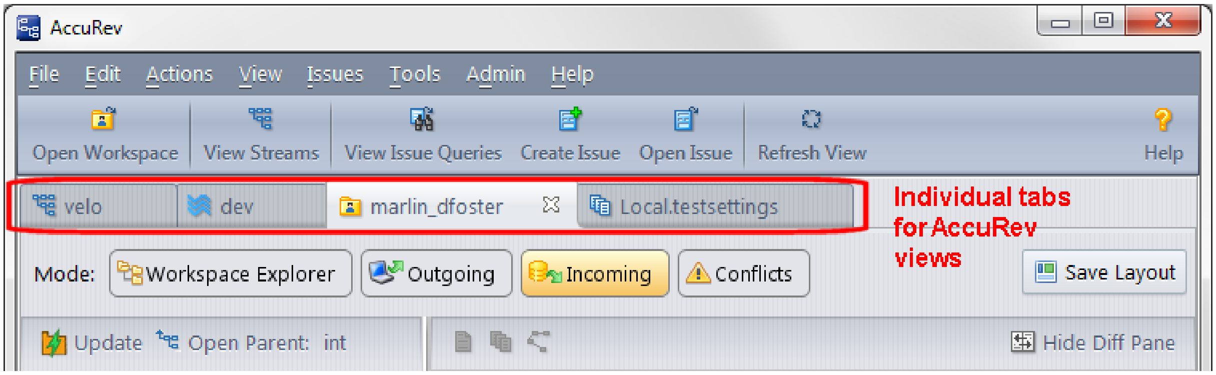
Password: *******

Confirm Password: *******

The change takes place only if your two entries match exactly.

The Multiple-Tab Display

The AccuRev GUI window uses multiple tabs to enable you to switch quickly among several activities.

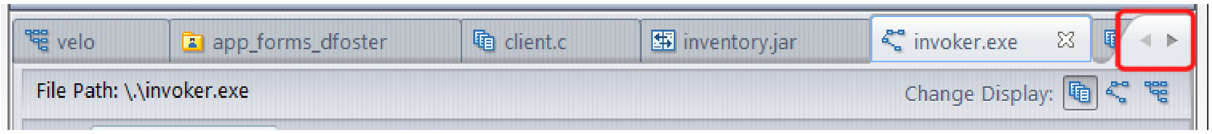


For example, you might wish to switch among:

- Viewing the contents of your main workspace, using the File Browser.
- Viewing some or all the versions of a particular file, using the Version Browser.
- Viewing a list of all depots in the AccuRev repository.
- Viewing the streams in a particular depot, using the StreamBrowser.

Organizing Tabs

By default, AccuRev displays tabs in a single row. When the number of tabs exceeds the available space to display them, AccuRev adds a scroll button that provides access to tabs that are not currently visible:

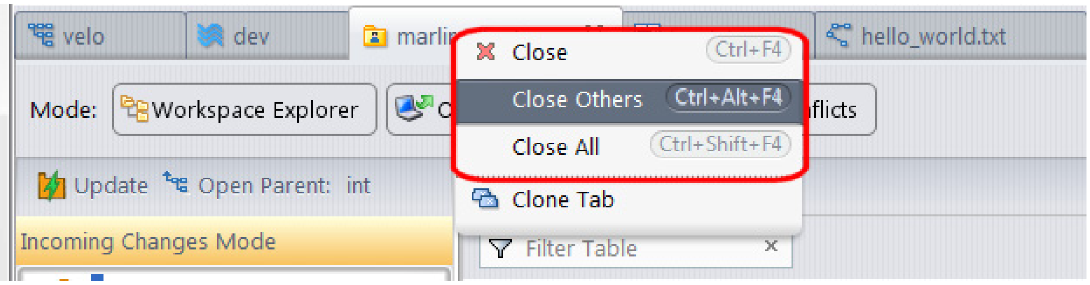


If you prefer, you can set a preference so that AccuRev displays the tabs in as many rows as needed. See Tab Layout on page 42 for more information.

Closing Tabs

At any time, you can "clean up" by closing one or more of the tabs in either of these ways:

- Right-click on a tab control to display its context menu, then select *Close*, *Close Others*, or *Close All*.



- Click the "X" icon on the tab control.

Cloning Tabs

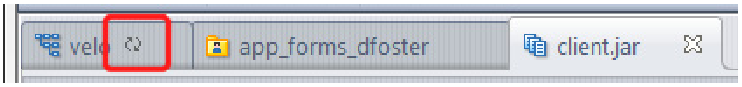
The AccuRev GUI assumes that you want to avoid having multiple tabs with the same contents. So, for example, if there's already a File Browser tab on workspace *app_forms_dfoster* and you execute an *Open Workspace* command on the same workspace, the GUI simply brings the existing tab to the front.

If you do want to have two or more tabs showing the same data structure, use the *File > Clone Tab* command (or choose *Clone Tab* from the tab context menu). Cloned tabs enable you, for example, to work in two different parts of a large stream hierarchy (two StreamBrowser tabs), or to work in two different folders of a workspace (two File Browser tabs).

> **Note:** The History Browser, Version Browser, and Stream Version Browser do not support tab-cloning .

Refreshing Tabs

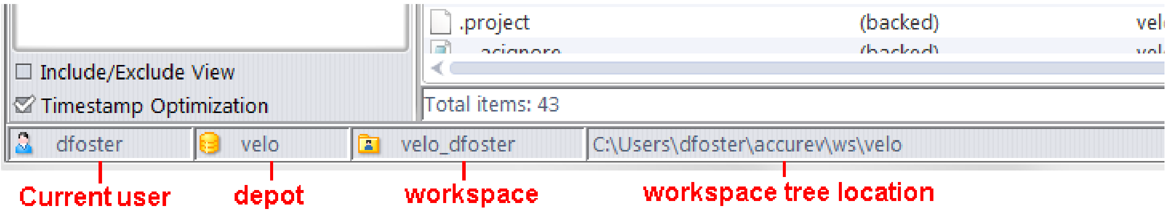
The data displayed on a GUI tab can become out of date as a result of your work on other GUI tabs, your work using the AccuRev CLI, and/or other users' work. AccuRev displays a "refresh" icon next to the tab title if it determines that the tab's data may be out of date.



The command *View > Refresh* (or function key **F5,** or the main GUI menu toolbar ***Refresh*** icon) updates a tab's data. You can configure the GUI to refresh tab data automatically whenever you switch tabs: invoke the command *Tools > Preferences*, and clear the *Require Manual Refresh* checkbox.

Current User, Depot, and Workspace

With a few exceptions, you must be logged in (See *"The Login Command"* on page 6.) as a particular AccuRev user in order to execute AccuRev commands. An indicator in the lower left corner of the GUI window shows the **current user** (the one who is currently logged in).



Typically, the AccuRev repository is organized into multiple depots. For example, there might be separate depots for the development, testing, and documentation groups. Most GUI tabs display the data from one particular depot. When you're using a particular tab, the associated depot is termed the **current depot**. Its name is displayed at the bottom of the GUI window.

Many tabs have a **current workspace** context: the workspace that is the source of (or the possible destination of) the data that the tab displays. The name of the current workspace is displayed at the bottom of the GUI window, along with the pathname of its *workspace tree*.

Keyboard Accelerators

In a dialog box, you can "press" buttons using **Alt** key sequences, such as **Alt-C** for ***Cancel***. The **Esc** key also performs a ***Cancel*** operation in any dialog box.

In some dialog boxes, you don't even need to use the **Alt** key when invoking a keyboard accelerator: In a "Yes/No" dialog box, you can type **Y** or **N** (either uppercase or lowercase). In a message or confirmation box (no input field), you can indicate ***Ok*** by typing **O** (either uppercase or lowercase), or by pressing **Enter**.

Many commands can be invoked with a keyboard accelerator, without opening any menu at all. The accelerators for such commands are listed on the menu itself. Keyboard accelerators for frequently performed commands are summarized in the following table.

Command	Keyboard Accelerator	Availability
Update	Ctrl+U	File Browser
Add	Ctrl+Shift+A	▪ File Browser ▪ Default Group
Keep	Ctrl+K	▪ File Browser ▪ Default Group
Promote	Ctrl+Shift+P	▪ File Browser ▪ Default Group
Merge	Ctrl+M	▪ File Browser ▪ Default Group
Element Permissions	Ctrl+E	▪ File Browser ▪ Default Group
Defunct	Ctrl+T	▪ File Browser ▪ Default Group
Diff Against ▪ Most Recent Version ▪ Backed Version ▪ Basis Version ▪ Overlapping Changes (Mine) ▪ Overlapping Changes (Theirs) ▪ File in Other Stream ▪ File on Disk	 ▪ Ctrl+Shift+D ▪ Ctrl+D ▪ Ctrl+Shift+B ▪ Ctrl+Shift+M ▪ Ctrl+Shift+T ▪ Ctrl+Shift+O ▪ Ctrl+Shift+F	▪ File Browser ▪ Default Group
Revert to ▪ Most Recent ▪ Basis	 ▪ Ctrl+R ▪ Ctrl+Shift+R	▪ File Browser ▪ Default Group
History ▪ Show History ▪ Browse Versions ▪ Browse Stream Versions	 ▪ Ctrl+H ▪ Ctrl+Shift+H ▪ Ctrl+Shift+Alt+H	▪ File Browser ▪ Default Group

Command	Keyboard Accelerator	Availability
Go to ▪ Previous diff ▪ Next diff ▪ First diff ▪ Last diff	▪ Ctrl+↑ ▪ Ctrl+↓ ▪ Ctrl+Shift+↑ ▪ Ctrl+Shift+↓	Diff/Merge view
Search	Ctrl+F	▪ File Browser ▪ Diff/Merge view
Cut	Ctrl+X	Text fields and some tables
Copy	Ctrl+C	Text fields and some tables
Paste	Ctrl+V	Text fields and some tables
Select All	Ctrl+A	Text fields and some tables
Save	Ctrl+S	Diff/Merge views
Print	Ctrl+P	Tables in all views

Integration with the AccuRev Web UI

In addition to the AccuRev GUI described in this document, AccuRev also provides a Web User Interface (Web UI) which gives users access to many non-filesystem-based AccuRev features from any supported web browser. The Web UI server must be installed and configured by an AccuRev administrator, who will provide AccuRev users with a web address through which they can access the Web UI functionality.

Once configured, the Web UI can be easily accessed from the AccuRev GUI via two choices on the **View** menu:

Open in Web

When you click this button, the current view will be opened in a web browser.

Copy URL to Clipboard

Stores a URL that you can use to bring up the current view in a web browser. This is useful if you would like to share the current view with another user. Use keyboard sequence CTRL-V, or an application “paste” command to retrieve the text from the clipboard. (For example, Windows users might open Notepad, and use the Edit>Paste command to enter the URL into the open document.)

Also, the Web UI will automatically come up when you access an AccuWork issue if any of the following three conditions is true:

- If the AccuRev administrator has configured the site to view issues only in the Web UI (see *Configuring the Web User Interface (Web UI)* on page 29 of the AccuRev *Administrator’s Guide*).
- If you have set your local preferences to view issues only in the Web UI (see *AccuRev Preferences (Tools > Preferences Command)* on page 39).

There are certain views that cannot be opened from the AccuRev GUI in the Web UI. For example, manipulation of files in a local file system cannot be performed from a web browser for security reasons, so while you can get a File Browser web view of a dynamic stream, you cannot bring up a File Browser web view of your workspace. In these cases, you will see the message “URL not available for this view” when you try to use these buttons.

Note: For these features to function, the AccuRev Web UI server must have been installed and configured with your AccuRev server. If your web browser or clipboard show a URL containing the string “null” (for example, `null/stream/1/?view=sb`), the AccuRev Web UI server is not available. Information about installing and configuring the AccuRev Web UI server is available to AccuRev administrators in the AccuRev *Installation and Release Notes,* the AccuRev *Web Interface Installation and Release Notes*, and the AccuRev *Administrator's Guide*.

While some features that are available in the AccuRev GUI are not available in the Web UI, other features, such as ***Bulk Update*** for AccuWork issues, are available only in the Web UI.

Find, Search, and Filter Features

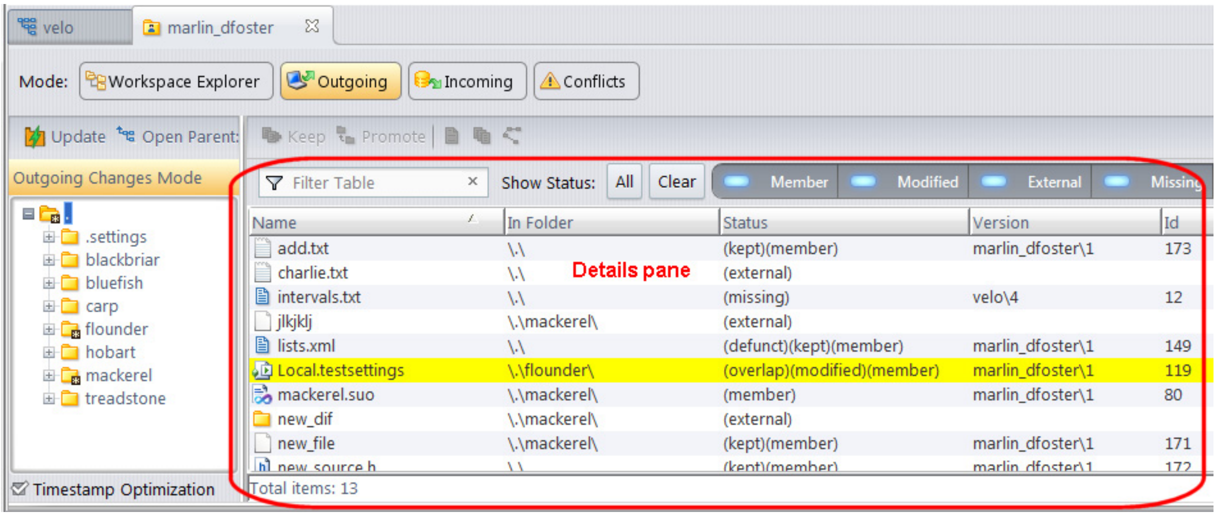
The AccuRev GUI provides several ways to locate and filter information:

- To display elements with specific statuses, see *Chapter 3 The File Browser*.
- To limit the number of streams displayed in the StreamBrowser, see *Filtering Streams, Snapshots, and Workspaces* on page 149 and *Using Stream Favorites* on page 160.
- To search for elements by name in your workspace, see *Searching the Workspace* on page 63.
- To find streams by name in the StreamBrowser, see *Find Stream* on page 168.
- To filter data displayed in tables, see *Filtering Tables* on page 15.
- To filter the results of various list dialogs, such as the **Choose Depot** dialog (see *Filter Results* on page 21).

Working with Tables

The AccuRev GUI often presents information in the form of a table, with multiple rows and multiple columns. In many contexts, the AccuRev GUI displays data in the form of a rows-and-columns table.

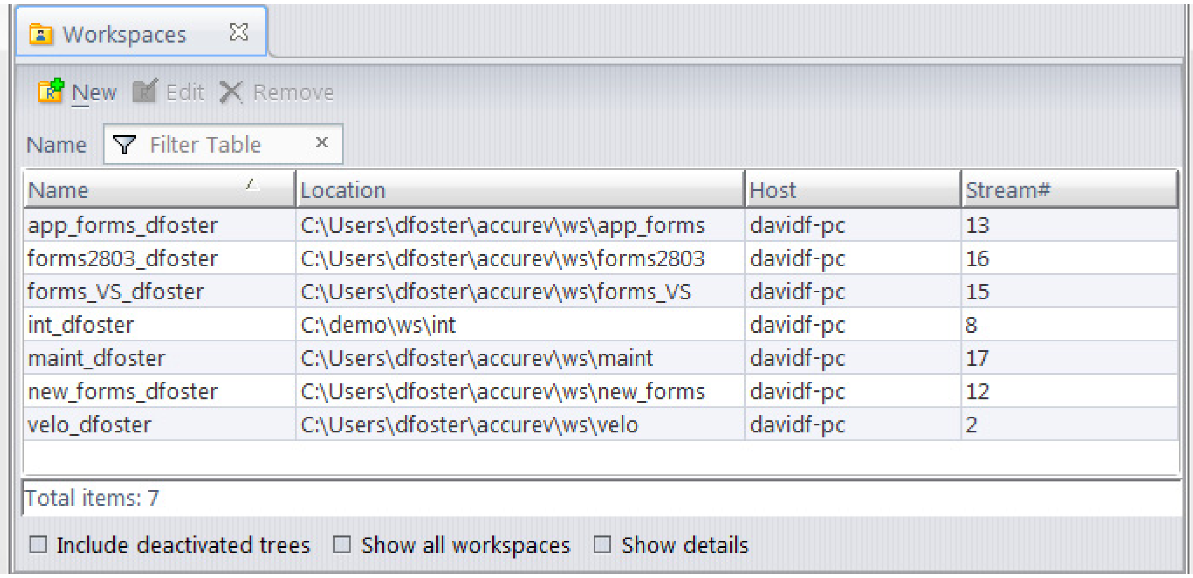
Example: File Browser's Details pane



Name	In Folder	Status	Version	Id
add.txt	\.\	(kept)(member)	marlin_dfoster\1	173
charlie.txt	\.\	(external)		
intervals.txt	\.\	(missing)	velo\4	12
jlkjklj	\.\mackerel\	(external)		
lists.xml	\.\	(defunct)(kept)(member)	marlin_dfoster\1	149
Local.testsettings	\.\flounder\	(overlap)(modified)(member)	marlin_dfoster\1	119
mackerel.suo	\.\mackerel\	(member)	marlin_dfoster\1	80
new_dif	\.\mackerel\	(external)		
new_file	\.\mackerel\	(kept)(member)	marlin_dfoster\1	171
new_source.h	\.\	(kept)(member)	marlin_dfoster\1	172

Total items: 13

Example: Workspaces tab



Name	Location	Host	Stream#
app_forms_dfoster	C:\Users\dfoster\accurev\ws\app_forms	davidf-pc	13
forms2803_dfoster	C:\Users\dfoster\accurev\ws\forms2803	davidf-pc	16
forms_VS_dfoster	C:\Users\dfoster\accurev\ws\forms_VS	davidf-pc	15
int_dfoster	C:\demo\ws\int	davidf-pc	8
maint_dfoster	C:\Users\dfoster\accurev\ws\maint	davidf-pc	17
new_forms_dfoster	C:\Users\dfoster\accurev\ws\new_forms	davidf-pc	12
velo_dfoster	C:\Users\dfoster\accurev\ws\velo	davidf-pc	2

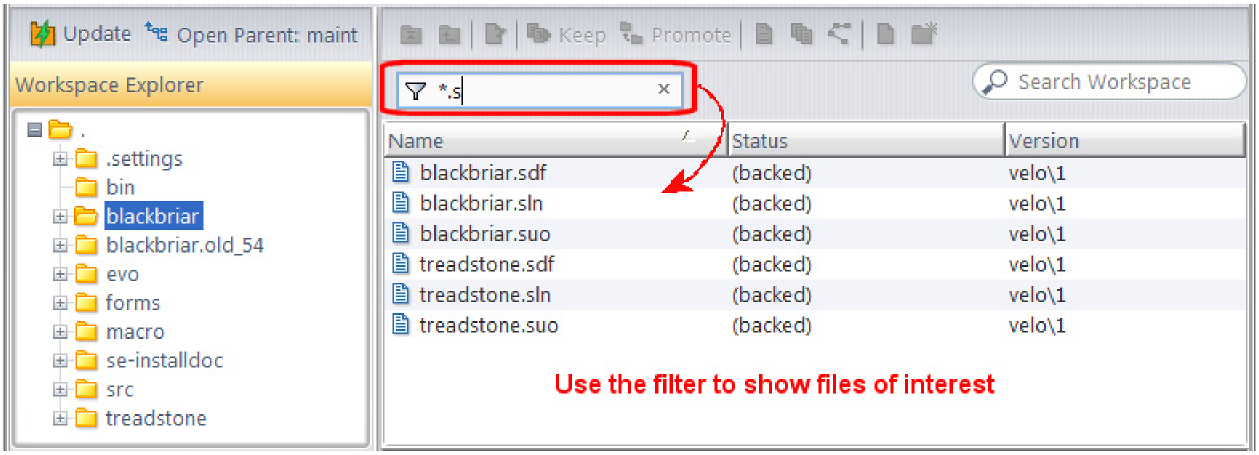
Total items: 7

☐ Include deactivated trees ☐ Show all workspaces ☐ Show details

You can adjust all such tables, to maximize their usefulness, as described in the following sections.

Filtering Tables

Generally speaking, the details panel displays the contents of whatever node has been selected in the explorer tree. You can use the Filter Table entry field to limit the display to only those files of interest to you -- for example, you might want to see only those files that contain the letter "s"in the file extension, as shown in the following illustration:



*.s

Name	Status	Version
blackbriar.sdf	(backed)	velo\1
blackbriar.sln	(backed)	velo\1
blackbriar.suo	(backed)	velo\1
treadstone.sdf	(backed)	velo\1
treadstone.sln	(backed)	velo\1
treadstone.suo	(backed)	velo\1

Use the filter to show files of interest

As you type in the Filter Table field, the files displayed in the details pane are narrowed down to only those containing the characters you enter. The Filter Table field takes wildcards like ? and *.

To clear the filter, click the X button in the **Filter Results** field.

Tip: The results of the Filter Table field are temporary -- AccuRev lets you filter tables independently, from view to view. If you prefer, you can prevent AccuRev from displaying files based on an ignore patterns that you specify. You might want to exclude from view all backup and temporary files, for example. See *Controlling the Display of External Objects* on page 58 for more information.

Selecting Files in a Table

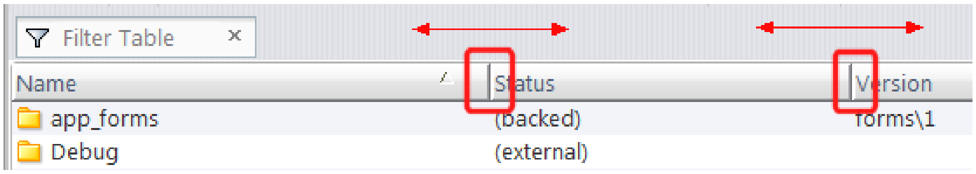
The file-selection gestures are the same ones used by Windows Explorer:

- To select a file, click it with mouse button 1.
- To select a contiguous range of files, click-and-drag with mouse button 1.
- To select all the files in a directory, select any one of them, then press **Ctrl-A**.
- To add or subtract a file from an existing selection, hold down the **Ctrl** key and click that file with mouse button 1.
- To extend an existing selection to a particular file, hold down the **Shift** key and click that file with mouse button 1.
- Typing a character selects the next file or directory whose name begins with that character (if the Details pane listing is sorted on the Name or Element column).

Adjusting the Column Width

You can adjust columns widths in one of two ways:

- Drag the column separators.



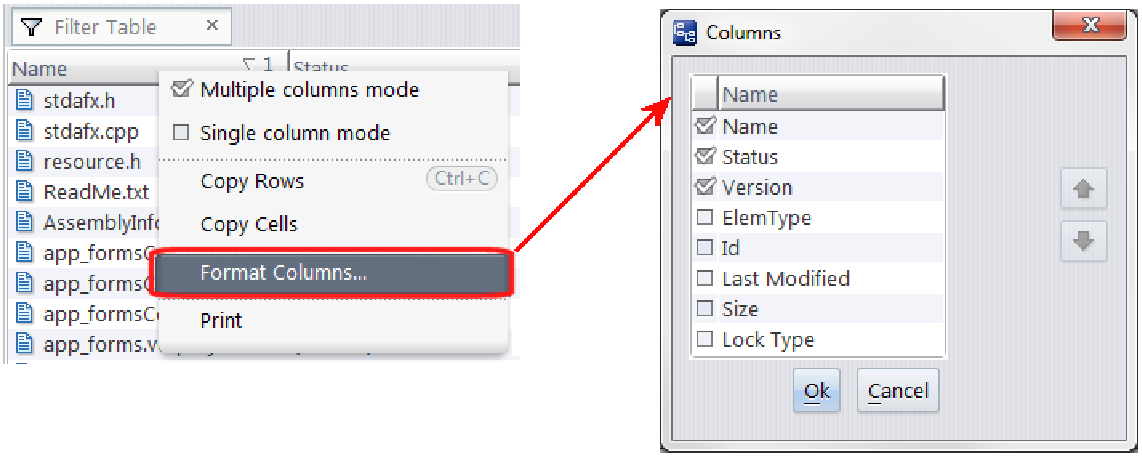
- Double-click the column separator to a column's right to resize that column to exactly fit its longest value.

Adding and Removing Columns

The columns AccuRev displays in tables vary across views. If you want, you can change the columns that are displayed using the Columns dialog box. To change the columns that are displayed in a table:

1. Right-click the column heading and choose **Format Columns**.

   The Columns dialog box appears.



2. Select the columns you want to display in the details pane and click **Ok**.

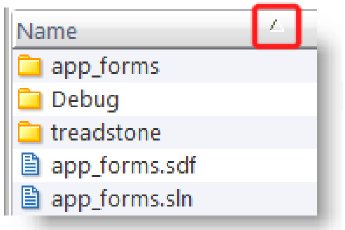
Changing Column Order

You can change the order in which columns appear in two ways:

- In the details pane, drag and drop the column to the place you want it.
- In the Columns dialog box:
  1. Select the column you want to move in the Columns dialog box.
  2. In the Columns dialog box, Use the up and down arrows to move the column.
  3. Click **Ok**.

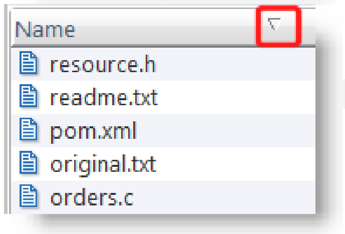
Sorting the Rows of a Table

Initially, the rows of a table are sorted on one column (**single-column mode**). A direction icon in the header for that column indicates whether the sort is lowest-to-highest or highest-to-lowest.



Name △
app_forms
Debug
treadstone
app_forms.sdf
app_forms.sln

Ascending order



Name ▽
resource.h
readme.txt
pom.xml
original.txt
orders.c

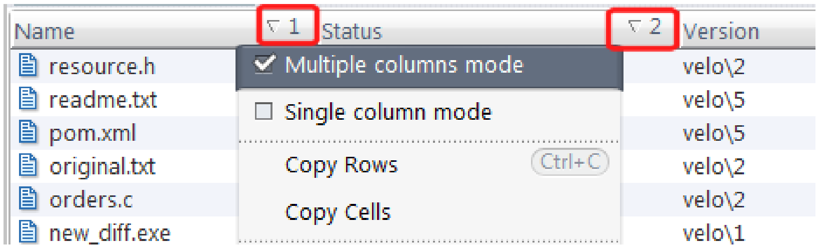
Descending order

Note: In the Details pane of the File Browser (and any other table that lists file names and directory names in the same column), a lowest-to-highest sort places all directories before all files; a highest-to-lowest sort places all files before all directories.

Left-click on any column header to continue in this mode:

- Click on the current sort column to reverse the sort order.
- Click on another column to make it the sort column.

You can switch at any time to **multiple-columns mode**, in which you define a primary sort column, a secondary sort column, and so on. Right-click any column header to switch sort modes.



Name ▽ 1	Status ▽ 2	Version
resource.h		velo\2
readme.txt		velo\5
pom.xml		velo\5
original.txt		velo\2
orders.c		velo\2
new_diff.exe		velo\1

A "1" appears next to the direction icon in the current sort column, indicating that this is now the primary sort column.

Click another column to make it the secondary sort column. A direction icon annotated with "2" appears in this column, and the rows are reordered according to the two-level sort. Continue in this way to define additional sort levels.

In multiple columns mode, you can:

- Left-click on any column header without a direction icon: this defines an additional sort level, using that column.
- Left-click on any column header that already has a direction icon: this reverses the sort order at that sort level.
- Left-click on any sort column's level number (1, 2, etc.) to change the sort level of that column.

The table's rows are resorted automatically each time you perform any of these actions.

Using the Keyboard to Navigate through a Table

The keyboard's navigation keys -- up-arrow, down-arrow, **PgUp**, **PgDn**, **Ctrl-PgUp**, **Ctrl-PgDn** --move a table's selection highlight in the expected way. In addition, you can navigate by typing any alphanumeric character; this moves the selection highlight to the next row whose entry in the (primary) sort column begins with that character.

Printing Tables

You can print tables in views throughout the AccuRev GUI by choosing **File** > **Print** from the AccuRev menu or by typing **Ctrl+P**. You cannot print tables that appear in dialog boxes.

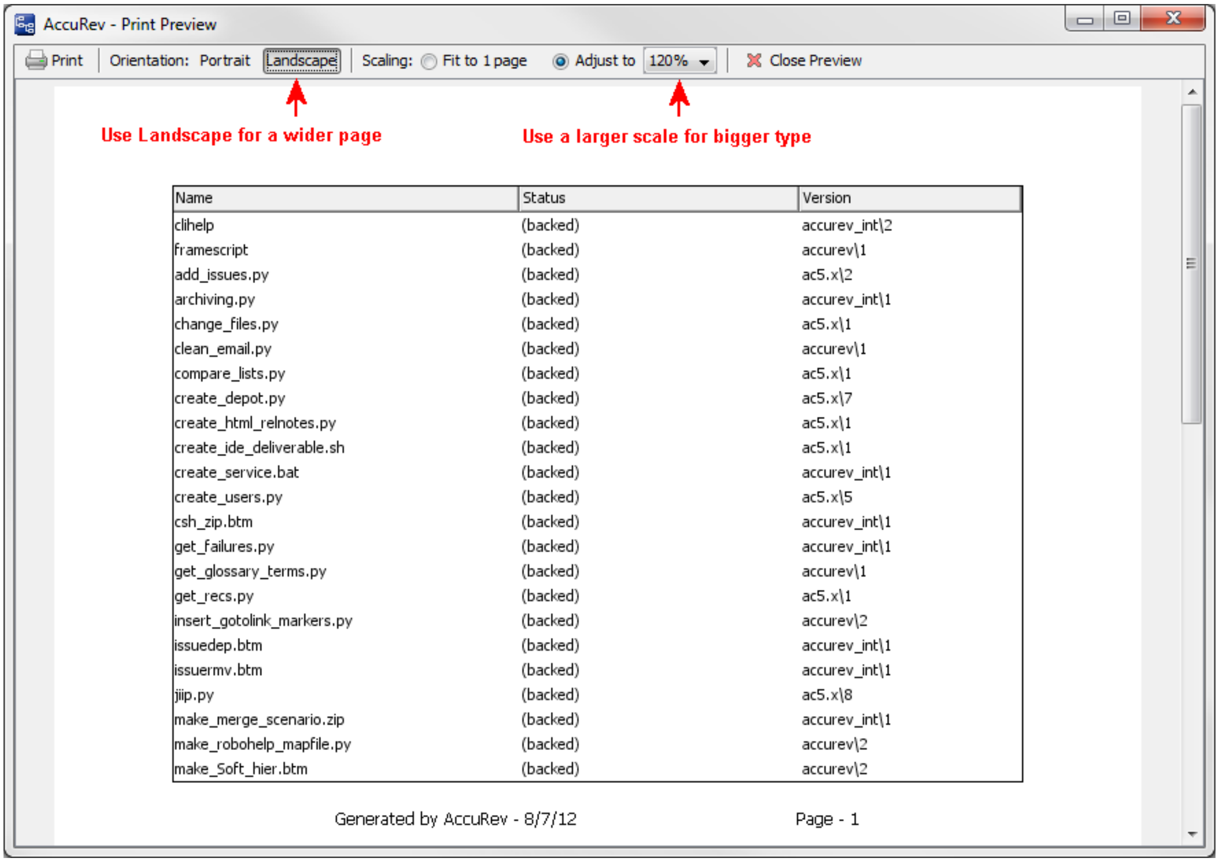
What Gets Printed

When you print a table, AccuRev prints the table headings and table contents for every column and row in the table. Column widths and sort order are preserved; you might want to adjust the width of a table’s columns or change the sort order before you print. Printed pages include a footer that displays the date the table was printed and a page number. If you are printing a table of query results, the query name is printed at the top of each page.

Note that AccuRev prints a table’s entire contents, even if only a portion of it is visible online. For example, if a query returns 65 records, the printed table includes all 65 records, even if only 30 are visible at a time. Similarly, some tables in AccuRev and AccuWork use a single line per row and display ellipses (...) for columns that are not wide enough to display their full contents. Printed tables are not limited to one line per row, and AccuRev wraps column contents, if necessary, when printing the table.

Previewing What You Print

When you print a table, AccuRev displays a preview window that lets you see what the printed page will look like.



Use Landscape for a wider page

Use a larger scale for bigger type

Name	Status	Version
clihelp	(backed)	accurev_int\2
framescript	(backed)	accurev\1
add_issues.py	(backed)	ac5.x\2
archiving.py	(backed)	accurev_int\1
change_files.py	(backed)	ac5.x\1
clean_email.py	(backed)	accurev\1
compare_lists.py	(backed)	ac5.x\1
create_depot.py	(backed)	ac5.x\7
create_html_relnotes.py	(backed)	ac5.x\1
create_ide_deliverable.sh	(backed)	ac5.x\1
create_service.bat	(backed)	accurev_int\1
create_users.py	(backed)	ac5.x\5
csh_zip.btm	(backed)	accurev_int\1
get_failures.py	(backed)	accurev_int\1
get_glossary_terms.py	(backed)	accurev\1
get_recs.py	(backed)	ac5.x\1
insert_gotolink_markers.py	(backed)	accurev\2
issuedep.btm	(backed)	accurev_int\1
issuermv.btm	(backed)	accurev_int\1
jiip.py	(backed)	ac5.x\8
make_merge_scenario.zip	(backed)	accurev_int\1
make_robohelp_mapfile.py	(backed)	accurev\2
make_Soft_hier.btm	(backed)	accurev\2

Generated by AccuRev - 8/7/12   Page - 1

The preview window provides several settings that give you additional control over the output, as summarized in the following table.

Setting	Description
Orientation	Lets you choose between Portrait and Landscape layout. Portrait is the default. *Tip*: Try changing the orientation to Landscape to fit more columns on a single page.
Scaling	Lets you specify whether you want: ▪ All table columns printed on the same page (Fit to 1 page) ▪ The scale of the table to determine how many columns fit on a single page By default, the Scaling setting is set to 100%, which provides good readability. *Tip*: Use a larger scale for bigger type.

How to Print a Table

To print a table:

1. Display the table you want to print.
2. Choose **File** > **Print** from the AccuRev menu, or type **Ctrl+P**.

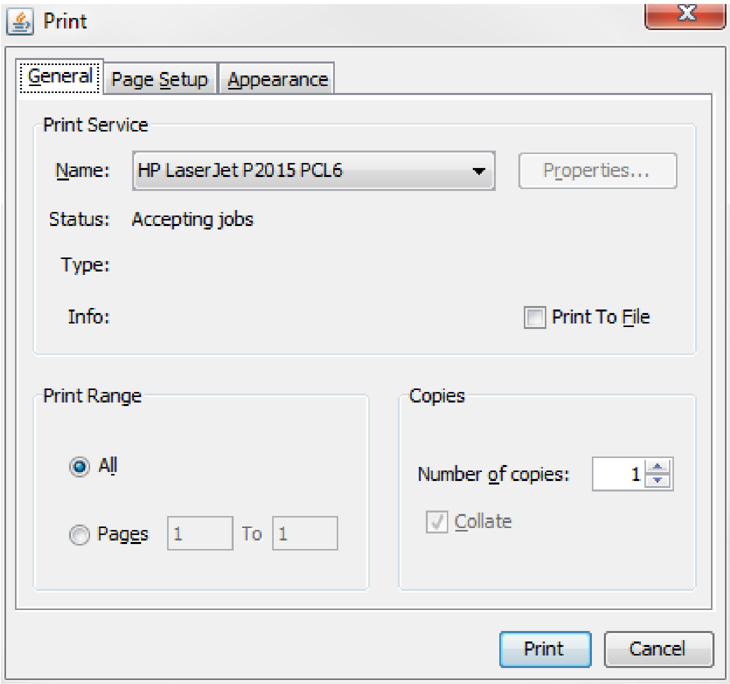
AccuRev displays the Print Preview window.

3. Optionally, use the preview settings to adjust the output. See *Previewing What You Print* on page 19 if you need help with this step.

   Tip: If using the preview settings does not provide the output you expect, click the **Close Preview** button and make adjustments on the source table (resizing or adding/dropping columns, for example).

4. When you are satisfied with the output displayed in the Print Preview window, click the **Print** button (🖨).

   AccuRev displays the familiar Print dialog box, an example of which is shown here.



Name: HP LaserJet P2015 PCL6

Status: Accepting jobs

Type:

Info:

☐ Print To File

● All

○ Pages 1 To 1

Number of copies: 1

☑ Collate

You use the fields on the Print dialog box to specify characteristics of the print job (specifying a printer, the number of pages you want to print, and so on).

5. Change any print job settings as needed and click **Print** to print the table.

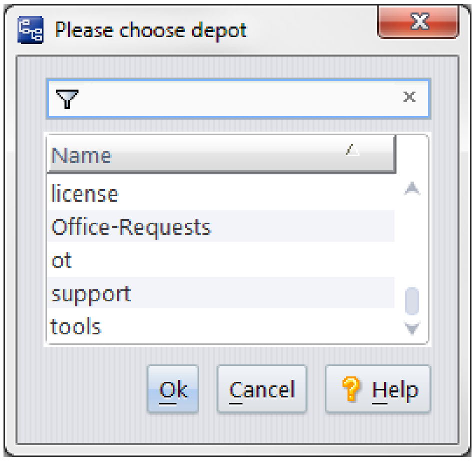
Choosing a Depot to be Used for a Command

In many situations, AccuRev must execute a command in the context of a particular depot. In many cases, it selects the depot context automatically. In other cases, it prompts you to select a depot from a list of all the depots in the AccuRev repository.

Using the Choose Depot Dialog

The depot names are initially sorted alphabetically (A-Z). You can click the **Name** header to reverse the sort order (Z-A).Choose a depot in either of these ways:

- Select a depot name and click ***Ok***.
- Double-click a depot name.



Filter Results

Optionally, use the filter field at the top of the dialog to narrow down your results. If only one element remains, AccuRev automatically selects it.

Choosing a Stream to be Used for a Command

In some situations, AccuRev prompts you to select a stream from a list of all the streams in a particular depot. If you have set a Stream Favorite, not all streams appear in this dialog. See *Using Stream Favorites* on page 160 for more information.

Using the Choose Stream Dialog

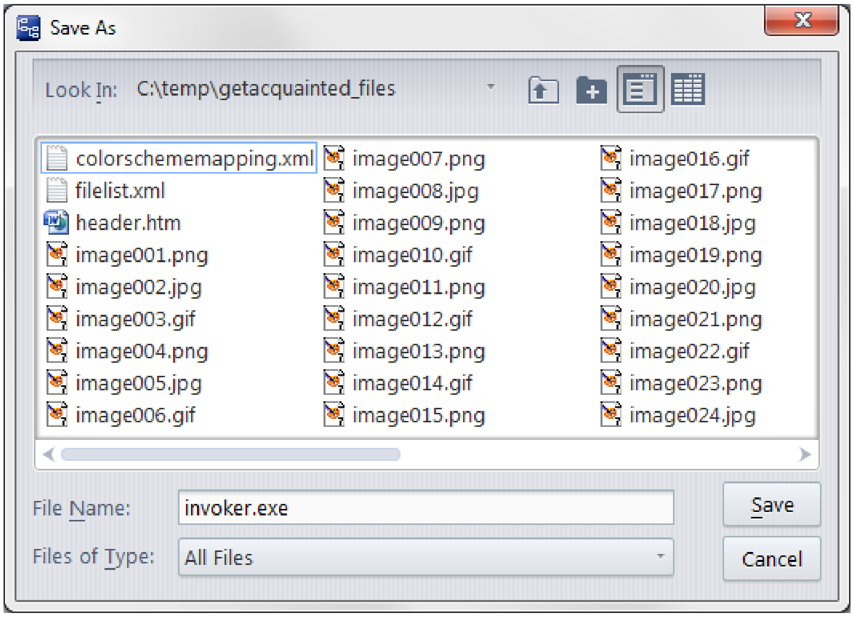
The stream names are initially sorted alphabetically (A-Z). You can click the **Name** header to reverse the sort order (Z-A).

Choose a stream in any of these ways:

- Select a stream name and click ***Ok***.
- Double-click a stream name.

The File Chooser Dialog

In many contexts, AccuRev prompts you to specify a location in your machine's local disk storage.



Look In: C:\temp\getacquainted_files

File Name: invoker.exe

Files of Type: All Files

In each context where you choose a file for input or output, AccuRev suggests the directory you most recently used in that context (or defaults to your home directory).

Where appropriate — for example, in specifying attachments to an issue record — you can specify multiple files at the same time using the standard multiple-selection mechanism: hold down the CTRL key while selecting the items .

In some contexts in which you specify a file (or other data structure) to be created, the dialog includes a ***New Folder*** button. This enables you to create a new directory 'on the fly' in which to create the data structure.

AccuWork Export File Types

The AccuWork ***Export*** and ***Export Table*** commands create a file from the contents of one or more issue records. In these contexts, the export file must have one of these formats, which you choose from the File Type listbox:

HTML

Standard Web-page format. On Windows machines, AccuWork automatically invokes your default web browser, to display the export file.

The contents of the issue record are rendered as one or more HTML tables. In a web browser, each table cell changes height and width when you adjust the size of the browser window. This is particularly useful for viewing the contents of multiple-line text fields.

XML

A more highly structured document, suitable for data interchange. Many web browsers can display XML documents.

Note: The XML-format file does not include a DTD or Schema.

CSV

("comma-separated values") A line-by-line text format, suitable for importing into various applications. On Windows machines, AccuWork automatically attempts to invoke Microsoft Excel, to display the export file.

User-Specified Names for AccuRev Entities

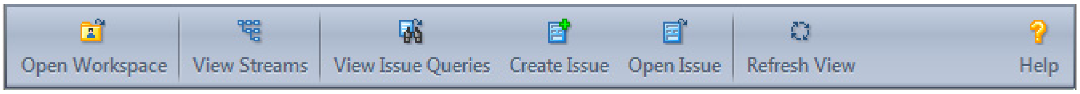
AccuRev deals with many kinds of named entities: files, directories, pathnames, depots, users, and so on. The table below details the restrictions on user-defined names for the various entities. Note that all names are case-sensitive; for example, user *john* is not the same as user *John*.

Note: In the table below, “characters” means NON-NULL characters. Do not subtract 1 to account for the terminating NUL.

Kind	Maximum Name Length	Characters Allowed in Name	Character Restrictions
depot stream snapshot workspace reference tree	79	A-Z,a-z,0-9 - _ . + @ *SPACE*	cannot begin with a digit or with "."
user group	99	same as above	must begin with a letter
password	19	any	none (caution: this means that special characters, such as *BACKSPACE* are valid password characters)
pathname of slice (of a depot or replica)	255	name validation performed by the operating system	applied by the operating system
pathname of workspace pathname of reference tree	127	name validation performed by the operating system	applied by the operating system
segment (between slashes) of element pathname	255	name validation performed by the operating system	applied by the operating system (Not to be confused with the "multi-segment" feature of AccuRev change packages.)

The GUI Main Toolbar

The main toolbar gives you quick access to commonly used features:



From left to right, the icons are:

Open Workspace

See: *Opening a Workspaces Tab* on page 32

View Streams

See: *Opening a StreamBrowser Tab* on page 145

View Issue Queries, ***Create Issue***, ***Open Issue***

See: *Creating AccuWork Queries* on page 309, and *Working with Issues* on page 287

Refresh

See: *Refreshing Tabs* on page 11

Help

See: *Overview of the On-Line Help* on page 1

The GUI Main Menu

The following sections describe the menus found along the toolbar of the GUI display.

The File Menu

- *The New Depot Command*
- *The New Workspace Command*
- *The New Stream and Change Stream Commands (Stream Configuration dialog)*
- *The Workspaces Tab / The Open Workspace Dialog*
- *The New Snapshot Command*
- *The Properties Command*

The New Depot Command

The ***New Depot*** command creates a new depot in the AccuRev repository.

Invoking the New Depot Command

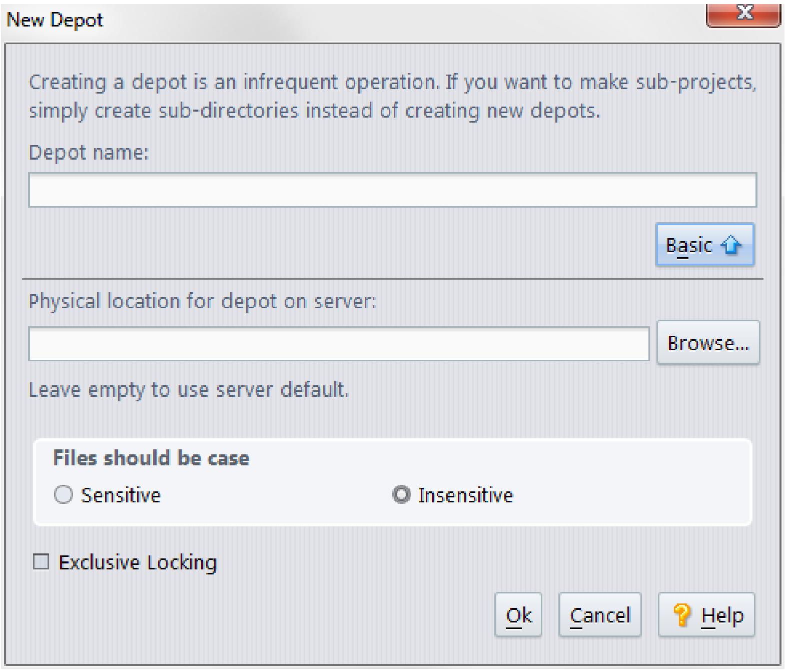
You can invoke the ***New Depot*** command in any of these ways:

- Choose ***File*** > ***New*** > ***Depot*** from the GUI main menu.

- Choose ***Admin*** > ***Depots*** from the GUI main menu to open a Depots tab (See *"The Depots Tab"* on page 46.). Then click the ***New*** button.

Using the New Depot Dialog

The ***New Depot*** dialog includes basic options (always visible) and advanced options (visibility controlled by a Basic/Advanced button).



Creating a depot is an infrequent operation. If you want to make sub-projects, simply create sub-directories instead of creating new depots.

Depot name:

Physical location for depot on server:

Leave empty to use server default.

Files should be case

○ Sensitive ● Insensitive

□ Exclusive Locking

Basic Options

Depot Name

No two depots in the same AccuRev repository can have the same name. See *User-Specified Names for AccuRev Entities* on page 23.

Note: Reusing a depot name

This can be done—but it's risky (it cannot be undone) and it takes some work. You must first remove the existing depot with the desired name from the repository. For details, see "Removing a Depot from the AccuRev Repository" in the *AccuRev Administrator's Guide*.

Advanced Options

Physical location ...

The location you choose here must be on a hard drive on the machine where the AccuRev Server process runs. Be sure that the location does not overlap the location of any other depot, or any AccuRev workspace or reference tree. See *User-Specified Names for AccuRev Entities* on page 23.

If you do not specify a physical location, the new depot is placed under directory *storage/depots* within the AccuRev installation area.

A depot's physical location is said to be a *slice* of the overall AccuRev repository . To change the location of an existing depot, use the CLI command **chslice**.

Files should be case ...

Note: We strongly recommend that you make your depots case-insensitive, for compatibility with Microsoft Windows. Some Windows applications can change the case of a filename without informing you; if your depot is case-sensitive, the file appears to be renamed.

All depots store names of files and directories exactly as they are originally entered (e.g. *WidgetGetCount* or *cmdListAll*). Likewise, AccuRev client programs display these names exactly as they were originally entered. The difference is that:

- A case-sensitive depot allows users to create two objects in the same directory, with names that differ only in their case (e.g. *makefile* and *Makefile*).
- A case-insensitive depot (default) does not allow two objects in the same directory to have names that differ only in their case.

A depot's case-sensitivity can be changed with the CLI **chdepot** command, as follows:

You can always change a depot's case from insensitive to sensitive.

You can only change from sensitive to insensitive if the depot contains no elements.

See *chdepot* in the *AccuRev CLI User's Guide* for more information.

Exclusive locking

Selecting the exclusive file locking option causes all workspaces created for this depot to use AccuRev exclusive file locking.

See *File Locking* on page 273.

The New Workspace Command

The ***New Workspace*** command creates a workspace, with a particular stream in a particular depot as its backing stream. The workspace can subsequently be reparented to another stream in the same depot.

Invoking the New Workspace Command

You can invoke the ***New Workspace*** command in a number of ways:

- Choose ***File*** > ***New*** > ***Workspace*** from the GUI main menu.
- In an ***Open Workspace*** dialog, click the ***New*** button.

Note: To open this dialog, select **File > Open Workspace** from the GUI main menu, or click the button on the GUI main toolbar.

- Choose ***New Workspace*** from the context menu of a stream or snapshot in the StreamBrowser.

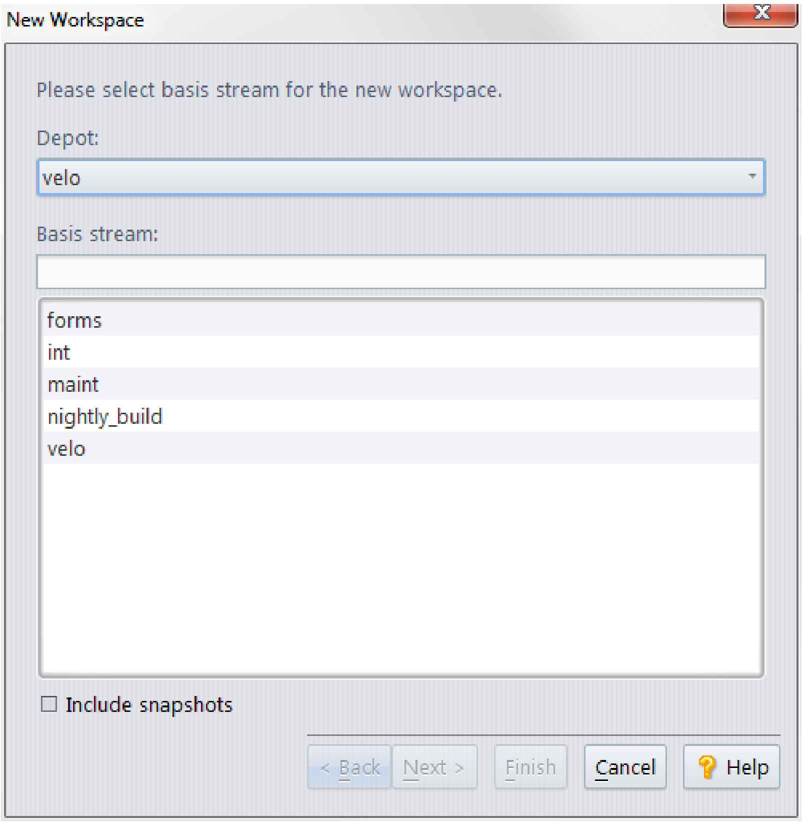
Using the New Workspace Wizard

AccuRev prompts you to specify the settings for the new workspace with a multiple-screen "wizard".

The New Workspace Command: Screen 1 of 3—Specifying the Depot and Backing Stream

Video Tip	Click *this link* to view an introductory video on creating a workspace.

With the ***File*** > ***New*** > ***Workspace*** command, there may be no "current stream" context. In this case, AccuRev starts with this screen, prompting you to specify a stream context:



New Workspace

Please select basis stream for the new workspace.

Depot:
velo

Basis stream:

forms
int
maint
nightly_build
velo

☐ Include snapshots

Depot

Select one of the repository's depots. This populates the ***Basis Stream*** listbox with the streams (and optionally, snapshots) of the specified depot.

Basis Stream

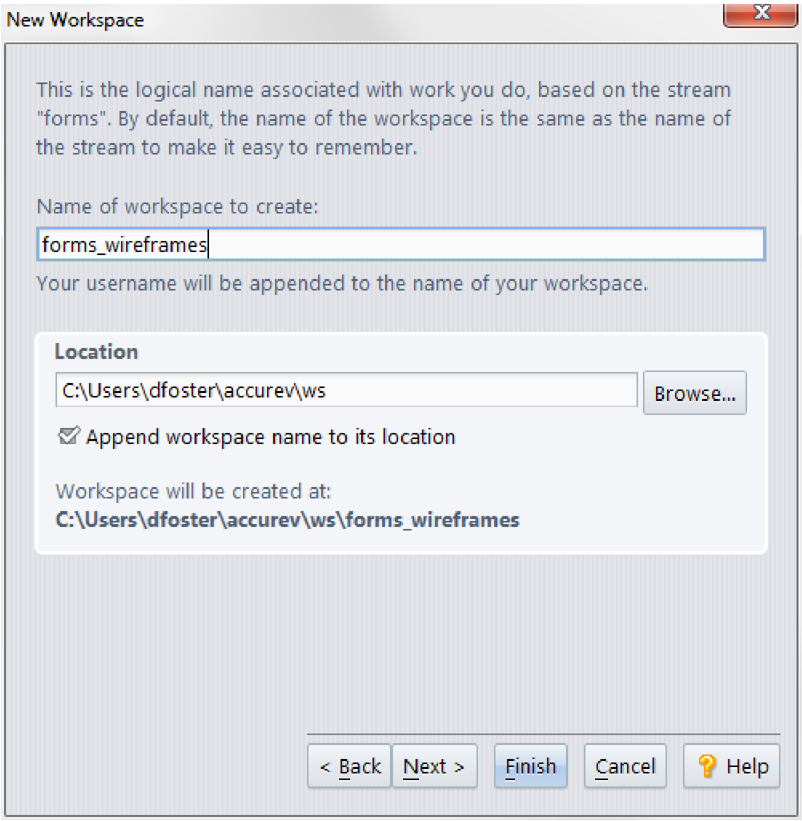
Select a stream or snapshot, to act as the backing stream for the new workspace.

Note: Is it advisable to base a workspace on a snapshot?

You can create new versions in a workspace based on a snapshot. But you won't be able to promote those versions to the snapshot (which is immutable, and so can't "accept" new versions). To get one or more of the versions out of the workspace, you can:

- Reparent the workspace to a dynamic stream, then promote the versions you've created in the workspace.
- Using the File Browser in another workspace, open one of the version tools on a particular element. Then use the *Send to Workspace* command to get the version that was created in the snapshot-based workspace. (See *"Send to Workspace"* on page 127.)

The New Workspace Command: Screen 2 of 3—Choosing a Name and Location for the Workspace)



New Workspace

This is the logical name associated with work you do, based on the stream "forms". By default, the name of the workspace is the same as the name of the stream to make it easy to remember.

Name of workspace to create:

forms_wireframes

Your username will be appended to the name of your workspace.

Location

C:\Users\dfoster\accurev\ws

☑ Append workspace name to its location

Workspace will be created at:
C:\Users\dfoster\accurev\ws\forms_wireframes

Name of Workspace to Create

No two workspaces in the same AccuRev repository can have the same name, even if they are in different depots. Moreover, streams, snapshots, workspaces, and reference trees all share the same namespace: no two of them can have the same name. See*User-Specified Names for AccuRev Entities* on page 23.

A workspace name always ends with a username suffix. For example, if your username is **akp** and you specify **thorn_dvt** as the workspace name, AccuRev creates a workspace named **thorn_dvt_akp**.

Notes:

Can I type in the username suffix myself?

Yes. If your username is **akp** and you enter a workspace name that ends with **_akp**, AccuRev doesn't change the name before creating the workspace.

Can I type in someone else's username suffix?

No, you cannot create a workspace for some other user in this way (or in any other way). If your username is **akp** and you enter the workspace name **thorn_dvt_john**, AccuRev creates a workspace named **thorn_dvt_john_akp**.

Location

You can specify any location in the client machine's file system for the new workspace tree. You can also create a workspace tree on a remote machine's disk, as long as that location is accessible on your machine. Access controls imposed by the operating system and/or network file system may restrict where you can create a workspace. See *User-Specified Names for AccuRev Entities* on page 23.

Here are some guidelines for using the ***Append workspace name to its location*** checkbox:

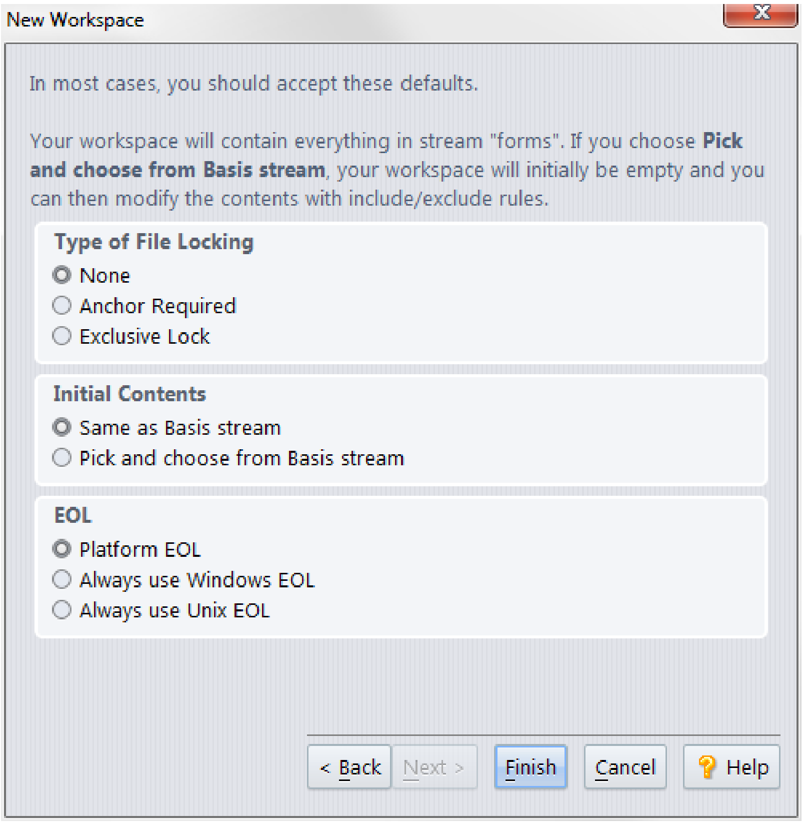
- If you are creating a new, empty workspace, enter the name of an existing directory (or ***Browse*** to it) and check the checkbox.
- If you are declaring that AccuRev should consider an existing directory to be a workspace, enter that directory's pathname and clear the checkbox.

In any case, make sure the ***Workspace will be created at*** value is correct before leaving this wizard screen.

Notes:

- Don’t create a workspace within the AccuRev installation area — typically, `C:\Program Files\AccuRev` (Windows) or `/opt/accurev` (UNIX/Linux).
- Don’t create a workspace within another workspace.
- Don't cross a physical file system boundary, whether local or remote. (For example: If you create a workspace at location `/opt/workspaces/derek`, an error occurs if a remote file system is mounted at `/opt/workspaces/derek/src` .)

The New Workspace Command: Screen 3 of 3—Configuring the Workspace)



New Workspace

In most cases, you should accept these defaults.

Your workspace will contain everything in stream "forms". If you choose **Pick and choose from Basis stream**, your workspace will initially be empty and you can then modify the contents with include/exclude rules.

Type of File Locking

● None
○ Anchor Required
○ Exclusive Lock

Initial Contents

● Same as Basis stream
○ Pick and choose from Basis stream

EOL

● Platform EOL
○ Always use Windows EOL
○ Always use Unix EOL

Type of File Locking

This setting controls whether development in this workspace uses serial or parallel methodology. If you choose parallel development, you may still be restricted to using serial development on certain files; AccuRev supports locks on individual file elements. (See *File Locking* on page 273.)

None (parallel development): The version-controlled files in the workspace tree are writable at any time. There is no need to perform an **anchor** operation on a file before beginning to work on it.

Anchor Required: Each version-controlled file in the workspace tree is read-only until you use the ***Anchor*** command to make it writable. This also makes the file *active* in the workspace.

Note: The Anchor command makes the version currently in the workspace.

Exclusive Lock: Like ***Anchor Required***, and also causes AccuRev to prevent users in sibling workspaces from working on the same file concurrently. (See *File Locking* on page 273.)

Initial Contents

Same as Basis stream For each element, AccuRev copies the current version in the backing stream to the workspace tree. If no version of an element currently appears in the backing stream, it won't appear in the new workspace, either.

Pick and choose from Basis stream: AccuRev doesn't copy any files to the workspace tree. To begin working in the workspace, use the File Browser's Include/Exclude facility to configure the set of elements to appear in the workspace. (*Including and Excluding Files* on page 103.)

EOL (Text-file line terminators)

Whenever AccuRev copies a version of a text file from the repository to the workspace tree, it uses the line terminator you configure here.

Platform EOL: Use the line terminator for the operating system on the machine where the AccuRev GUI is running.

Always use Windows EOL: Use the sequence CR-LF (hex 0C-0A) as the line terminator.

Always use UNIX EOL: Use NL (hex 0A) as the line terminator.

The New Stream and Change Stream Commands (Stream Configuration dialog)

The ***New Stream*** and ***Change Stream*** commands open a dialog in which you specify/revise the configuration of a new/existing stream.

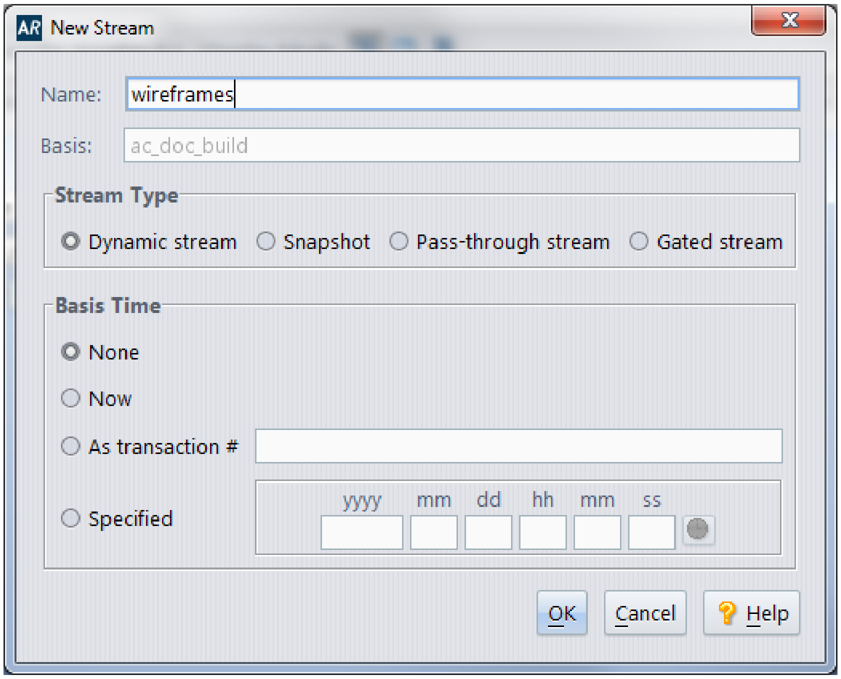
Invoking the Command

In the StreamBrowser, right-click a stream to select it and display its context menu. Then,

- Choose ***New Stream*** from the context menu to create a new stream below the selected stream.
- Choose ***Change Stream*** from the context menu to revise the configuration of the selected stream.

Using the New Stream / Change Stream Dialog

Enter or revise the following specifications, then click ***Ok***.



Name: wireframes

Basis: ac_doc_build

Stream Type

● Dynamic stream ○ Snapshot ○ Pass-through stream ○ Gated stream

Basis Time

● None

○ Now

○ As transaction #

○ Specified

Name

No two streams in the same AccuRev repository can have the same name, even if they are in different depots. See *User-Specified Names for AccuRev Entities* on page 23.

Basis

The parent stream of the stream you are reconfiguring. Changing this entry reparents the stream when you click ***Ok***.

In a ***New Stream*** dialog, this field is not editable, because you selected the parent stream when you invoked the command.

Stream Type

The type of the stream:

- **Dynamic stream** -- A stream whose configuration changes over time, with new versions promoted from child workspaces and/or from other dynamic streams. It also inherits versions from its parent stream.

- **Snapshot** -- An immutable ("frozen", "static") stream that captures the configuration of another stream at a particular time. A snapshot cannot be renamed or modified in any way. Selecting Snapshot is equivalent to invoking the command ***New Snapshot*** instead of ***New Stream***.

- **Pass-through stream** -- A stream that is useful for grouping lower-level streams. A version that is promoted to a pass-through stream automatically *passes through* to the parent stream. The file does not become active in the pass-through stream; it does become active in the parent of the pass-through stream.

- **Gated stream** -- A stream that uses user-defined rules to control whether changes promoted into it are promoted out of it. These rules are often based on meeting externally established criteria (passing a test suite or a successful nightly build, for example).

Note: The **Stream Type** field is not editable in the ***Change Stream*** dialog. You cannot change the type of an existing stream.

Basis Time

By default, a stream inherits the versions currently in its parent stream. If you assign a basis time to a stream, it inherits the versions that were in the parent stream at a specified point in time. Basis time is not applicable to pass-through and gated streams.

- **None** -- (not applicable to snapshot streams) No basis time; in the future, the stream will inherit the versions currently in its parent stream.
- **Now** -- The stream or snapshot will contain the versions that were in its parent stream at the time you clicked ***Ok*** in this dialog.
- **As transaction #** -- Enter an integer transaction number. The stream or snapshot will contain the versions that were in its parent stream at the time the specified transaction was created.
- **Specified** -- Enter a specific time. The stream or snapshot will contain the versions that were in its parent stream at the specified time.

You can fill in the individual subfields manually, or use the *Select Date* button (🕒) to display choices ("2 days ago", etc.) that fill in the subfields automatically. Once these subfields are filled in, you can revise them individually or by clicking the *Select Date* button again.

Note:

1. *Dynamic stream with basis time vs. Snapshot*

Setting a basis time on a stream does make it like a snapshot of its parent stream -- initially. But after that, new versions can be promoted to such a "time-based stream" from child workspaces and from other streams; you cannot promote new versions to a snapshot. Since you can change the basis time of a time-based stream as often as you wish, it can serve as a "moving snapshot" of its parent stream.

2. *Another way to control which versions are inherited from the parent stream*

The basis-time facility includes versions based on when they were created. The include/exclude facility includes elements based on their pathnames.

The Workspaces Tab / The Open Workspace Dialog

The Workspaces tab displays summary information on some or all of the workspaces in the AccuRev repository. In addition, it lets you create, edit, and remove a workspace.

The Open Workspace dialog, a slight variant of the Workspaces tab, makes it easy to open a File Browser tab on a selected workspace.

Opening a Workspaces Tab

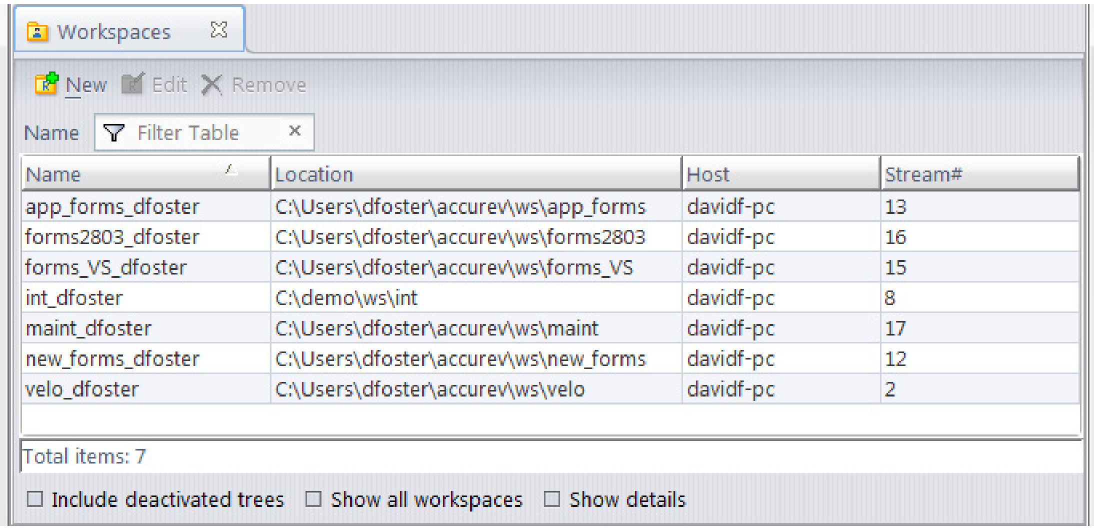
Choose ***View*** > ***Workspaces*** from the GUI main menu.

Opening the Open Workspace Dialog Box

Choose ***File*** > ***Open Workspace*** from the GUI main menu, or click the **Open Workspace** button in the GUI main toolbar.

Workspaces Tab Layout

The Workspaces tab contains a table with these columns:



Name	Location	Host	Stream#
app_forms_dfoster	C:\Users\dfoster\accurev\ws\app_forms	davidf-pc	13
forms2803_dfoster	C:\Users\dfoster\accurev\ws\forms2803	davidf-pc	16
forms_VS_dfoster	C:\Users\dfoster\accurev\ws\forms_VS	davidf-pc	15
int_dfoster	C:\demo\ws\int	davidf-pc	8
maint_dfoster	C:\Users\dfoster\accurev\ws\maint	davidf-pc	17
new_forms_dfoster	C:\Users\dfoster\accurev\ws\new_forms	davidf-pc	12
velo_dfoster	C:\Users\dfoster\accurev\ws\velo	davidf-pc	2

Total items: 7

□ Include deactivated trees □ Show all workspaces □ Show details

User

(appears if ***Show all workspaces*** is checked) The user who owns the workspace.

Status

(appears if ***Include deactivated trees*** is checked) **Active** indicates that the workspace is available for use. **Inactive** indicates that the workspace is currently deactivated. Inactive workspaces can be reactivated. See the Remove and Reactivate commands below.

Name

The simple name of the workspace, including the suffix that indicates the user who owns the workspace.

Location

The pathname of the workspace tree on the machine where it resides.

Host

The name or IP address of the machine where the workspace tree resides.

Stream#

The integer stream-ID of the workspace. This number is unique with the workspace's depot only, not across the entire repository.

Note: More precisely, this integer identifies the private workspace stream that is an integral part of the workspace. A depot's streams, snapshots, and workspaces share the same "pool" of stream-IDs.

Target Transaction

(appears if ***Show details*** is checked) The depot's transaction level at the time the ***Update*** command was most recently invoked on the workspace. ***Update*** attempts to load versions created in transactions up to and including the target transaction. If the **Target Transaction** and **Update Level** are the same, the workspace's most recent update completed successfully.

Update Level

(appears if ***Show details*** is checked) The highest-numbered transaction whose versions have been copied to the workspace in an update.

Type

(appears if ***Show details*** is checked) An integer that indicates the type of workspace stream: 1 (standard workspace), 9 (exclusive-file-locking), or 17 (anchor-required).

EOL

(appears if ***Show details*** is checked) The type of line terminator used when an update copies text-file versions to the workspace tree: **Auto** (line terminator used by the host machine's OS), **UNIX** (NL), or **Windows** (CR-LF).

Working in a Workspaces Tab

You can apply filters to control the amount of data displayed in the workspaces table, and you can perform a number of operations on a selected workspace.

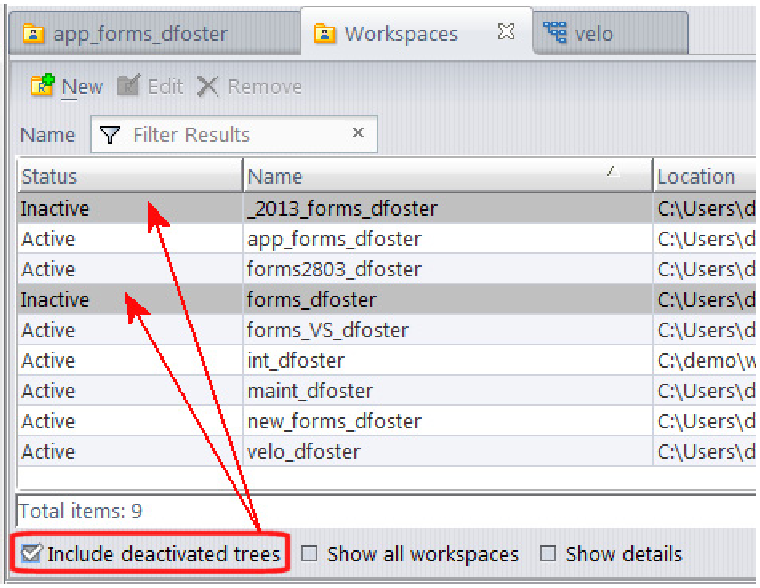
See also:

Working with Tables on page 14.

Controlling Which Workspaces are Displayed

Include deactivated trees

If checked, the table includes workspaces that have been deactivated with the Remove command. This also adds a Status column; deactivated workspaces have Inactive status, and are shaded.



Status	Name	Location
Inactive	_2013_forms_dfoster	C:\Users\d
Active	app_forms_dfoster	C:\Users\d
Active	forms2803_dfoster	C:\Users\d
Inactive	forms_dfoster	C:\Users\d
Active	forms_VS_dfoster	C:\Users\d
Active	int_dfoster	C:\demo\w
Active	maint_dfoster	C:\Users\d
Active	new_forms_dfoster	C:\Users\d
Active	velo_dfoster	C:\Users\d

Total items: 9

☑ Include deactivated trees □ Show all workspaces □ Show details

Show all workspaces

If checked, the table includes workspaces created by all users. This also adds a User column.

If cleared, the table includes your workspaces only.

Controlling the Data Displayed for Each Workspace

Show details

If checked, the table includes these columns: **Target Transaction**, **Update Level**, **Type**, and **EOL.**

Operating on a Selected Workspace

You can choose any of the following commands from the context menu of a selected workspace:

Open (Workspaces tab)

Ok (Open Workspace dialog)

Using a File Browser tab, display the entire contents of the workspace.

Edit

Modify the configuration of the workspace.

Reactivate

Bring back into active service a workspace that was previously deactivated with the ***Remove*** command.

✗Remove

Deactivate a workspace. The workspace cannot be used to perform AccuRev commands, although the workspace tree is not changed in any way. The workspace no longer appears in the StreamBrowser display (unless you use the StreamBrowser's ***Include deactivated trees*** option).

You can use the ***Reactivate*** command to bring back a ***Remove***'d workspace.

Creating a New Workspace

Click the ***New*** button at the bottom of the Workspaces tab to invoke the New Workspace wizard (see *The New Workspace Command* on page 26).

The New Snapshot Command

Video Tip	Click *this link* to view an introductory video on creating snapshots.

The ***New Snapshot*** command opens a dialog in which you define a snapshot of the stream you selected.

Invoking the New Snapshot Command

In the StreamBrowser, right-click a stream and choose ***New Snapshot*** from the context menu.

Note: There is no "Edit Snapshot" command, because a snapshot cannot be modified in any way.

Using the New Snapshot Dialog

The ***New Snapshot*** dialog is identical to the ***New Stream*** dialog (See *"Using the New Stream / Change Stream Dialog"* on page 30.), except that:

- The stream type ***Snapshot*** is preselected.
- You must specify a basis time (the ***None*** radio button is disabled).

The Properties Command

The ***Properties*** command displays a message box with information about the selected element.

Invoking the Properties Command

Choose ***Properties*** from the context menu of an element in the Details pane of the File Browser.

The Properties Display

File Path

The depot-relative pathname of the element in the current stream or workspace.

File Type

The element type of this version.

File EID

The integer that identifies this element. An element's simple name or pathname can change, but not its element-ID. No other element -- past, present, or future -- in the same depot can have this element-ID.

Version

The version-ID of this version.

File Status

The current status of this version.

Target Path

(element link or symbolic link) The depot-relative pathname of the link's target.

The Issues Menu

See *Creating AccuWork Queries* on page 309.

See *Working with Issues* on page 287.

The Tools Menu

- *The Synchronize Time Command*
- *The Show Info Dialog*
- *The Server Tasks Tab*

The Synchronize Time Command

The Synchronize Time command changes the system clock on the client machine to match the clock on the AccuRev Server machine. This succeeds only if the operating system grants you the right to make this change.

The Show Info Dialog

The Show Info dialog displays overall information about your AccuRev work environment.

Opening a Show Info Dialog

Choose ***Tools*** > ***Info*** from the GUI main menu.

Show Info Tab Layout

Principal

Your AccuRev user identity, as established by AccuRev's ***Login*** command.

Host

The name or IP address of the client machine on which the AccuRev GUI is running.

Server name

The name or IP address of the server machine on which the AccuRev Server process is running. You can switch servers as part of the login process.

Port

The port number on which the AccuRev Server listens to client requests.

ACCUREV_BIN

The directory on the client machine where AccuRev executables are stored. In older versions of AccuRev, you sometimes needed to create an environment variable with this name. This is no longer necessary.

Client time

The system time on the client machine. (The parenthesized integer is the number of seconds since Jan 1, 1970.)

Server time

The system time on the server machine. (The parenthesized integer is the number of seconds since Jan 1, 1970.)

If the client time and server time differ by more than 5 seconds, many AccuRev commands fail to execute. Use the ***Tools*** > ***Synchronize Time*** command to fix this problem.

The Server Tasks Tab

The Server Tasks tab lists the current set of "worker threads" in the multi-threaded AccuRev Server process. At any given moment, each worker thread is executing one of the server **subtasks** that implement a particular client command. Each command from a client program translates to at least two server subtasks.

Opening a Server Tasks Tab

Choose ***Tools*** > ***Server Tasks*** from the GUI main menu.

Server Tasks Tab Layout

The Server Tasks tab contains a table with the following columns:

Task ID

The ID number of the client-server connection. Each client command establishes a separate connection with the server.

Task Name

The name of the server subtask being executed by this worker thread. This typically matches the name of the client command, such as Update.

User

The AccuRev username of the user who issued the client command.

Host

The IP address of the client machine where the user invoked the client command.

Status

0 = slot assigned to worker thread, but thread not active yet (set by master thread).

1 = worker thread active (set by worker thread).

2 = worker thread completed, but master thread has not yet cleared the slot for reuse (set by worker thread).

Start Time

The time at which the worker thread started working on this subtask.

Seconds on Server

The number of seconds ("wall clock" time) that this worker thread has been running.

Note: Command-line interface to this command

You can use the AccuRev CLI to list the Server Tasks tab data:

1. Create a text file, for example *tasks.xml*, with this one-line XML document:

   ```
   <tasklist/>
   ```
2. Execute this AccuRev CLI command:

   ```
   accurev xml -l tasks.xml
   ```

Working in the Server Tasks Tab

The Server Tasks tab display is informational only. See also: See *“Working with Tables”* on page 14.

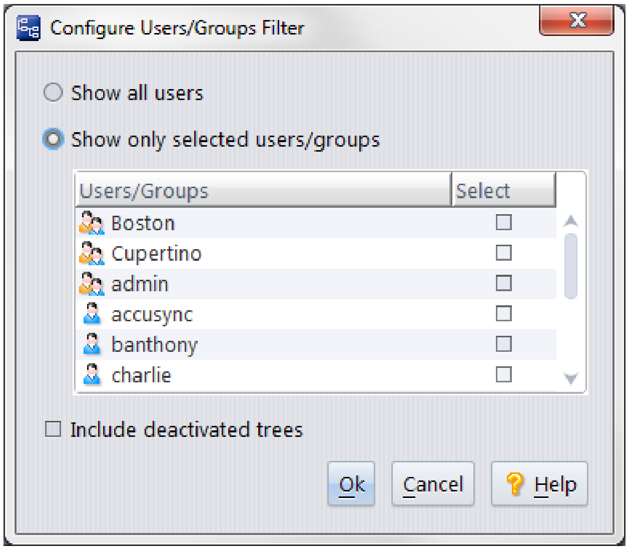
Filter Users/Groups (View Users/Groups Command)

Invoking the Filter Users/Groups Command

You invoke this command choosing ***View*** > ***Users/Groups*** from the GUI main menu.

Using the Dialog

The ***View*** > ***Users/Groups*** command allows you to restrict the set of users and groups displayed in certain places in the AccuRev GUI.



Configure Users/Groups Filter

○ Show all users

● Show only selected users/groups

Users/Groups	Select
Boston	□
Cupertino	□
admin	□
accusync	□
banthony	□
charlie	□

□ Include deactivated trees

When you invoke the command, a dialog shows a list of all the users and groups you have defined within AccuRev. You can:

- Click the *Show all users* radio button to remove any restrictions on the display of users.
- Click the *Show only selected groups/users* radio button to enable filtering. Then click the checkbox in the Select column for each user or group you would like to display. When you have finished your selections, click the ***Ok*** button to enable the filter.

Use the *Include deactivated trees* checkbox to include users that have been deactivated with the ***Remove*** command in the list shown in this dialog. These users are displayed with a gray background.

Scope of the Users/Groups Filter

The users and groups you specify will appear in lists of users in the following locations:

- ***Workspaces drop-down list*** in the StreamBrowser.
- All drop-down lists in AccuWork that contain user or group data (such as *Assigned to* or *Submitted by*), and throughout the workflow.
- ***Add to Stream Filter*** dialog.
- ***Send to Issue d***ialog (when change-package-level integration is set up between AccuRev and AccuWork).

AccuRev Preferences (Tools > Preferences Command)

The *Tools* > *Preferences* command opens a tabbed dialog, in which you can manage a set of parameters that influence the way the AccuRev looks and works. The preferences you set here apply only to you. They are stored in file *preferences.xml*, in subdirectory **.accurev** of your home directory. (By default, AccuRev determines your home directory by making a call to the operating system. If environment variable ACCUREV_HOME is set, AccuRev uses the value of this variable as your home directory.)

General Page

Enable Issue Preview

Controls the automatic inclusion of an Edit Form pane below the Query Results pane in a AccuWork Queries tab. The contents of the issue record that is currently selected in the Query Results pane is automatically displayed in the Edit Form pane.

The toolbar of the Query Results pane includes a *Show Issue Details* button, which toggles the inclusion of the Edit Form pane.

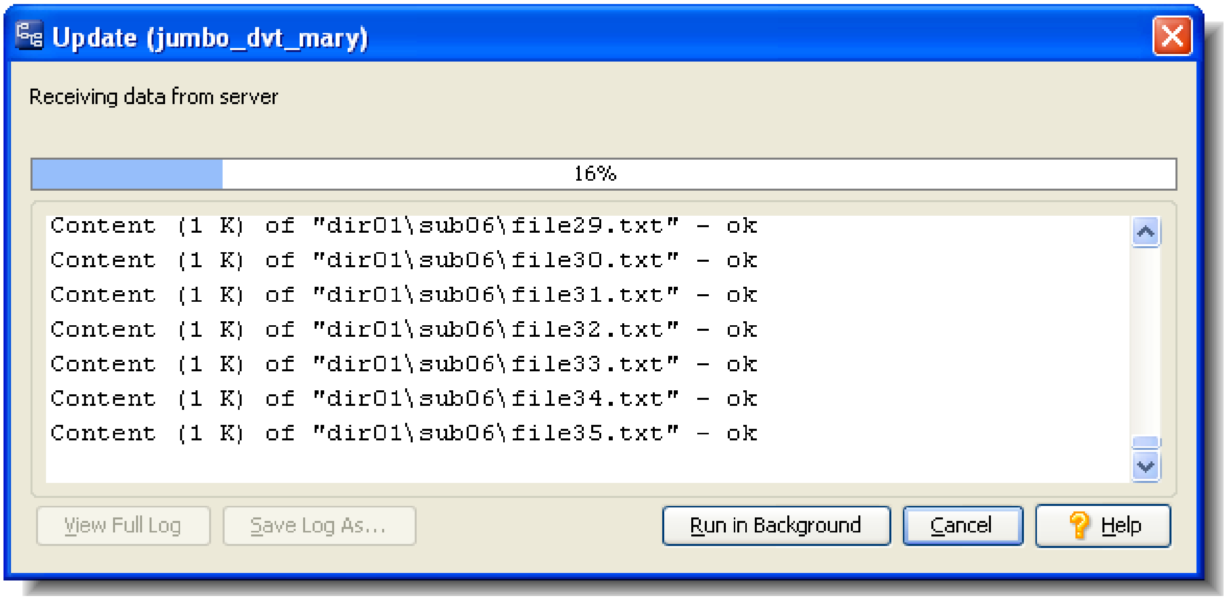
Despite the name "Preview", the Edit Form pane is fully functional: you can revise, save, and export the contents of issue records in this pane.

Note: This option will be disabled if your AccuRev GUI is configured to display issues in the AccuRev Web UI. This will be the case if your AccuRev Administrator has configured your site to use the Web UI for AccuWork.

Show Progress Log

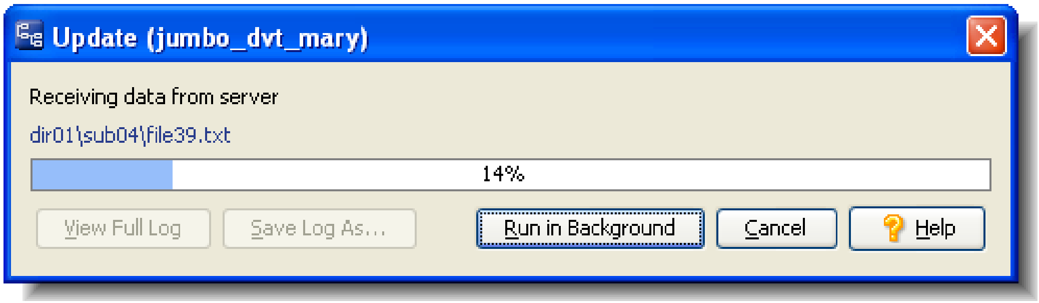
Controls whether the *Update* or *Update Preview* dialog includes a scrollable text field that displays the names of up to 2000 elements that are being updated (or would be updated).

If this option is checked, the dialog looks like this:



```
Content (1 K) of "dir01\sub06\file29.txt" - ok
Content (1 K) of "dir01\sub06\file30.txt" - ok
Content (1 K) of "dir01\sub06\file31.txt" - ok
Content (1 K) of "dir01\sub06\file32.txt" - ok
Content (1 K) of "dir01\sub06\file33.txt" - ok
Content (1 K) of "dir01\sub06\file34.txt" - ok
Content (1 K) of "dir01\sub06\file35.txt" - ok
```

If this option is cleared, the dialog looks like this:



After the command finishes its work, a complete log is always available, through the *View Full Log* button in the dialog.

Show External Objects

Controls whether the File Browser's Details pane displays files and directories with a status of **(external)**. External objects are files and directories that have not been placed under version control (using the add command, for example). This setting can be used with the *Include Ignored Objects* preference. See *Controlling the Display of External Objects* on page 58 for more information on using these preferences.

Tip: If you change this setting, use the *View* > *Refresh* command (or function key **F5**) to refresh the data.

Include Ignored Objects

Controls whether ignored objects are displayed in the File Browser's Details pane. An ignored object is an external file or directory that matches a pattern specified using the **.acignore** file. Objects matching an ignore pattern are displayed with a status of **(ignored)**. This preference is available only if the Show External Elements preference is also selected. See *Controlling the Display of External Objects* on page 58 for more information on using this preference.

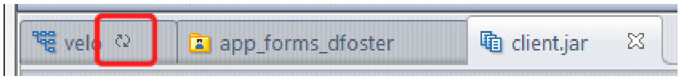
See *Chapter 3 Specifying Ignore Patterns for External Objects* in *AccuRev Technical Notes* for more information on specifying ignore patterns.

Confirm On Exit

Controls whether the AccuRev GUI prompts for confirmation when you invoke the *File* > *Exit* command or attempt to shut the GUI window.

Require Manual Refresh

If this preference is cleared, the data displayed on a "buried" GUI tab is automatically refreshed, if necessary, when you switch to that tab. The AccuRev GUI signals that a refresh is necessary with a "refresh" icon next to the tab title.



If this preference is checked, the data on a "buried" tab is not refreshed automatically when you switch to it. Use the *View* > *Refresh* command (or function key **F5**) to refresh the data.

Keep Session Active

Controls whether the session file in the **.accurev** subdirectory of your home directory will be deleted when you end the AccuRev GUI session. This file is created by the ***Tools > Login*** command, and is used by other AccuRev commands to establish your identity (and the identity of your client machine).

See *The "session" File* on page 7 for more information.

Update Resolves Trivial Merges

Enables the ***Update*** command to perform a merge operation on text file elements that are not in a workspace's default group, but have **(overlap)(modified)** status. The merges are performed and the ***Update*** proceeds only if for *all* such files, the merge is completely automatic (there are no conflicting changes).

Display of element name in tables

Controls how element pathnames are displayed in tables: in a single *Element* column, or in separate *Name* and *In Folder* columns.

Dialog Confirmations

Controls whether or not AccuRev displays confirmation messages when you:

- Close all other tabs open in the AccuRev GUI (by right-clicking on a tab and choose Close Others, for example)
- Click the **Show All Streams** button in the StreamBrowser, which is an operation that can take a long time depending on the size and complexity of your stream hierarchy

Tab Layout

When you open more tabs than can be displayed in the AccuRev window, a scroll button appears to provide access to tabs that are not currently visible. If you prefer, you can have AccuRev display all tabs at the same time; in this case, tabs are displayed in as many rows as needed.

Options are:

- One row (the default)
- Multiple rows

Tip: Press Ctrl+F6 to display a dialog box that lists all open tabs.

Online Help Browser

The full pathname of the web browser to be used for displaying the GUI’s context-sensitive help screens. This field also determines what web browser to use if your client is configured to work with the AccuRev Web User Interface (Web UI).

Diff/Merge Page

The **Diff/Merge** tab allows you to specify supported third-party tools to use with AccuRev. **Note:** Any tools that you specify must be defined in your PATH variable before launching the AccuRev GUI. If you specify a tool and it does not work, this is the most common cause.

Diff

Specifies the graphical file-comparison tool to be invoked by the GUI's *Diff* command. You can choose AccuRev's own tool or one of the supported third-party tools from the combo-box. Alternatively, you can type a command line to be executed when the *Diff* command is invoked. This command line must include substitution patterns:

`%1%` Quoted pathname of the first version to be compared.

`%2%` Quoted pathname of the second version to be compared.

`%3%` Unquoted title string for the first version (can be displayed by file-comparison tool).

`%4%` Unquoted title string for the second version (can be displayed by file-comparison tool).

Make sure the file-comparison tool is located in a directory on your search path. Alternatively, use a full pathname to specify the file-comparison tool. Examples:

```
MyDiff /G %1% %2%
```

```
C:\SuperCompare\SCgui %1% %2%
```

Notes:

*What's the difference between **Ignore Whitespace** and **Ignore Changes in Whitespace**?*

Ignore Whitespace causes all SPACEs, TABs, and empty lines to be ignored, and thus overrides any setting of **Ignore Changes in Whitespace**.

Ignore Changes in Whitespace applies only where whitespace already exists; it does not apply where new whitespace is introduced or existing whitespace has been completely deleted.

Example 1: When comparing these two text strings:

```
compareme
compare me
```

Table 1.

[ ] Ignore Whitespace	[ ] Ignore Changes in Whitespace	Does Diff see difference?
X	either	No
-	X	**Yes**
-	-	Yes

Example 2: When comparing these two text strings:

```
compare     me
compare me
```

Table 2.

[ ] Ignore Whitespace	[ ] Ignore Changes in Whitespace	Does Diff see difference?
X	either	No
-	X	**No**
-	-	Yes

I changed my preferences, but I don't see any changes in the Diff tab.

Changing preferences does not automatically affect Diff tabs that are currently open. To apply the preferences, refresh the tab in any of these ways:

- Click the Refresh button on the GUI window's main toolbar.
- Run the *View* > *Refresh* command on the GUI window's main menu.
- Press function key **F5**.

Merge

Specifies the graphical text-file merge tool by be invoked by the GUI's *Merge* command. This command line must include substitution patterns:

`%a%` quoted filename of the closest common ancestor version.

`%1%` quoted filename of the version in the backing stream, or other non-workspace version.

`%2%` quoted filename of the version in the workspace.

`%3%` unquoted title string for the closest common ancestor version (can be displayed by text-file-merge tool).

`%4%` unquoted title string for the backing-stream version (can be displayed by text-file-merge tool).

`%5%` unquoted title string for the workspace version (can be displayed by text-file-merge tool).

`%o%` quoted name of the merge-output file -- a temporary file for storing the results of the merge.

Make sure the text-file-merge tool is located in a directory on your search path. Alternatively, use a full pathname to specify the text-file-merge tool.

Note: Do I have to use a 3-way merge tool?

No. You're free to use a 2-way merge tool, which takes into account the two contributor versions only, not any of their ancestors.

But even if you don't use a 3-way merge tool, AccuRev records the merge operation (which appears as a red line in the Version Browser). Future merges involving this element can use this record to simplify the work needed to perform a 3-way merge.

Tab size

(AccuRev's graphical Diff and Merge tools only) The number of spaces to be displayed for each *TAB* character.

Character Encoding

(AccuRev's graphical Diff and Merge tools only) Specifies the character encoding of the contributor versions: either UTF-8 or the operating system default.

Ignore Whitespace

(AccuRev's graphical Diff tool only) Controls whether whitespace is taken into account when comparing text lines.

Ignore Changes in Whitespace

(AccuRev's graphical Diff tool only) Controls whether a change in the amount of whitespace in a text line is considered to be a change to that line.

Ignore Case

(AccuRev's graphical Diff and Merge tools only) Controls whether uppercase and lowercase characters are considered to be the same when comparing text lines.

Version Browser Page

Initial transaction count

Controls the amount of information initially displayed in the Version Browser. The default value displays only the 10 most recent transactions. You can change the default value to 20, 50, 100, or "Everything". Once the Version Browser is displayed, you can control the amount of information on the screen with the **Show:** and **Filter Workspaces:** controls.

Scroll to highlighted issues

When this option is selected, the upper pane of the Version Browser will scroll to the most recent transaction for the issue that you select in the Issues tab of the lower pane. Enabled by default. Takes effect immediately.

Scroll to highlighted workspaces in groups

When this option is selected, the upper pane of the Version Browser will scroll to the most recent transaction for the workspace (or stream) that is selected. Whether or not this option is checked, the selected workspace (or stream) will be highlighted. Disabled by default. Takes effect immediately.

Limit Streams: only show a subset of the streams

Check this setting to control the amount of information displayed in the Version Browser, by showing only up to the five most recent streams and, if applicable, the streams that are in the hierarchy of the selected transaction. Takes effect upon refresh, or upon changing the transaction count.

StreamBrowser Page

Display Default Group in Stream Browser

Controls whether the StreamBrowser displays an active file, transaction, or issue icon (or or) for each stream and workspace with active elements. Disabling this feature can significantly improve StreamBrowser performance.

Enable StreamBrowser History

Controls the inclusion in the StreamBrowser toolbar of controls that enable you to view a depot's stream hierarchy as it existed at any point in the past. You must refresh any existing StreamBrowser tabs to make a new setting effective.

Screen Layout

Gives you control over the display of the StreamBrowser. The options include:

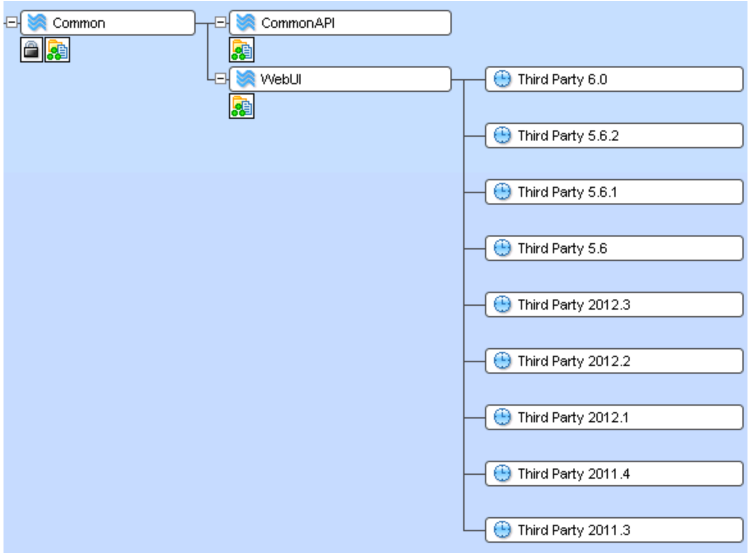
Horizontal: Child streams fan out to the right of their parents, and the stream boxes are aligned in columns, with the size of the stream boxes determined by the maximum name size within that column. For example, a stream named **ac_5.5_build** is as wide as a stream named **ac_5.5_dev_workflow_itr4_doc_dfoster**. This setting provides a uniform presentation that makes viewing the stream hierarchy easy, but it also requires a lot of horizontal space.

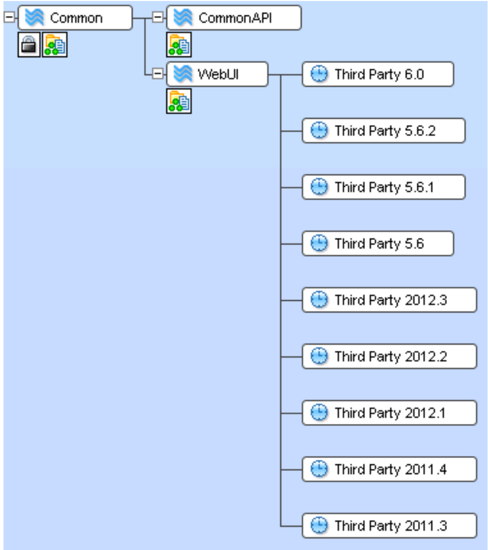
Fit Stream Width to Stream Name: This preference is optional only for horizontal displays. It sizes each stream box to be only as wide as its name. This makes the stream hierarchy display more compact, with more streams visible at a time.

Vertical: Child streams appear underneath their parents, and indented. The stream boxes are always sized to be the width of the stream name.

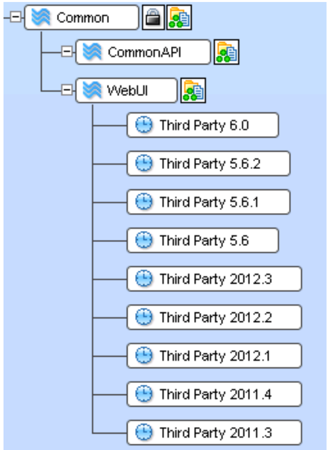
Examples of these display modes are shown below:

Horizontal, default (left), sized by name (right):





Vertical:



The Admin Menu (Administrators Only)

The options on the **Admin** menu are primarily meant for administrators rather than end-users. The ability to use administrative functions is controlled by a customized server trigger script called server_admin_trig.pl. A sample script with documentation is found on the server in *<ac_install>*/examples.

If you do not have administrative privileges, you are able to view these options, but will not be allowed to modify or save data.

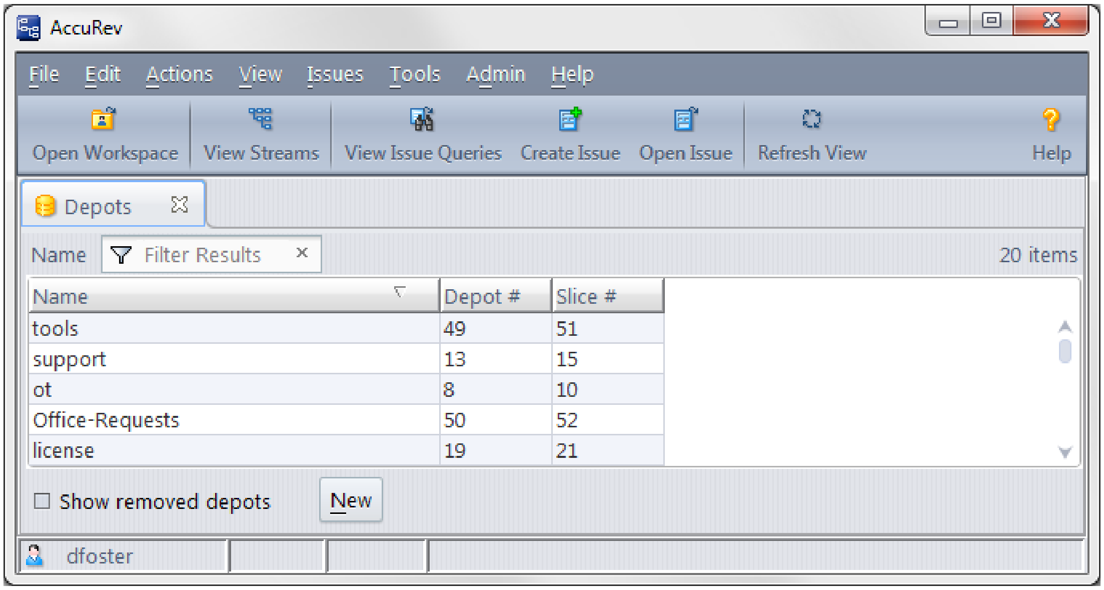
The Depots Tab

The Depots tab lists some or all of the depots in the AccuRev repository.

Opening a Depots Tab

Choose ***Admin*** > ***Depots*** from the GUI main menu.

Depots Tab Layout



Name Filter Results 20 items

Name	Depot #	Slice #
tools	49	51
support	13	15
ot	8	10
Office-Requests	50	52
license	19	21

□ Show removed depots   New

dfoster

Status

(appears if ***Show removed depots*** is checked) **Active** indicates that the depot is available for use. **Removed** indicates that the depot has been permanently removed from the repository. (The **maintain rmdepot** command permanently and irreversibly removes a depot from the repository. See "Removing a Depot from the AccuRev Repository" in the AccuRev *Administrator's Guide*.)

Depot

The unique integer that identifies a particular depot. This identifier cannot be modified.

Name

The depot's name, which must differ from all other depot names in the repository. A depot can be renamed, but its depot number never changes.

Slice

Same as Depot#.

Working in a Depots Tab

You can control the display of removed depots.

See also: *Working with Tables* on page 14

Controlling Which Depots are Displayed

Show removed depots

If checked, the table includes the depots that have been permanently deactivated with the **maintain** utility (see the AccuRev *Administrator's Guide*). This also adds the ***Status*** column to the table; removed depots have **Inactive** status, and are shaded.

Creating a New Depot

New

Click this button to invoke the *The New Depot Command* on page 24.

The Reference Trees Tab

The Reference Trees tab displays summary information on some or all of the reference trees in the AccuRev repository.

Opening a Reference Trees Tab

Choose ***Admin*** > ***Reference Trees*** from the GUI main menu.

Reference Trees Tab Layout

The Reference Trees tab contains a table with these columns:

Status

(appears if ***Include deactivated trees*** is checked) **Active** indicates that the reference tree is available for use. **Inactive** indicates that the reference tree is currently deactivated (but can be reactivated).

Name

The name of the reference tree.

Location

The pathname of the reference tree on the machine where it resides.

Host

The name or IP address of the machine where the reference tree resides.

Stream#

The integer stream-ID of the stream on which the reference tree is based. This number is unique with the reference tree's depot only, not across the entire repository.

Target Transaction

(appears if ***Show details*** is checked) The depot's transaction level at the time the ***Update*** command was most recently invoked on the reference tree. ***Update*** attempts to load versions created in transactions up to and including the target transaction. If the **Target Transaction** and **Update Level** are the same, the reference tree's most recent update completed successfully.

Update Level

(appears if ***Show details*** is checked) The highest-numbered transaction whose versions have been copied to the reference tree in an update.

Type

(appears if ***Show details*** is checked) The integer 3, indicating that the type of data structure is a reference tree.

EOL

(appears if ***Show details*** is checked) The type of line terminator used when an update copies text-file versions to the reference tree: **Auto** (line terminator used by the host machine's OS), **UNIX** (NL), or **Windows** (CR-LF).

Working in a Reference Trees Tab

You can apply filters to control the amount of data displayed in the reference trees table, and you can perform a number of operations on a selected reference tree. See also: *Working with Tables* on page 14

Controlling Which Reference Trees are Displayed

Include deactivated trees

If checked, the table includes reference trees that have been deactivated with the ***Remove*** command. This also adds a ***Status*** column; deactivated reference trees have **Inactive** status, and are shaded.

Controlling the Data Displayed for Each Reference Tree

Show details

If checked, the table includes these columns: **Target Transaction**, **Update Level**, **Type**, and **EOL.**

Operating on a Selected Reference Tree

You can choose any of the following commands from the context menu of a selected reference tree:

Open

In this release, the ***Open*** command is disabled. You cannot view the contents of a reference tree using the AccuRev GUI.

Edit

Modify the configuration of the reference tree.

Reactivate

Bring back into active service a reference tree that was previously deactivated with the ***Remove*** command.

Remove

Deactivate a reference tree. The reference tree cannot be updated.

You can use the ***Reactivate*** command to bring back a removed reference tree.

Creating a New Reference Tree

Click the ***New*** button at the bottom of the Reference Trees tab to invoke the *Using the New Workspace Wizard* on page 50.

The New Reference Tree Command

The ***New Reference Tree*** command creates a reference tree for a specified stream.

Invoking the New Reference Tree Command

1. Choose ***Admin*** > ***Reference Trees*** from the GUI main menu, to open a Refs tab.
2. Click the ***New*** button at the bottom of the tab.

Using the New Workspace Wizard

The wizard for creating a new reference tree is essentially similar to the *The New Workspace Command* wizard.

The Triggers Tab

The Triggers tab displays the triggers defined for a specified depot. Detailed information about these triggers is found in Chapter 9, "AccuRev Triggers", in the AccuRev *Administrator's Guide*.

See also: *Security* on page 263.

Opening a Triggers Tab

Choose ***Admin*** > ***Triggers*** from the GUI main menu.

Triggers Tab Layout

The Triggers table includes these columns:

Type

pre-create-trig, **pre-keep-trig**, **pre-promote-trig**, or **server-post-promote-trig**.

Execute

The simple name or pathname of the script (or other program) that executes when the trigger fires.

Working in a Triggers Tab

The ***Triggers*** tab is informational only. All trigger maintenance is performed with the AccuRev CLI. See the description of the ***mktrig*** and ***rmtrig*** commands. The data in the Triggers tab is the same as that displayed by the CLI command ***show triggers***.

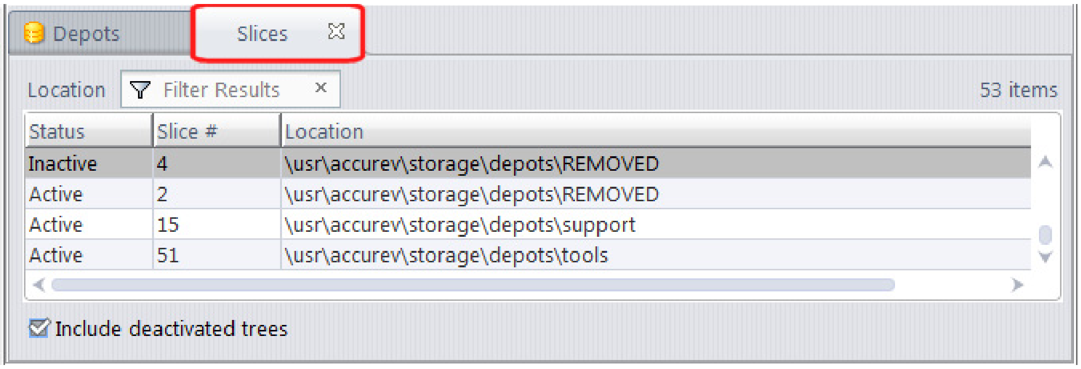
The Slices Tab

The Slices tab shows the physical storage locations of the depots in the AccuRev repository. If the repository is replicated, it shows the locations of the depots in the local replica repository only.

Opening a Slices Tab

Choose ***Admin*** > ***Slices*** from the GUI main menu.

Slices Tab Layout



Status	Slice #	Location
Inactive	4	\usr\accurev\storage\depots\REMOVED
Active	2	\usr\accurev\storage\depots\REMOVED
Active	15	\usr\accurev\storage\depots\support
Active	51	\usr\accurev\storage\depots\tools

☑ Include deactivated trees

The Slices tab contains a multi-column table.

Status

(appears if ***Include deactivated trees*** is checked) **Active** indicates that the depot is available for use. **Inactive** indicates that the depot has been permanently removed from the repository. (The **maintain rmdepot** command permanently and irreversibly removes a depot from the repository. See "Removing a Depot from the AccuRev Repository" in the AccuRev *Administrator's Guide*.)

Slice#

The unique integer that identifies a particular depot. This identifier cannot be modified.

Location

The pathname of the slice's top-level directory. This location can be changed -- for example, to move the depot to a larger and/or faster disk.

Working in a Slices Tab

You can control the display of removed depots. See also: *Working with Tables* on page 14.

Controlling Which Slices are Displayed

Include deactivated trees

If checked, the table includes the slices of depots that have been permanently deactivated with the **maintain** utility (see the AccuRev *Administrator's Guide*). This also adds a ***Status*** column; removed depots have **Inactive** status, and are shaded.

The Locks Command

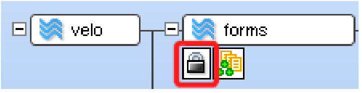
The ***Locks*** command maintains the set of locks that apply to a dynamic stream. Locks control the ability to Promote versions to/from the stream, the ability to configure the stream's contents with the include/exclude facility, and the ability to modify the stream's settings.

See also: *Security* on page 263.

Invoking the Locks Command

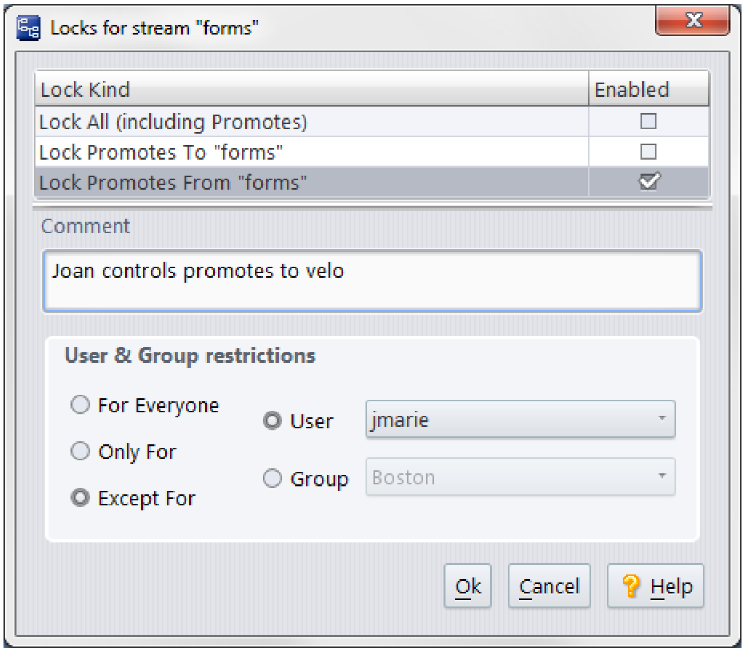
You can open a Locks dialog in the ***StreamBrowser*** in any of these ways:

- Right-click a dynamic stream, and select ***Locks*** from the context menu.
- (if there are currently one or more locks on the stream) Click the ***Locks*** icon that appears below the stream



The Locks Dialog Box

On a given stream, you can create one lock of each type:



Locks for stream "forms"

Lock Kind	Enabled
Lock All (including Promotes)	□
Lock Promotes To "forms"	□
Lock Promotes From "forms"	☑

Comment

Joan controls promotes to velo

User & Group restrictions

○ For Everyone
○ Only For
● Except For

● User: jmarie
○ Group: Boston

Lock All

Disables promotion of versions to/from the stream; disables ***include/exclude mode*** changes to the stream; disables the ***Change Stream*** command for the stream. (Does not disable the ***Remove*** command.)

Lock Promotes To

Prevents versions from being added to the specified stream's default group. Disables promotion to the stream.

Lock Promotes From

Prevents the removal of versions from the stream's default group. Disables promotion and purging from the stream.

A *Lock All* setting takes precedence over "to" and/or "from" settings.

You can restrict the effect of the lock:

Only for

The lock applies only to the specified user, or to all the members of the specified group.

Except for

The lock applies to all users, except for the specified user, or except for all the members of the specified group.

Security

The features of the Security menu option are covered in their own chapter: *Chapter 9 AccuRev Security Overview*.

Schema Editor

The Schema Editor is covered in its own chapter: *Chapter A The Schema Editor (Administrators Only)*

3. The File Browser

The *File Browser* is AccuRev's primary GUI tool for working with files in AccuRev streams and workspaces. You can use the File Browser to perform the daily tasks typically associated with software development, such as viewing and editing files, sharing your work with others, and updating your workspaces with changes from others on your team. The File Browser provides focused views of your workspaces to help you accomplish these and other tasks.

File Browser Modes

The File Browser provides different modes for viewing the files and directories in AccuRev streams and workspaces. Note that not all modes are available in both workspace and stream File Browsers.

- **Explorer Mode** Workspace Explorer **--** The *Explorer* mode displays all the files and directories that are in the current workspace or stream. This can include files and directories that are known to AccuRev, and, for workspaces only, files and directories that are not yet under AccuRev control. The Explorer mode is the default for both the stream and workspace File Browser. See *Explorer Mode* on page 60 for more information.

- **Outgoing Changes Mode** Outgoing **--** The *Outgoing Changes* mode displays only those files and directories that have been added to, changed in, or removed from the current workspace or stream. Use the Outgoing Changes mode to save your work, share your work with others, and more. See *Outgoing Changes Mode* on page 65 for more information.

- **Incoming Changes Mode** Incoming **--** The *Incoming Changes* mode provides a preview of any files and directories that will be changed in, added to, or removed from your workspace when you update your workspace to refresh its contents. You can use the Incoming Changes mode to diff the content in the backing stream against the workspace version since your last update, take a look at the current workspace or stream's parent stream, and more. See *Incoming Changes Mode* on page 70 for more information.

  Note: The Incoming Changes mode is available only for the workspace File Browser -- changes in a parent stream are automatically inherited by all child streams.

- **Conflicts Mode** ⚠ Conflicts **--** The *Conflicts* mode displays the files in the current workspace or stream whose changes conflict with the changes in another version of the same file, either in the parent stream (called an *overlap*) or elsewhere in the stream hierarchy (a *deep overlap*). You can use the Conflicts mode to compare these versions to understand the source of the conflict and merge changes when it makes sense to do so. See *Conflicts Mode* on page 73 for more information.

Changing File Browser Modes

To change a File Browser mode, just click the mode button. File Browser modes can only be displayed one at a time: you cannot display both incoming changes and outgoing changes at the same time, for example.

The following sections describe each of these modes in more detail.

Displaying External Files

In AccuRev, files and directories that are not yet under AccuRev control have a status of **(external)**. You can instruct AccuRev to ignore all external files, in which case external objects are not visible in any File Browser mode. See *Controlling the Display of External Objects* on page 58 for more information. If you prefer, you can simply hide external files from the File Browser display. See *Using the Status Filter Toolbar* on page 70 for more information.

File Browser Types

AccuRev uses separate File Browsers for workspaces and streams (including snapshot and pass-through streams). Workspace and stream File Browsers share basic layout and functionality, but each has a specialized set of features. The Incoming Changes mode in the workspace File Browser, for example, shows the changes others have made to shared files that you will get the next time you update your workspace. (Since streams inherit changes automatically, the stream File Browser does not have the Incoming Changes mode.)

AccuRev also supports a data structure similar to a workspace called a *reference tree*. You cannot open a File Browser for a reference tree. To view the reference trees created for a particular depot, choose **Admin** > **Reference Trees** from the AccuRev menu. See *The Reference Trees Tab* on page 48 for more information.

The remainder of this section describes general File Browser features.

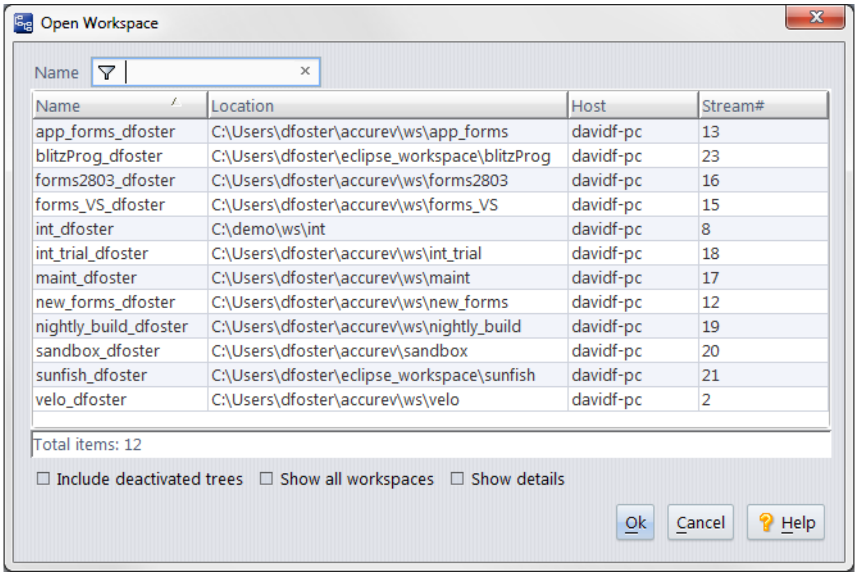
Opening and Closing the File Browser

There are several ways to open a File Browser in the AccuRev GUI:

- **For workspaces and streams:** In the StreamBrowser, double-click the workspace or stream whose files you want to review. (Note that File Browser operations differ across workspaces and streams, including special stream types like pass-through streams and snapshots.)

  Alternative: Right-click the desired workspace or stream and choose **Open** from the context menu.

- **For workspaces only:** Click the **Open Workspace** button on the AccuRev toolbar to display the Open Workspace dialog box.



Name	Location	Host	Stream#
app_forms_dfoster	C:\Users\dfoster\accurev\ws\app_forms	davidf-pc	13
blitzProg_dfoster	C:\Users\dfoster\eclipse_workspace\blitzProg	davidf-pc	23
forms2803_dfoster	C:\Users\dfoster\accurev\ws\forms2803	davidf-pc	16
forms_VS_dfoster	C:\Users\dfoster\accurev\ws\forms_VS	davidf-pc	15
int_dfoster	C:\demo\ws\int	davidf-pc	8
int_trial_dfoster	C:\Users\dfoster\accurev\ws\int_trial	davidf-pc	18
maint_dfoster	C:\Users\dfoster\accurev\ws\maint	davidf-pc	17
new_forms_dfoster	C:\Users\dfoster\accurev\ws\new_forms	davidf-pc	12
nightly_build_dfoster	C:\Users\dfoster\accurev\ws\nightly_build	davidf-pc	19
sandbox_dfoster	C:\Users\dfoster\accurev\sandbox	davidf-pc	20
sunfish_dfoster	C:\Users\dfoster\eclipse_workspace\sunfish	davidf-pc	21
velo_dfoster	C:\Users\dfoster\accurev\ws\velo	davidf-pc	2

Total items: 12

  Then, to display the workspace File Browser, either:

  - Double-click the workspace
  - Select the workspace and click the **Ok** button
  - Right-click the workspace and choose **Open** from the context menu

- **For streams only:** In the workspace File Browser, click the **Open Parent** button to display the File Browser for that workspace's parent stream.

Note that you can open multiple File Browsers concurrently -- for different workspaces, streams, or snapshots, for example. Each new instance of the File Browser is displayed on its own tab.

Refreshing the File Browser

The File Browser displays current data when you first open it, but over time the File Browser view can get out-of-date. For example, if you do work in the workspace directory outside AccuRev -- such as adding a file using operating system commands -- the File Browser will not reflect this change until you refresh it.

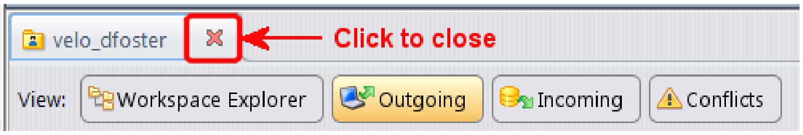
To refresh the File Browser (or any view in AccuRev) manually you can:

- Press F5
- Click the **Refresh View** button (⟳) on the AccuRev toolbar
- Choose **View** > **Refresh** from the AccuRev menu

The File Browser is refreshed automatically any time you change modes or reselect the current mode. Note, however, that the File Browser is not refreshed simply by displaying it. For example, if you have both a StreamBrowser tab and a File Browser tab open in AccuRev and you are currently viewing the StreamBrowser, AccuRev does not refresh the File Browser when you click its tab to display it.

Closing the File Browser

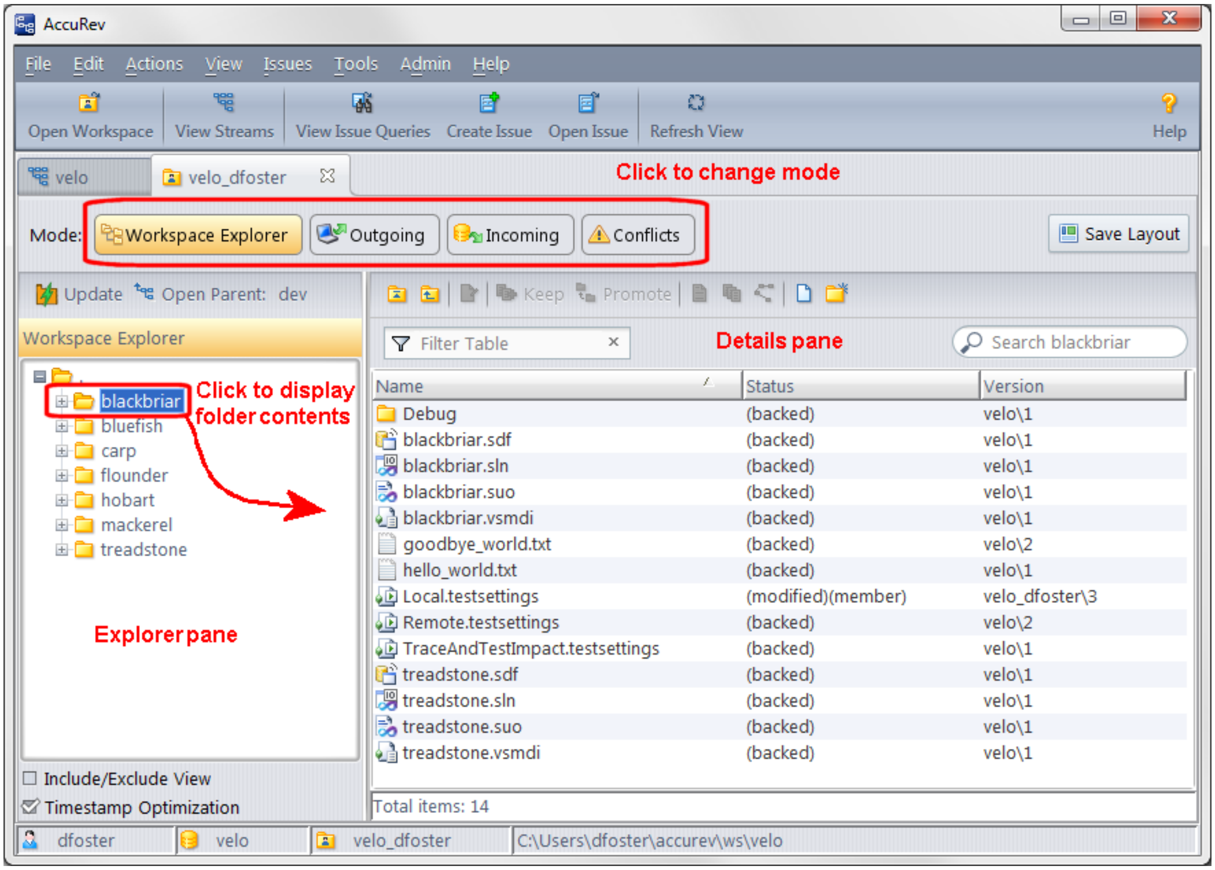
To close an instance of the File Browser, just click the X on the File Browser tab:



Alternative: Right-click the tab and choose **Close** from the context menu.

Parts of the File Browser

The File Browser consists of a single tab that is divided into two or more panes, like the Explorer pane and Details pane shown in the following illustration of the workspace File Browser. (There are separate File Browsers for workspaces and streams. See *File Browser Types* on page 54 for more information.)



Click to change mode

Details pane

Click to display folder contents

Explorer pane

Name	Status	Version
Debug	(backed)	velo\1
blackbriar.sdf	(backed)	velo\1
blackbriar.sln	(backed)	velo\1
blackbriar.suo	(backed)	velo\1
blackbriar.vsmdi	(backed)	velo\1
goodbye_world.txt	(backed)	velo\2
hello_world.txt	(backed)	velo\1
Local.testsettings	(modified)(member)	velo_dfoster\3
Remote.testsettings	(backed)	velo\2
TraceAndTestImpact.testsettings	(backed)	velo\1
treadstone.sdf	(backed)	velo\1
treadstone.sln	(backed)	velo\1
treadstone.suo	(backed)	velo\1
treadstone.vsmdi	(backed)	velo\1

Total items: 14

Except where noted, the following general descriptions apply to the File Browser for both workspaces and streams.

Explorer Pane

The *Explorer pane* displays a tree that represents the directory structure of the current workspace or stream. Individual folders in the tree can be expanded and collapsed by clicking the plus (+) and minus (-) symbol, respectively, next to the folder icon. Clicking a folder displays the name and other information about the files and directories in the associated directory in the Details pane.

Explorer pane toolbar. A toolbar above the Explorer pane lets you open the stream File Browser for the parent of the current workspace or stream in a new File Browser instance. In the workspace File Browser, the toolbar includes a button that lets you update the workspace with changes from the parent stream. See *Updating Your Workspace* on page 73 for more information.

Tip: The AccuRev Include/Exclude View enables you to define and work with a subset of the depot's elements in a workspace or stream. Turning on the Include/Exclude View can affect what appears in both the Explorer pane and the Details pane. See *Including and Excluding Files* on page 103 for more information.

Details Pane

Generally speaking, the *Details pane* displays the files and directories contained in the currently selected folder in the Explorer pane. (The Details pane can also display the results of a search of the entire workspace. See *Searching the Workspace* on page 63 for more information.) AccuRev refreshes the Details pane each time you select a folder in the Explorer pane.

Note: The display is not recursive in Explorer mode. For example, if you select the workspace root, which contains 3 directories and 10 files, the **Total items**: counter at the bottom of the Details pane will show 13 -- even if there are multiple subdirectories and files in each of the 3 directories at the workspace root.

The columns displayed in the Details pane table differ based on a number of factors (like which File Browser you have opened, and whether or not you have performed a workspace search, for example), but they typically include Name, Status, and Version. See *Details Pane Table Columns Reference* on page 141 for descriptions of the columns that can be displayed in the Details pane.

You can choose which columns to display, and how you want them ordered and sorted. And you can use splitters to resize panes. For example, you can use the horizontal splitter that separates the Explorer, and Details panes to change their relative sizes. See *Working with Tables* on page 14 to learn about customizing tables in AccuRev. See *Saving Layout and Other Usage Preferences* on page 58 to learn how to save the changes you make your new File Browser default.

Details pane toolbar. The Details pane toolbar provides quick access to actions that are frequently performed on files and elements. For example, in the workspace File Browser you can use the toolbar to:

- Create new files and folders
- Place files under AccuRev control
- Save your changes
- Share your changes with others
- Display a graph of an element’s versions

Tip: The toolbar contains buttons for only the most commonly performed actions, and toolbars vary based on the active File Browser mode. To see all the actions available for selected files or directories, right-click to display the context menu.

Diff Pane

In some File Browser modes, AccuRev displays a read-only Diff pane that shows you different versions of the file you have selected in the Details pane. The versions AccuRev displays vary based on mode, so see the Diff pane discussion for the mode you are interested in for more information.

You can, of course, diff workspace and stream versions of a file using an interactive Diff tool (either AccuRev’s built-in tool or a third-party tool). See *The Diff Tool* on page 203 for more information.

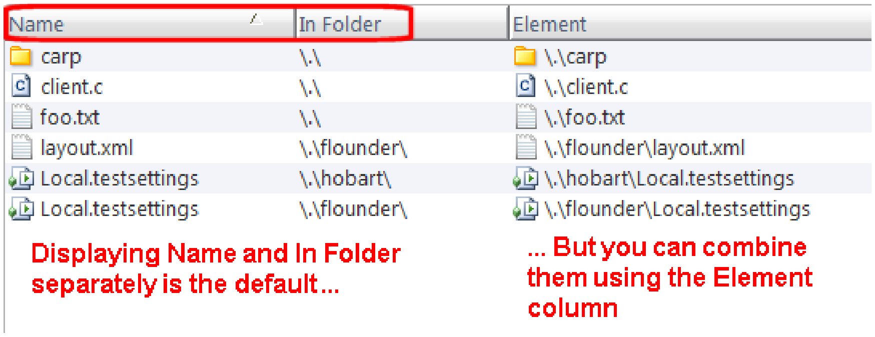
Tip: You can hide the Diff pane by clicking the **Hide Diff Pane** button (Hide Diff Pane) on the Details pane toolbar.

Controlling the Display of Element Names

Typically, the elements displayed in the Details pane are located in a number of different folders, throughout the workspace. A user preference controls how to display the locations of the elements:

- Display the complete depot-relative pathname of the object as a single string (*Element* column).

- Display the object's simple name (*Name* column) separately from the pathname to its parent directory (*In Folder* column).



Name	In Folder	Element
carp	\.\	\.\carp
client.c	\.\	\.\client.c
foo.txt	\.\	\.\foo.txt
layout.xml	\.\flounder\	\.\flounder\layout.xml
Local.testsettings	\.\hobart\	\.\hobart\Local.testsettings
Local.testsettings	\.\flounder\	\.\flounder\Local.testsettings

Displaying Name and In Folder separately is the default...

... But you can combine them using the Element column

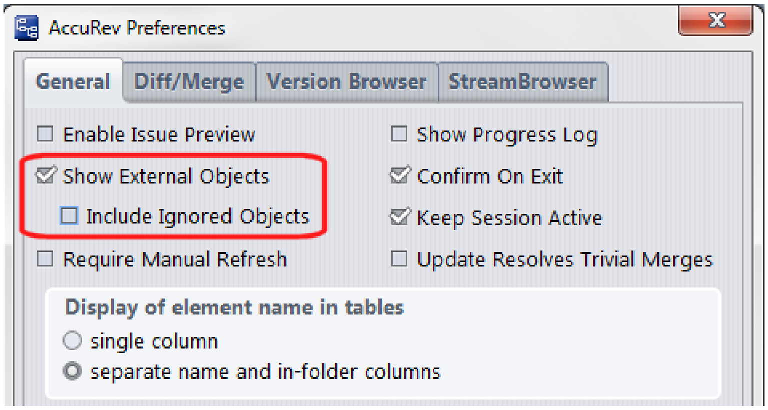
In **Tools** > **Preferences**, use the **Display of element name in tables** preference to make this choice. It affects all instances of the File Browser, along with other tools that display element pathnames.

Tip: You can add and remove columns in the Details pane as you would with any other table in AccuRev. See *Working with Tables* on page 14 for more information.

Controlling the Display of External Objects

Note: This topic applies to a File Browser displaying the contents of a workspace -- not a dynamic stream or snapshot.

By default, the Details pane of the File Browser shows both elements and external objects. The **Show External Objects** preference provides an all-or-none control over the display of external objects -- even those external objects you might have instructed AccuRev to ignore when they match a user-specified ignore pattern.



☐ Enable Issue Preview

☑ Show External Objects

☐ Include Ignored Objects

☐ Require Manual Refresh

☐ Show Progress Log

☑ Confirm On Exit

☑ Keep Session Active

☐ Update Resolves Trivial Merges

Display of element name in tables

○ single column

● separate name and in-folder columns

If you also want AccuRev to display ignored external objects, select the **Include Ignored Objects** preference, too.

See *AccuRev Preferences (Tools > Preferences Command)* on page 39 to learn more about setting preferences in AccuRev. See *Chapter 3 Specifying Ignore Patterns for External Objects* in the *AccuRev Technical Notes* for more information and examples of how to specify ignore patterns.

Saving Layout and Other Usage Preferences

As described previously, you can customize the File Browser to suit your needs -- you can choose which columns you want to display, and in which order; you can change the relative widths of the Explorer and

Details panes; and you can choose to hide the Diff pane in the modes that support it, for example. The **Save Layout** button (Save Layout) lets you save these and other changes you make to default File Browser layout settings as a personal preference. Once saved, AccuRev uses your customized layout settings the next time open a new File Browser.

For example, you might decide that the Last Modified column is useful in the Outgoing Changes mode, that the default height for the Diff pane is too tall, and that you are generally interested in seeing only those files with a (modified) status. Once you set up Outgoing Changes mode to your liking, clicking the **Save Layout** button saves these settings -- as well as any changes made in other File Browser modes -- and makes Outgoing Changes the new default mode the next time the File Browser is opened.

What Settings are Saved?

The **Save Layout** button saves these settings:

- Mode -- The mode you are in when you click **Save Layout** becomes the default mode the next time you open the File Browser. Within the Explorer mode, AccuRev also saves the setting for the Include/ Exclude View as a preference.
- Columns -- AccuRev lets you save changes to Details pane columns, column width, order, and sort order.
- Diff pane -- The Diff pane is displayed by default, but you can change this setting and save it using the **Save Layout** button.
- Status filters -- AccuRev remembers status filter settings for modes that support them.
- Splitters -- AccuRev lets you save any changes you make to the relative layout of the Explorer, Details, and Diff panes.
- Timestamp Optimization option -- Whether or not AccuRev uses cached server information to identify modified files in the workspace. See *The Explorer and Details Panes* on page 61 for more information. (This setting is not available in Incoming Changes mode.)
- Include Deep Overlaps option -- Whether or not AccuRev searches for overlap files beyond time-basis streams to identify elements with a so-called "deep" **(overlap)** status. See *Types of Conflicts Displayed in Conflicts Mode* on page 76 for more information. (This setting is applicable to Conflicts mode only.)

Note that AccuRev saves layout settings for the workspace File Browser and stream File Browser separately.

How to Save Layout Settings

To save layout settings:

1. Open the File Browser for the workspace or stream whose default layout settings you want to change.
2. Make the changes that you want in each mode -- add and remove columns, changes their order and width, and so on.
3. Are you in the mode that you want AccuRev to use as the default mode the next time you open the File Browser?

   - If *yes*, click the **Save Layout** button.
   - If *no*, change to the desired default mode and click the **Save Layout** button.

   AccuRev uses these settings the next time you open the File Browser of the type whose settings you just modified.

Alternatives to the File Browser

The File Browser is not intended to completely replace the Windows Explorer or comparable tools on UNIX systems. For example, the File Browser does not attempt to implement the Windows "file types" functionality, based on filename suffixes.

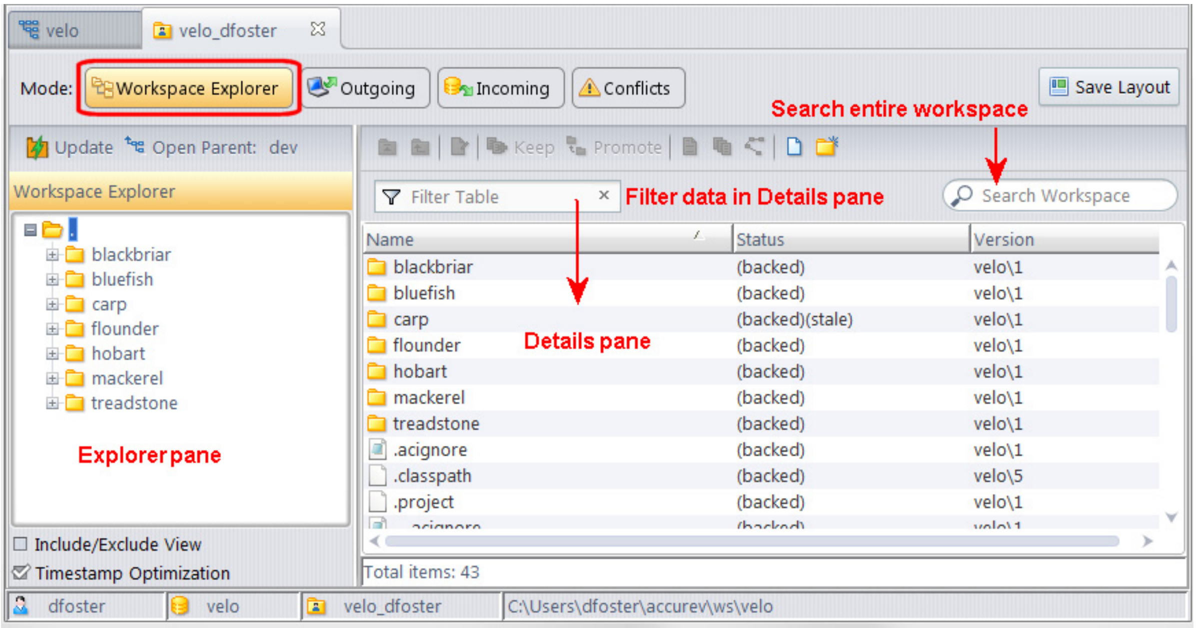
Some people do most of their work inside an integrated development environment (IDE), such as Visual Studio or Eclipse. These environments have their own "explorer" or "file browser" built in, typically organizing files into separate "projects". You can use AccuRev commands within an IDE if an AccuRev Plug-In integration (or an integration from a third-party developer) for that environment is available. Even if you use an IDE most of the time, there are several reasons to consider using the File Browser occasionally:

- The IDE does not display all of a file's configuration-management properties, such as the current version-ID.
- Certain AccuRev commands cannot be executed through the IDE integration, only from the AccuRev File Browser.
- Specialized modes in the workspace File Browser can help you quickly understand which workspace files are out-of-date and require updating and any file whose workspace version is in conflict with another version elsewhere in the workspace stream hierarchy.

It is important to keep in mind that while you can use a wide variety of tools to make *content changes* to a file, you must use AccuRev commands to make valid *namespace changes*: namespace changes include renaming an element, moving an element to another directory, or deleting an element. The AccuRev Plug-In IDE integrations support these namespace operations.

Explorer Mode

The *Explorer* mode displays the files and directories that are in the current workspace or stream, as shown in the following illustration of the workspace File Browser. The Explorer mode is selected by default when you open the File Browser for a workspace or a stream. (See *File Browser Modes* on page 53 for information on other modes.)



Search entire workspace

Filter data in Details pane

Details pane

Explorer pane

Name	Status	Version
blackbriar	(backed)	velo\1
bluefish	(backed)	velo\1
carp	(backed)(stale)	velo\1
flounder	(backed)	velo\1
hobart	(backed)	velo\1
mackerel	(backed)	velo\1
treadstone	(backed)	velo\1
.acignore	(backed)	velo\1
.classpath	(backed)	velo\5
.project	(backed)	velo\1

Total items: 43

This section covers the following topics:

- *The Explorer and Details Panes* on page 61
- *How the Workspace Explorer Differs from the Stream Explorer* on page 62
- *Identifying Current Workspace Location* on page 63
- *Searching the Workspace* on page 63

The Explorer and Details Panes

Generally speaking, the Explorer mode displays all the contents of the current stream or workspace. When you select a directory in the Explorer pane, the Details pane shows only the files and subdirectories for that directory.

Note the following settings, however, which can affect what files and directories are displayed in the Explorer and Details panes and how AccuRev determines whether or not a file should have a status of **(modified)**.

- **Include/Exclude View Option** -- The Include/Exclude view lets you review and set rules that influence which directories and files are visible in the Explorer and other File Browser modes. See *Including and Excluding Files* on page 103 for more information.
- **Timestamp Optimization Option** -- By default, AccuRev uses cached server information to identify which files in the workspace should have a status of **(modified)**. To learn more about this option and when you might want to turn it off, see *Chapter 5 Timestamp Optimization (TSO)* in the AccuRev *Administrator's Guide*. See *AccuRev Element Status Reference* on page 136 for a discussion of **(modified)** and other statuses.

In addition, you can instruct AccuRev to ignore any files and directories that are not under AccuRev control. See *Controlling the Display of External Objects* on page 58 for more information.

Details Pane Toolbar Reference

The following table summarizes the tools available on the Details pane toolbar and where to find more information.

Tool	Description	Available For...
Go to Root	Displays the contents of the workspace's (and stream's) top-level directory.	Workspace and Stream
Go Up One Level	Displays the contents of the current directory's parent. If the current directory is the target of a directory link, this command switches to the target's parent, not to the directory containing the link.	Workspace and Stream
Edit	Opens the selected file in a text editor. See *Editing a File* on page 77.	Workspace
Keep	Saves your work by creating a new version in the workspace stream. See *Keep: Creating a New Version* on page 110.	Workspace
Promote	Shares your work with others by sending the selected files to the backing stream. See *Promote: Sharing Your Work* on page 113.	Workspace and Stream
Demote	Backs out changes by sending the selected files to a child stream or workspace. See *Demote: Moving Elements Out of a Stream* on page 99.	Stream
Annotate	Lists the contents of the selected text file, annotating each line. See *Annotate: Reviewing Transactions on an Element* on page 91.	Workspace and Stream
History	Displays the transactions associated with the selected element in the History Browser. See *Chapter 5 The History Browser*.	Workspace and Stream
Version Browser	Displays all the versions, and their relationships, of the selected element in the Version Browser. See *Chapter 6 The Version Browser*.	Workspace and Stream
New File	Creates a new file in the current workspace. See *Creating a New File* on page 92.	Workspace
New Folder	Creates a new folder in the current workspace. See *Creating a New Folder* on page 94.	Workspace
Hide Diff Pane Hide Diff Pane	Hides/shows the Diff pane in Outgoing Changes, Incoming Changes, and Conflicts modes. See *Diff Pane* on page 57.	Workspace and Stream

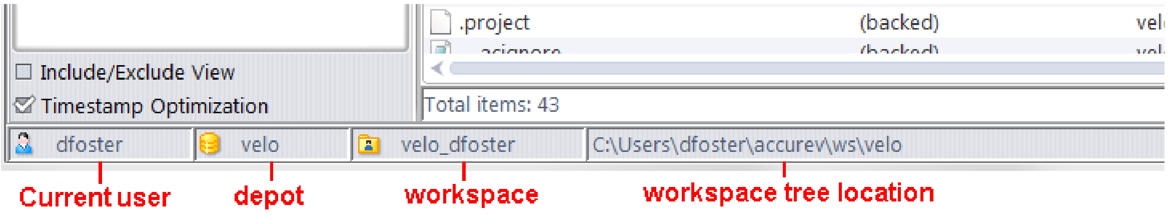
How the Workspace Explorer Differs from the Stream Explorer

While the Explorer mode is similar for workspaces and streams, the workspace File Browser provides features and functionality that take advantage of its direct access to your local file system. These features include:

- **Workspace search --** The workspace File Browser lets you search the entire workspace. See *Searching the Workspace* on page 63 for more information.
- **Update --** By definition, the Update command refreshes the workspace with changes from the parent stream, so Update is available only in the workspace File Browser. (In AccuRev, streams inherit changes from parent streams automatically.) See *Updating Your Workspace* on page 73 for more information.
- **File-based operations --** You can accomplish any operation that requires direct access to your local file system using the workspace File Browser -- creating, editing, renaming, or deleting a file; cut/copy/paste operations; locking a file using AccuRev's anchor command; and so on. See *Common Usage Scenarios* on page 77 for more information.

Identifying Current Workspace Location

In the workspace File Browser, indicators at the bottom of the AccuRev GUI window show the name and location of the current workspace.



.project (backed) vel

□ Include/Exclude View

☑ Timestamp Optimization

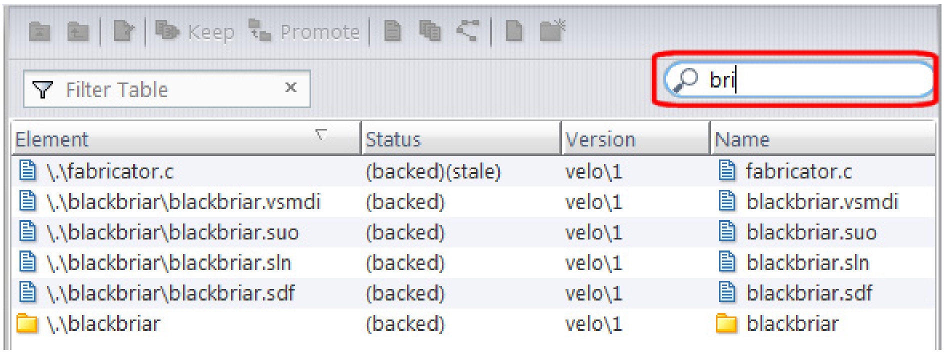
Total items: 43

dfoster | velo | velo_dfoster | C:\Users\dfoster\accurev\ws\velo

In the stream File Browser, only the names of the current user and depot are displayed.

Searching the Workspace

The **Search Workspace** field in the Workspace Explorer mode lets you search the current workspace for any files and directories whose name matches the string you enter. You can enter any character that is legal for an AccuRev element name. (See *User-Specified Names for AccuRev Entities* on page 23 for details.) Wildcards ("*" and "?") are accepted. If you do not use wildcards, AccuRev searches the workspace for any object whose name *contains* the string you enter. Entering "bri" returns both fabricator.c and all blackbriar files and directories, for example:



bri

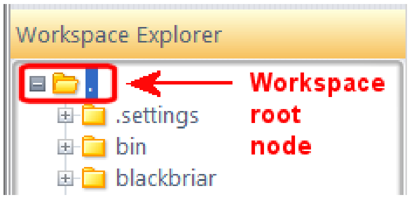
Element	Status	Version	Name
\.\fabricator.c	(backed)(stale)	velo\1	fabricator.c
\.\blackbriar\blackbriar.vsmdi	(backed)	velo\1	blackbriar.vsmdi
\.\blackbriar\blackbriar.suo	(backed)	velo\1	blackbriar.suo
\.\blackbriar\blackbriar.sln	(backed)	velo\1	blackbriar.sln
\.\blackbriar\blackbriar.sdf	(backed)	velo\1	blackbriar.sdf
\.\blackbriar	(backed)	velo\1	blackbriar

> **Note:** The scope of the search is determined by the currently selected node in the Workspace Explorer tree. For example, if you select a specific folder, AccuRev searches only that folder for objects that match the string you enter in the **Search Workspace** field. (AccuRev changes **Workspace** to the name of the folder you select to reinforce this point.) If you want to search the entire workspace, be sure to first select the workspace root node in the Workspace Explorer tree.

In addition, search is available only for locally accessible workspaces, not for workspaces on a different machine. Search is also not available for streams.

To search the current workspace:

1. Select the workspace folder you want to search. If you want to search the entire workspace, select the root node.



Workspace root node

2. In the **Search Workspace** field, enter the string or name you want to search the workspace for. You can use "*" and "?" wildcards.

3. Press Enter.

   Depending on the size of the workspace, AccuRev displays a status box while it searches the workspace for the files and directories that match the string you entered in the **Search Workspace** field.

4. Click the **Run in Background** button if you want to do other things while AccuRev searches.

   Search results are displayed in the Details pane when the search is complete. If the search results return a directory, you can double-click it to view directory contents in the Details pane. The directory tree in the Workspace Explorer expands and selects the correct node/directory.

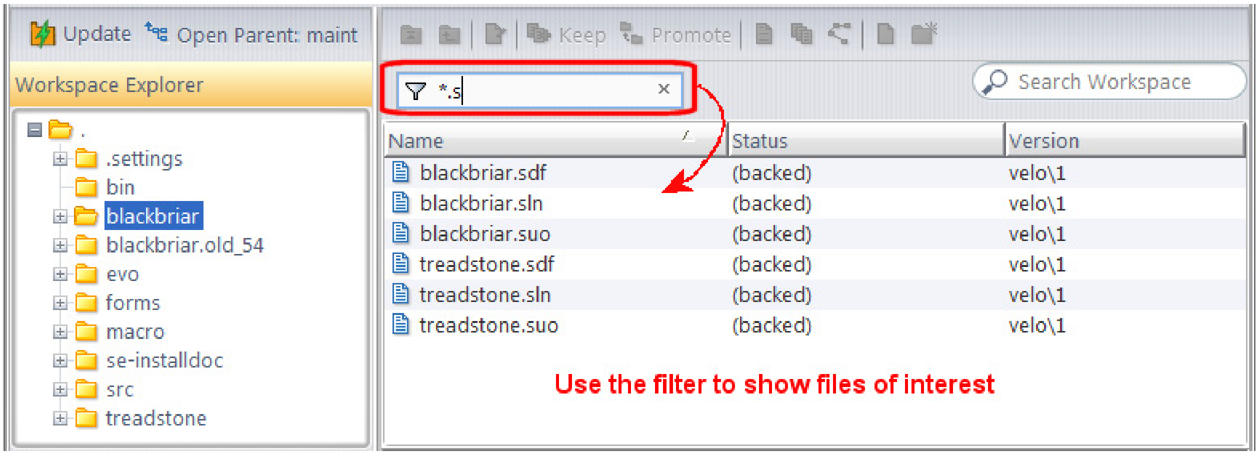
To clear the search field:

1. Delete the current entry in the **Search Workspace** field.

2. Press Enter.

Filtering the Details Pane Table

You can use the **Filter Table** field to filter the objects displayed in the Details pane. As you type, the Details pane is filtered to show only those objects containing the characters you enter. The rules for filter strings are the same as for the search.



*.s

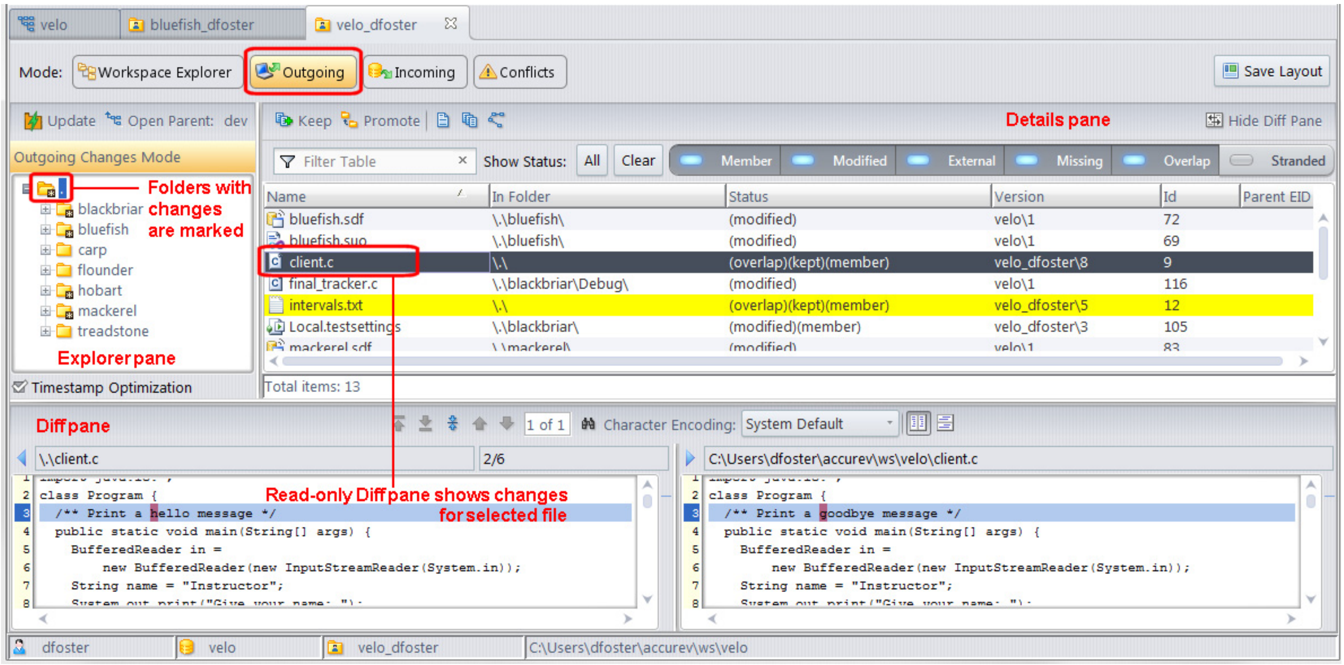
Name	Status	Version
blackbriar.sdf	(backed)	velo\1
blackbriar.sln	(backed)	velo\1
blackbriar.suo	(backed)	velo\1
treadstone.sdf	(backed)	velo\1
treadstone.sln	(backed)	velo\1
treadstone.suo	(backed)	velo\1

Use the filter to show files of interest

To clear the filter, click the X button in the **Filter Table** field.

Outgoing Changes Mode

The *Outgoing Changes* mode displays changes to the current workspace or stream. For the workspace File Browser, Outgoing Changes mode shows only those files and directories that have been added to, changed in, or removed from the workspace, as shown in the following illustration. Outgoing Changes mode for the stream File Browser shows any work that has been promoted to that stream from a workspace or child stream.



Name	In Folder	Status	Version	Id	Parent EID
bluefish.sdf	\.\bluefish\	(modified)	velo\1	72	
bluefish.suo	\.\bluefish\	(modified)	velo\1	69	
client.c	\.\	(overlap)(kept)(member)	velo_dfoster\8	9	
final_tracker.c	\.\blackbriar\Debug\	(modified)	velo\1	116	
intervals.txt	\.\	(overlap)(kept)(member)	velo_dfoster\5	12	
Local.testsettings	\.\blackbriar\	(modified)(member)	velo_dfoster\3	105	
mackerel.sdf	\.\mackerel\	(modified)	velo\1	83	

Total items: 13

\.\client.c 2/6

```
class Program {
  /** Print a hello message */
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Instructor";
```

C:\Users\dfoster\accurev\ws\velo\client.c

```
class Program {
  /** Print a goodbye message */
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Instructor";
```

Tip: If you are an existing AccuRev user, the Outgoing Changes mode displays the same results as if you had performed a search for Pending elements in AccuRev 5.7 and earlier. To these results, it adds **(external)** files and **(stranded)**, **(missing)**, and **(member)** elements.

You can use the Outgoing Changes mode to accomplish many tasks in AccuRev, but its primary purpose is to provide a streamlined view of changes to the workspace and stream, and the tools to let you quickly save and share those changes with others.

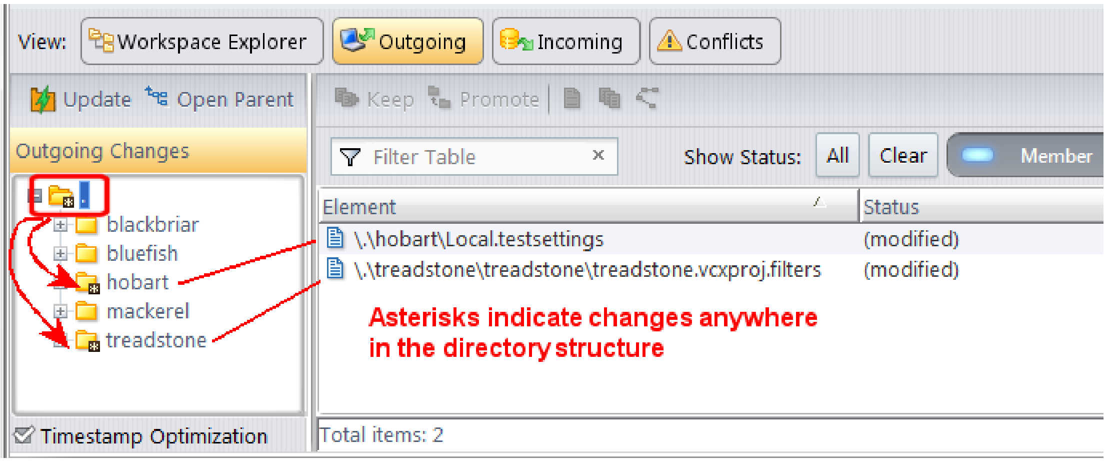
This section covers the following topics:

- *The Explorer and Details Panes* on page 65
- *Filtering the Details Pane by AccuRev Status* on page 69
- *Saving and Sharing Your Work* on page 70

See *File Browser Modes* on page 53 for more information on other modes.

The Explorer and Details Panes

In Outgoing Changes mode, the Explorer pane shows the complete directory structure of the current workspace or stream. Directories that contain changes -- whether modified elements, or files that have not yet been added to AccuRev, for example -- are identified with an asterisk, as shown in the following illustration.



Element	Status
\.\hobart\Local.testsettings	(modified)
\.\treadstone\treadstone\treadstone.vcxproj.filters	(modified)

Asterisks indicate changes anywhere in the directory structure

Note that directories are marked with an asterisk regardless of where in the directory the change is located. Here, for example, the **\treadstone** directory has an asterisk even though the modified file is located in the **\treadstone\treadstone** subdirectory, and not at the directory root (as is the case with the change in the \hobart directory). Similarly, the root node is marked with an asterisk if there are changes anywhere in the workspace or stream.

Clicking a directory with an asterisk displays the changes in that directory. To help you quickly identify where in the workspace a specific change exists, the Element column in the Details pane displays the full path of the changed element.

Tip: Since the Outgoing Changes mode shows only changes in the workspace or stream, the Details pane will be empty if you click a directory that is *not* marked with an asterisk

You can format columns, change the sort order, and filter displayed objects in the Details pane as you would with any other table in AccuRev. See *Working with Tables* on page 14 for more information. In addition, you can filter the table based on AccuRev status. See *Filtering the Details Pane by AccuRev Status* on page 69 for more information.

Timestamp Optimization option. By default, AccuRev uses cached server information to identify which files in the workspace should have a status of **(modified)**. To learn more about this option and when you might want to turn it off, see *Chapter 5 Timestamp Optimization (TSO)* in the AccuRev *Administrator's Guide*. See *AccuRev Element Status Reference* on page 136 for a discussion of **(modified)** and other statuses.

Details Pane Toolbar Reference

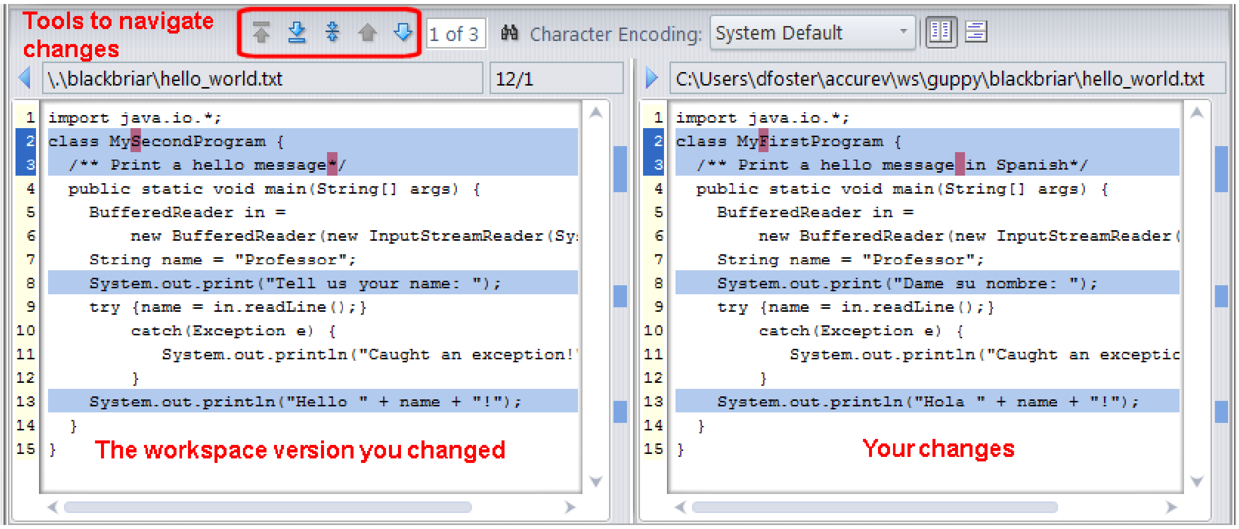
The following table summarizes the tools available on the Details pane toolbar and where to find more information.

Tool	Description	Available For...
Keep	Saves your work by creating a new version in the workspace stream. See *Keep: Creating a New Version* on page 110.	Workspace
Promote	Shares your work with others by sending the selected files to the backing stream. See *Promote: Sharing Your Work* on page 113.	Workspace and Stream

Tool	Description	Available For...
Demote	Backs out changes by sending the selected files to a child stream or workspace. See *Demote: Moving Elements Out of a Stream* on page 99.	Stream
Annotate	Lists the contents of the selected text file, annotating each line. See *Annotate: Reviewing Transactions on an Element* on page 91.	Workspace and Stream
History	Displays the transactions associated with the selected element in the History Browser. See *Chapter 5 The History Browser*.	Workspace and Stream
Version Browser	Displays all the versions, and their relationships, of the selected element in the Version Browser. See *Chapter 6 The Version Browser*.	Workspace and Stream
Revert to Basis	Used to undo changes to the version in a stream. See *Revert: Undoing Changes* on page 126.	Stream
Send to Change Palette	Used to resolve conflicts when promoting one or more versions from a dynamic stream to a stream other than its parent stream. See *Chapter 10 Change Packages*.	Stream

The Diff Pane

When you select a file in the Details pane, AccuRev displays versions of that file in a read-only Diff pane that shows you the workspace version that you modified, on the left, and the most recent changes you made to that version, on the right, as shown in the following illustration. These are the changes that will be saved to the workspace version when you keep the file, and written to the workspace's parent stream when you promote the file.


Tools to navigate changes
1 of 3
Character Encoding: System Default
\.\blackbriar\hello_world.txt
12/1
C:\Users\dfoster\accurev\ws\guppy\blackbriar\hello_world.txt
The workspace version you changed
Your changes

```
import java.io.*;
class MySecondProgram {
  /** Print a hello message*/
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(Sy:
    String name = "Professor";
    System.out.print("Tell us your name: ");
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!'
        }
    System.out.println("Hello " + name + "!");
  }
}
```

```
import java.io.*;
class MyFirstProgram {
  /** Print a hello message in Spanish*/
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(
    String name = "Professor";
    System.out.print("Dame su nombre: ");
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exceptic
        }
    System.out.println("Hola " + name + "!");
  }
}
```

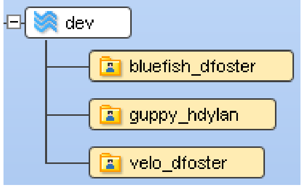
If you select a file that AccuRev cannot diff (a file with a status of **(external)** or some types of binary files, for example), AccuRev displays an informational message in the Diff pane.

The Diff pane is read only, but the changes navigation, coloring, and other features it uses are similar to those of the AccuRev Diff tool. See *Diff Tab Layout* on page 207 for more information.

Tip: You can hide the Diff pane by clicking the **Hide Diff Pane** button (Hide Diff Pane).

Example

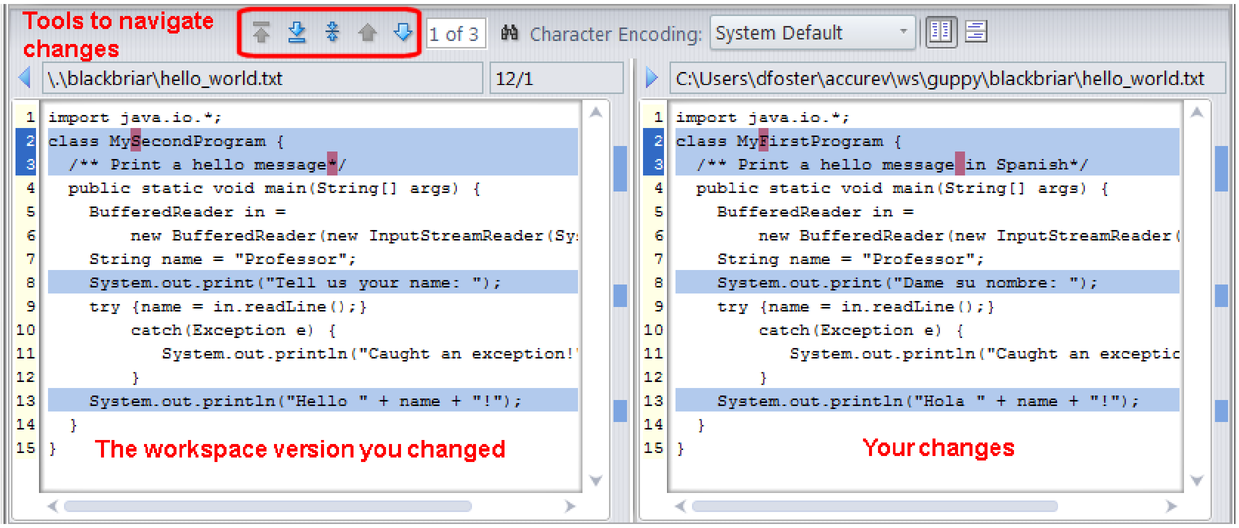
Consider the following **dev** stream and its workspaces:



Each of these workspaces has a current version of the file **hello_world.txt**:

```
import java.io.*;
class MySecondProgram {
  /** Print a hello message*/
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Professor";
    System.out.print("Give your name: ");
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!");
        }
    System.out.println("Hello " + name + "!");
  }
}
```

If user hdylan makes changes to the file for a Spanish speaking user and then keeps his changes, the Outgoing Mode diff pane shows the version of the file he modified in his workspace on the left, and the changes he made to that version on the right, as shown here:



\\blackbriar\hello_world.txt   12/1

```
import java.io.*;
class MySecondProgram {
  /** Print a hello message*/
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(Sy:
    String name = "Professor";
    System.out.print("Tell us your name: ");
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!'
        }
    System.out.println("Hello " + name + "!");
  }
}
```

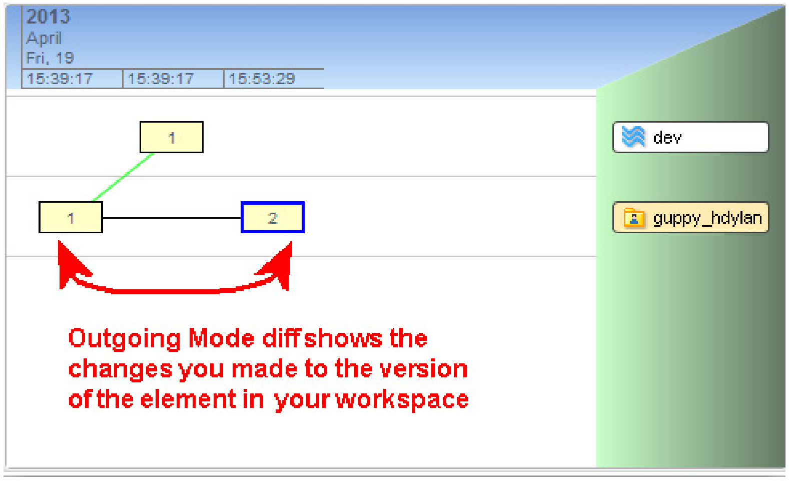
The workspace version you changed

C:\Users\dfoster\accurev\ws\guppy\blackbriar\hello_world.txt

```
import java.io.*;
class MyFirstProgram {
  /** Print a hello message in Spanish*/
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(
    String name = "Professor";
    System.out.print("Dame su nombre: ");
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exceptic
        }
    System.out.println("Hola " + name + "!");
  }
}
```

Your changes

The Version Browser depicts these change events as follows -- version 2 is the most recent version, the one that includes hdylan’s changes:

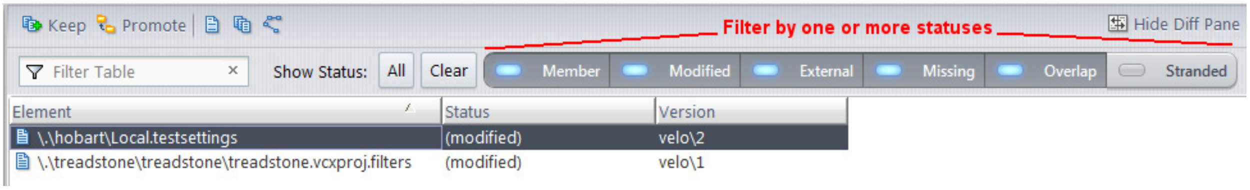


Outgoing Mode diff shows the changes you made to the version of the element in your workspace

See *Chapter 6 The Version Browser* to learn more about the Version Browser and its features.

Filtering the Details Pane by AccuRev Status

The Details pane toolbar includes a filter that lets you filter by one or more of AccuRev statuses.



Filter by one or more statuses

Element	Status	Version
\.\hobart\Local.testsettings	(modified)	velo\2
\.\treadstone\treadstone\treadstone.vcxproj.filters	(modified)	velo\1

These filters, and their availability in both workspace and stream File Browsers, are summarized in the following table.

Filter	Status Description	Available For...
Member	The element has been changed and the changes have been saved to the workspace.	Workspaces and streams
Modified	The element has been changed but has not been saved to the workspace (using Keep).	Workspaces only
External	The file was added to the workspace but has not yet been placed under AccuRev control.	Workspaces only
Missing	The element was deleted from the workspace (using a file system delete command, for example).	Workspaces only
Overlap	Changes have been made to the element in both the workspace and in the parent stream.	Workspaces and streams
Stranded	The element is active in the workspace or stream but it cannot be accessed through the file system. This can occur when an element's directory is defuncted or excluded, for example.	Workspaces and streams

See *AccuRev Element Status Reference* on page 136 for more information on AccuRev element statuses, their definitions, and how they occur.

Using the Status Filter Toolbar

Status buttons act like toggles and can be turned on and off by clicking them. For example, if you do not want to see files with an **(external)** status, click the **External** button. Here are some other usage tips:

- To show only objects with one or more specific statuses, click **Clear**, and then click the buttons for those statuses.

  Tip: When you click **Clear**, the Details pane will be empty because the Outgoing Changes mode shows only those objects and elements with statuses represented by a filter.

- To quickly select all status filters, click the **All** button.
- To save the set of selected status filters as your default, click the **Save Layout** button.

Saving and Sharing Your Work

The Outgoing Changes mode shows you the changes in the current workspace or stream -- work that at some point you probably want to share with others. In AccuRev, you share your work by *promoting* changes from a stream or workspace to the parent stream.

To promote changes from the File Browser Outgoing Changes mode:

1. Select the files you want to promote.
2. Click the Promote button ( ) on the toolbar.

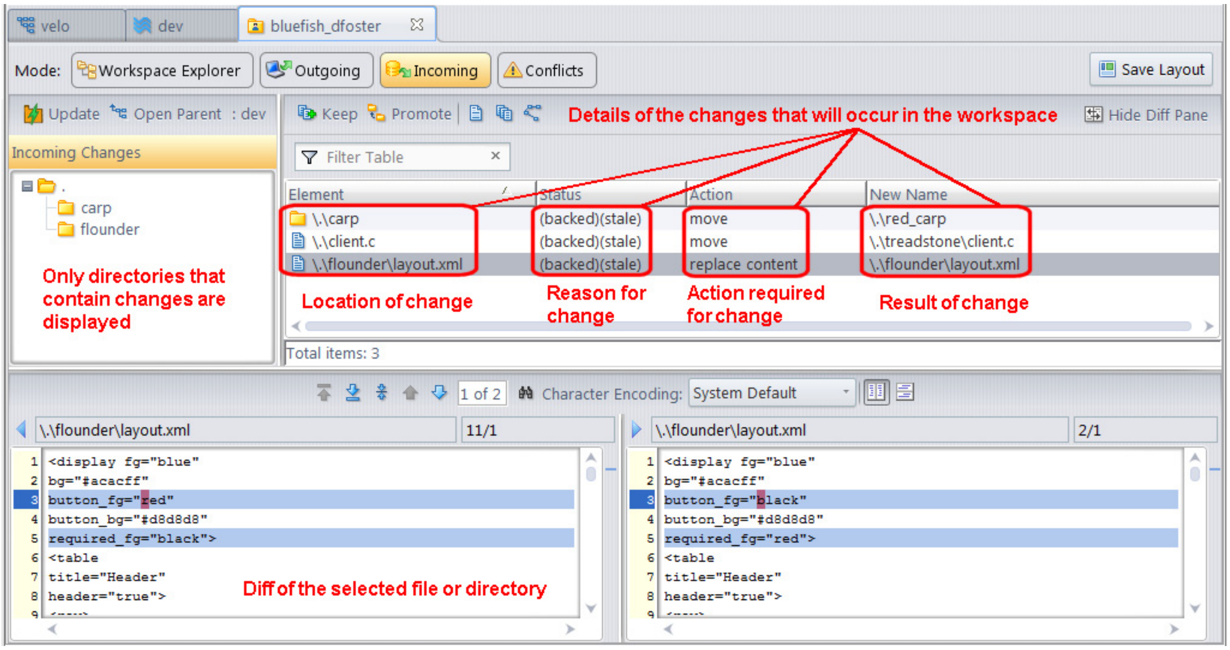
   Tip: In the workspace File Browser, the Promote action automatically performs any other actions that might be required in order to promote the file. For example, when you promote an external file (one that is not currently under AccuRev control), AccuRev first performs the Add and Keep actions.

   In addition, if you have selected one or more files that cannot be promoted (because they have an **(overlap)** status, for example), AccuRev informs you of the situation and gives you the opportunity to continue the Promote action for all eligible files.

See *Saving a File* on page 78 and *Sharing Your Work with Others* on page 80 for more detailed information on saving and sharing work in AccuRev.

Incoming Changes Mode

The *Incoming Changes* mode displays all of the changes that have occurred in the workspace's parent stream since the last time you refreshed your workspace content (called *updating* in AccuRev). New or renamed directories, files that have been moved from one directory to another, and files whose content has changed are all examples of the types of changes displayed in this mode -- the same changes that will be made in the workspace when you update it. Because streams automatically inherit changes from their parent streams, the Incoming Changes mode is available only in the workspace File Browser.



Details of the changes that will occur in the workspace

Element	Status	Action	New Name
\.\carp	(backed)(stale)	move	\.\red_carp
\.\client.c	(backed)(stale)	move	\.\treadstone\client.c
\.\flounder\layout.xml	(backed)(stale)	replace content	\.\flounder\layout.xml

Total items: 3

Only directories that contain changes are displayed

Location of change

Reason for change

Action required for change

Result of change

\.\flounder\layout.xml 11/1

```
<display fg="blue"
bg="#acacff"
button_fg="red"
button_bg="#d8d8d8"
required_fg="black">
<table
title="Header"
header="true">
```

\.\flounder\layout.xml 2/1

```
<display fg="blue"
bg="#acacff"
button_fg="black"
button_bg="#d8d8d8"
required_fg="red">
<table
title="Header"
header="true">
```

Diff of the selected file or directory

Tip: If you are an existing AccuRev user, Incoming Changes mode shows the same results as those displayed by the Update Preview search in AccuRev 5.7 and earlier.

This section covers the following topics:

- *The Explorer and Details Panes* on page 71
- *The Diff Pane* on page 72
- *Updating Your Workspace* on page 73

See *File Browser Modes* on page 53 for more information on other modes.

The Explorer and Details Panes

The Explorer pane in Incoming Changes mode shows *only* those directories that have themselves changed or that contain one or more files that have changed since the last time you updated the current workspace. For example, if a file in the **\flounder** directory was modified, the **\flounder** directory will appear in the Explorer pane. If there are no changes in the parent stream, the Explorer pane displays only the root node (and the Details pane displays a message indicating that the workspace is up-to-date).

To display all changes in the parent stream, click the root node. In the Details pane, the Element column displays the relative path of the changed element to help you quickly identify its location in the stream structure. The Action column identifies the action AccuRev will perform on the element in the workspace in order to make the change.

To display only the changes associated with a specific directory, select that directory.

You can format columns, change the sort order, and filter displayed objects in the Details pane as you would with any other table in AccuRev. See *Working with Tables* on page 14 for more information. You can also save any changes you make to the default layout as your preference. See *Saving Layout and Other Usage Preferences* on page 58 for more information.

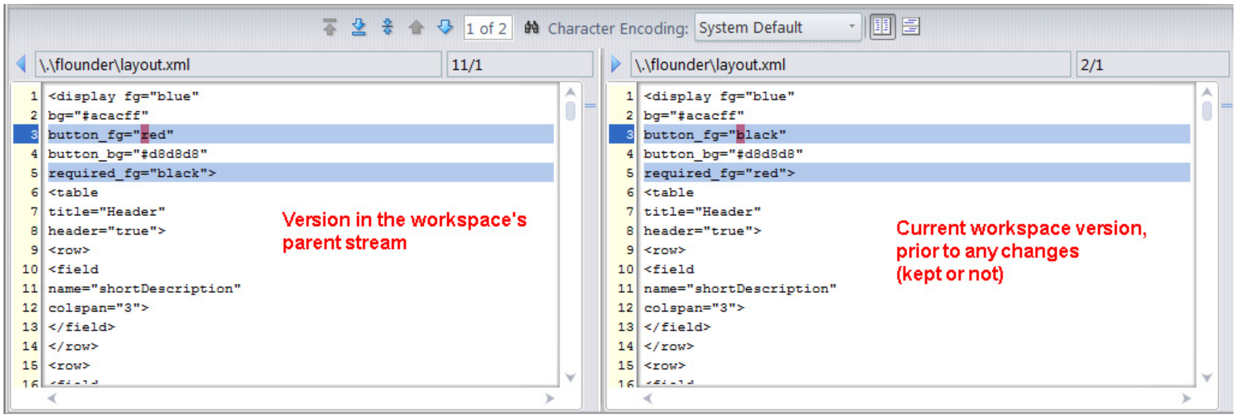
Details Pane Toolbar Reference

The following table summarizes the tools available on the Details pane toolbar and where to find more information.

Tool	Description
Annotate	Lists the contents of the selected text file, annotating each line. See *Annotate: Reviewing Transactions on an Element* on page 91.
History	Displays the transactions associated with the selected element in the History Browser. See *Chapter 5 The History Browser*.
Version Browser	Displays all the versions, and their relationships, of the selected element in the Version Browser. See *Chapter 6 The Version Browser*.

The Diff Pane

When you select an element in the Details pane, AccuRev populates the Diff pane. The panel on the left displays the real version associated with the parent stream -- this is the version that will replace the version currently in the workspace (represented in the panel on the right) when you update. Note that the panel on the right displays the current workspace version before any changes you have made, no matter how recently, and regardless of whether or not you kept them.



1 of 2

Character Encoding: System Default

\.\flounder\layout.xml 11/1

```
<display fg="blue"
bg="#acacff"
button_fg="red"
button_bg="#d8d8d8"
required_fg="black">
<table
title="Header"
header="true">
<row>
<field
name="shortDescription"
colspan="3">
</field>
</row>
<row>
```

Version in the workspace's parent stream

\.\flounder\layout.xml 2/1

```
<display fg="blue"
bg="#acacff"
button_fg="black"
button_bg="#d8d8d8"
required_fg="red">
<table
title="Header"
header="true">
<row>
<field
name="shortDescription"
colspan="3">
</field>
</row>
<row>
```

Current workspace version, prior to any changes (kept or not)

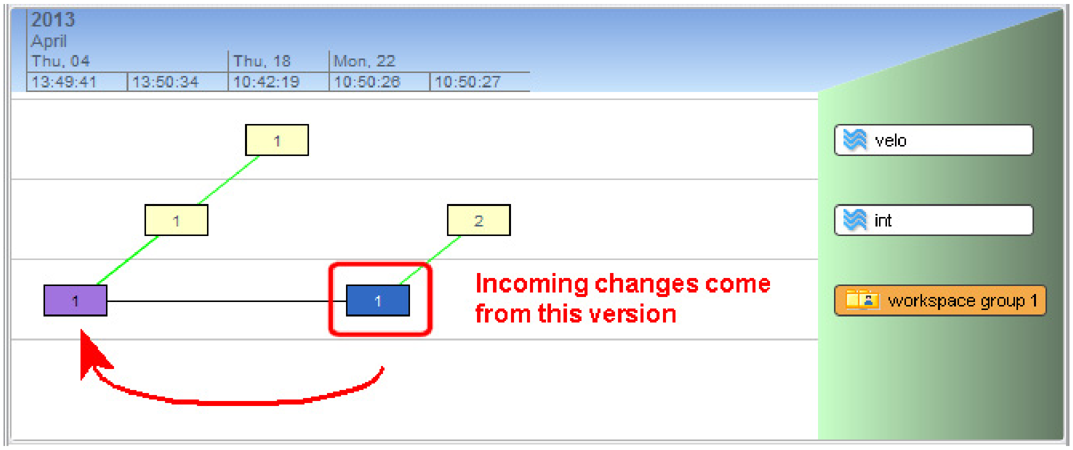
If you select a file for which no diff exists (a file with a status of **(no such elem)**, for example) AccuRev displays an informational message instead of displaying the Diff pane.

The Diff pane is read only, but the coloring, changes navigation, and other features it uses are similar to those of the AccuRev Diff tool. See *Diff Tab Layout* on page 207 for more information.

Tip: You can hide the Diff pane by clicking the **Hide Diff Pane** button (Hide Diff Pane).

Example

In this example, the values for `button_fg=` and `required_fg=` have been changed in the version in the parent stream. The Version Browser depicts the relationship between the two versions like this:



Note that AccuRev uses the real versions, created and modified in the workspace and then promoted, as the basis for comparison, and not the virtual version in the stream. (See Real Versions and Virtual Versions on page 136 for more information.)

Tip: Placing the pointer on a version icon displays a tooltip with detailed information about that version.

See *Chapter 6 The Version Browser* to learn more about the Version Browser and its features.

Updating Your Workspace

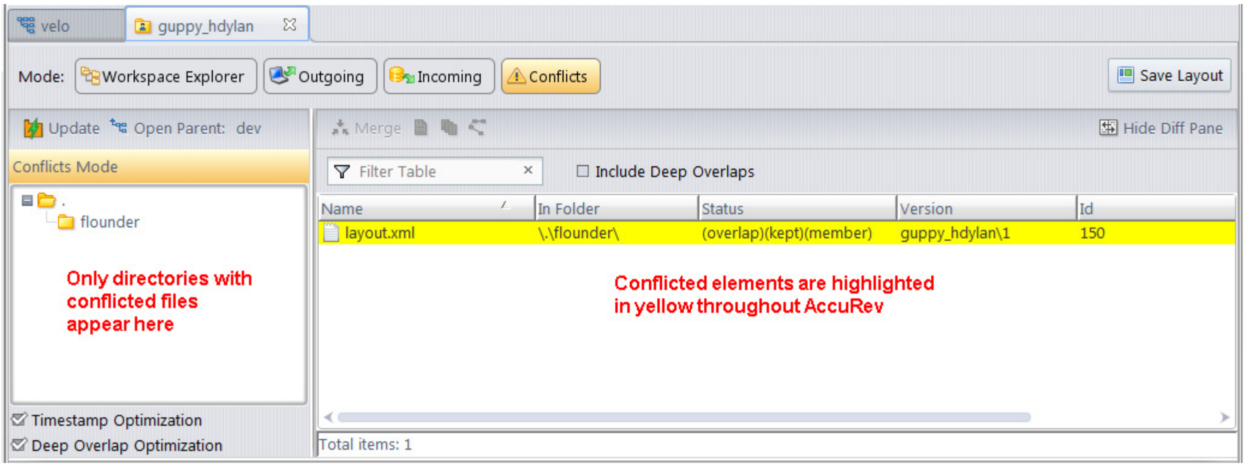
To update your workspace with the latest versions from the workspace's parent stream, click the Update button (⚡). AccuRev displays a dialog box notifying you when the update operation is complete, and the files displayed in the Details pane are replaced with a message indicating that your workspace is up-to-date.

See *Update: Keeping Your Workspace Current* on page 129 for more detailed information on keeping your workspace content up-to-date.

Tip: While you can update your workspace from any File Browser mode, consider using the Incoming Changes mode to get a preview of the changes AccuRev will make to the workspace.

Conflicts Mode

The *Conflicts mode* displays files in your workspace whose changes conflict with another version of that file elsewhere in the stream hierarchy -- often, but not always, the version in the parent stream. Overlaps, deep overlaps, and twins are all examples of the types of conflicts displayed in this mode. Note that while other File Browser modes also display conflicts, only the Conflicts mode can show elements with deep overlap conflicts.



□ Include Deep Overlaps

Name	In Folder	Status	Version	Id
layout.xml	\.\flounder\	(overlap)(kept)(member)	guppy_hdylan\1	150

Total items: 1

☑ Timestamp Optimization
☑ Deep Overlap Optimization

Only directories with conflicted files appear here

Conflicted elements are highlighted in yellow throughout AccuRev

This section covers the following topics:

- *The Explorer and Details Panes* on page 74
- *The Diff Pane* on page 75
- *Types of Conflicts Displayed in Conflicts Mode* on page 76
- *Resolving Conflicts* on page 77

See *File Browser Modes* on page 53 for more information on other modes.

The Explorer and Details Panes

The Explorer pane in Conflicts mode shows *only* those directories that are themselves in conflict or that contain one or more files that are in conflict with other versions elsewhere in the stream hierarchy. For example, if a file in the **flounder** directory conflicts with another version, the **flounder** directory is displayed in the Explorer pane.

To display all conflicts, click the root node. The In Folder column displays the relative path of the conflicted element to help you quickly identify the element’s location in the stream structure; this can be a useful feature when several directories contain conflicted elements.

To display only the conflicts occurring within a specific directory, select that directory.

You can format columns, change the sort order, and filter displayed objects in the Details pane as you would with any other table in AccuRev. See *Working with Tables* on page 14 for more information. You can also save any changes you make to the default layout as your preference. See *Saving Layout and Other Usage Preferences* on page 58 for more information.

Timestamp Optimization option. By default, AccuRev uses cached server information to identify which files in the workspace should have a status of **(modified)**. To learn more about this option and when you might want to turn it off, see *Chapter 5 Timestamp Optimization (TSO)* in the AccuRev *Administrator's Guide*. See *AccuRev Element Status Reference* on page 136 for a discussion of **(modified)** and other statuses.

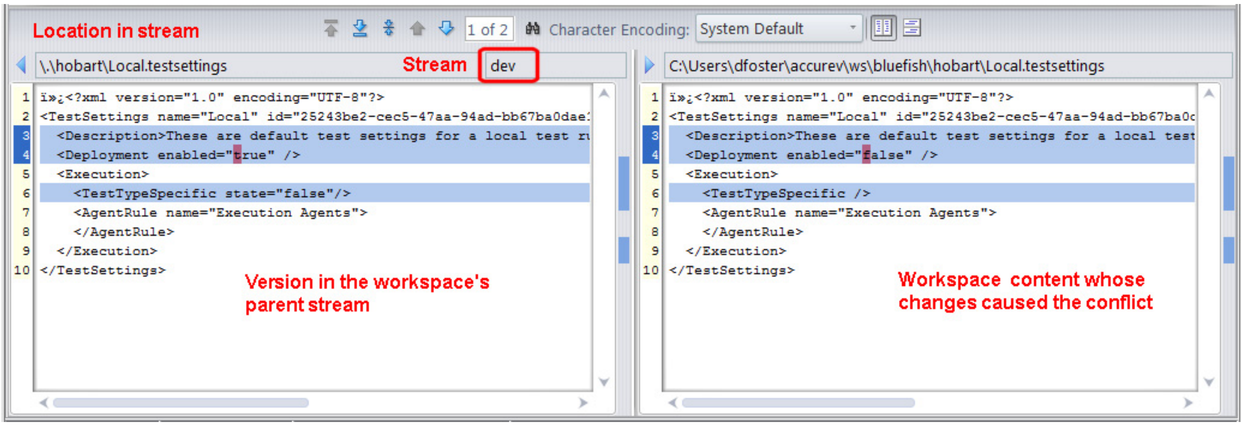
Details Pane Toolbar Reference

The following table summarizes the tools available on the Details pane toolbar and where to find more information.

Tool	Description	Available For...
Merge	Displays the Merge tool, which allows you to merge changes in conflicting versions of the same element. See *Chapter 7 Diff, Merge, Patch, and the Change Palette*.	Workspace and Stream
Annotate	Lists the contents of the selected text file, annotating each line. See *Annotate: Reviewing Transactions on an Element* on page 91.	Workspace and Stream
History	Displays the transactions associated with the selected element in the History Browser. See *Chapter 5 The History Browser*.	Workspace and Stream
Version Browser	Displays all the versions, and their relationships, of the selected element in the Version Browser. See *Chapter 6 The Version Browser*.	Workspace and Stream

The Diff Pane

When you select an element in the Details pane, AccuRev populates the Diff pane. The panel on the right displays the current workspace content whose changes conflict with a version elsewhere in the stream hierarchy; that version is displayed on the left.


Location in stream
1 of 2
Stream
Version in the workspace's parent stream
Workspace content whose changes caused the conflict

Character Encoding: System Default

\.\hobart\Local.testsettings

dev

```
1  ï»¿<?xml version="1.0" encoding="UTF-8"?>
2  <TestSettings name="Local" id="25243be2-cec5-47aa-94ad-bb67ba0dae1
3    <Description>These are default test settings for a local test ru
4    <Deployment enabled="true" />
5    <Execution>
6      <TestTypeSpecific state="false"/>
7      <AgentRule name="Execution Agents">
8      </AgentRule>
9    </Execution>
10 </TestSettings>
```

C:\Users\dfoster\accurev\ws\bluefish\hobart\Local.testsettings

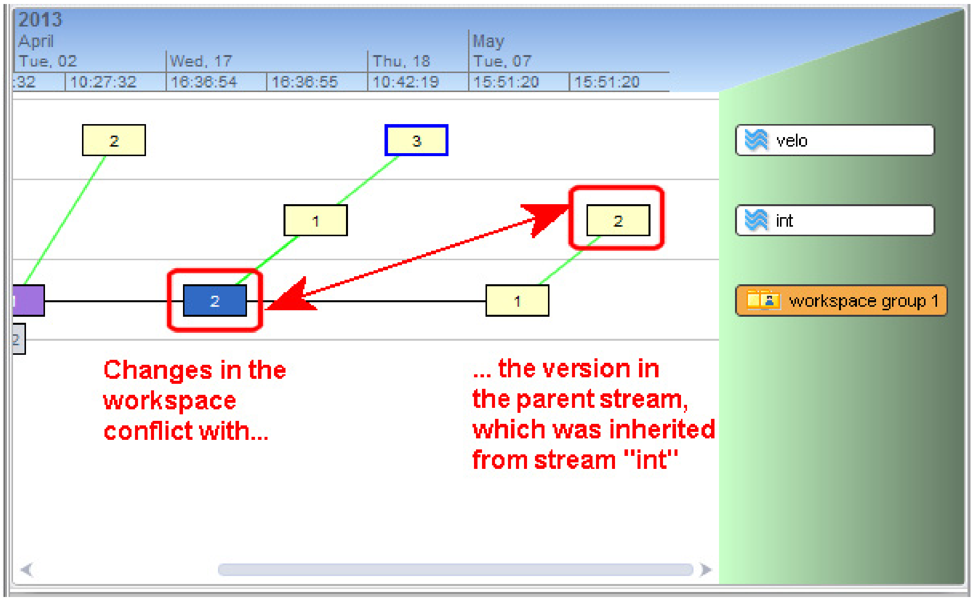
```
1  ï»¿<?xml version="1.0" encoding="UTF-8"?>
2  <TestSettings name="Local" id="25243be2-cec5-47aa-94ad-bb67ba0c
3    <Description>These are default test settings for a local test
4    <Deployment enabled="false" />
5    <Execution>
6      <TestTypeSpecific />
7      <AgentRule name="Execution Agents">
8      </AgentRule>
9    </Execution>
10 </TestSettings>
```

The Diff pane is read only, but the coloring, changes navigation, and other features it uses are similar to those of the AccuRev Diff tool. See *Diff Tab Layout* on page 207 for more information.

Tip: You can hide the Diff pane by clicking the **Hide Diff Pane** button (Hide Diff Pane).

Example

In this example, lines 3, 4, and 6 were edited in the workspace version, creating a conflict with the version in the workspace's parent stream. The Version Browser depicts the relationship between these two versions like this:



In this example, note that the Diff pane depicts the conflict between the workspace content and its parent, the *dev* stream, but that the Version Browser shows the stream where the change originated, *int*. This is because the *int* stream is the stream from which the workspace version's parent stream (*dev*) inherited the change.

Tip: Placing the pointer on a version icon displays a tooltip with detailed information about that version.

See *Chapter 6 The Version Browser* to learn more about the Version Browser and its features.

Types of Conflicts Displayed in Conflicts Mode

The Conflicts mode displays files with the following types of conflicts. Except where noted, conflicts can appear in other File Browser modes as well.

- **Overlap.** An **(overlap)** status occurs when the element has changed both in the backing stream and in your workspace. (In more technical terms, the current version in the parent stream is not an ancestor of the workspace version.) An **(overlap)** status can indicate content changes or namespace changes in the parent stream version that are not present in the selected workspace version.
- **Deep Overlap.** A deep **(overlap)** status occurs when an overlap condition exists farther up the stream hierarchy than the current parent stream -- in the *parent* stream, in the *grandparent* stream, and so on, all the way up the depot's stream hierarchy. Because searching the entire stream hierarchy for overlap conditions can take more time, the Conflicts mode does not display deep overlaps by default. See *Displaying Deep Overlaps* on page 76 for more information.
- **Twin.** A **(twin)** status occurs when two or more elements in a dynamic stream have the same pathname. This can occur in the following scenario, for example: (1) an element is defuncted in a workspace, (2) the element is promoted to the backing stream, (3) another element is created at the same pathname in the same workspace or a sibling workspace, (4) the new element is promoted to the backing stream.

Displaying Deep Overlaps

To display deep overlaps in the Conflicts mode, click the Include Deep Overlaps check box. When you do this, AccuRev refreshes the Details pane to include any elements with deep **(overlap)** status. In addition, the name of the stream associated with the conflicting version appears in the Overlap Stream column in the Details pane.

Depending on the depth and complexity of your stream structure, the search for elements with deep **(overlap)** status can take more time than a search for simple overlaps. To make the search more efficient, the Conflicts mode uses Deep Overlap Optimization, which stops searching for overlaps beyond any time basis stream in the workspace's stream hierarchy. This optimization improves performance, and also simplifies the display by not showing elements that are not relevant. If you want AccuRev to search the workspace's entire stream hierarchy for deep overlaps, clear the **Deep Overlap Optimization** check box. When you do this, any deep overlap elements beyond a time basis stream are displayed with orange instead of yellow highlighting to distinguish them from other conflict types.

Tip: The **Deep Overlap Optimization** check box setting is saved as a user preference in AccuRev's **preferences.xml** file. The setting is applied to all workspaces and streams you subsequently open in the File Browser, though you can change it any time you wish.

Resolving Conflicts

Generally speaking, the typical conflict that occurs in a workspace or a stream is an overlap -- the current version in the parent stream is not an ancestor of the version in the current workspace or stream. This can happen, for example, when users in separate workspaces are working on the same file and one of them promotes his or her changes to the workspaces' parent stream.

To resolve overlap and deep overlap conditions, you must merge your changes with the conflicting version and keep them in the workspace before you can promote your version. If the conflict is in a dynamic stream, you must perform the merge in the Change Palette. See *Chapter 7 Diff, Merge, Patch, and the Change Palette* for more information.

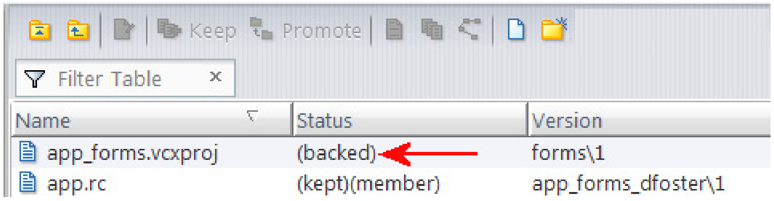
Twin elements can be resolved in the same way as overlaps -- by manually merging, keeping, and then promoting the merged version -- but the process can be a little more complicated. To help, AccuRev provides a twin resolution wizard that speeds and simplifies the twin resolution process. See *Resolving (twin) Status* on page 119 for more information.

Common Usage Scenarios

The following sections describe common usage scenarios and operations that you perform in the File Browser. More detailed information on the operations described in this section appears later in this chapter.

Editing a File

Perhaps the most common scenario is "non-conflicting development" -- modifying a file that no one else is working on concurrently and saving your work. Initially, the Status column in the details pane shows the file's status as **(backed)**: your workspace contains an unmodified copy of a version in the backing stream. The Version column contains a version-ID, indicating which backing-stream version it is:



Name	Status	Version
app_forms.vcxproj	(backed)	forms\1
app.rc	(kept)(member)	app_forms_dfoster\1

To edit a file:

1. Select the file in the Details pane.

   Note: You cannot edit a file in Incoming Changes mode.

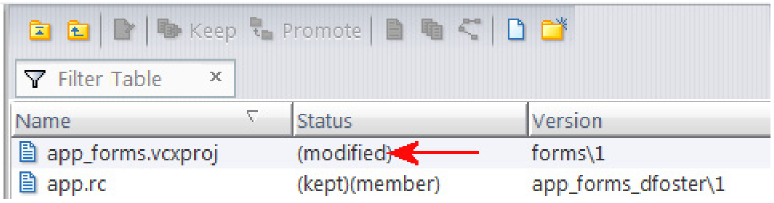
2. Click the Edit button () on the toolbar.

   Alternative: Right-click and choose **Edit** from the context menu.

   A text editor opens the selected file.

   Tip: You can use environment variable AC_EDITOR_GUI or EDITOR to control which text editor is opened. See *ENV_VARS* on page 264 of the AccuRev *CLI User's Guide* for more information.

When you close the editor, the file's status changes to **(modified)**. This indicates that you have modified the file in your workspace since the last time you synchronized the file in your workspace with the depot (with a keep or update command, for example). That is, you have changed the *file* in the local workspace, but not the *element* in the workspace stream.



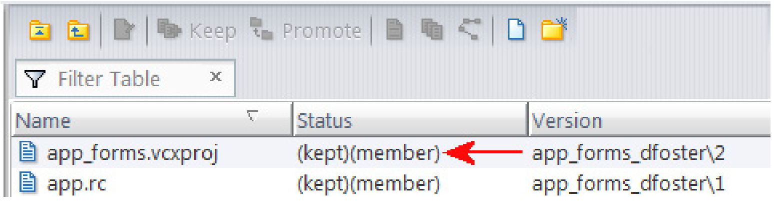
Name	Status	Version
app_forms.vcxproj	(modified)	forms\1
app.rc	(kept)(member)	app_forms_dfoster\1

You can edit the file as many times as you wish. The file's status remains **(modified)**.

Saving a File

To save your work in AccuRev, you use the Keep command. The Keep command creates a new version of the file in the workspace. The file's status changes to **(kept)(member)** to indicate that you have recorded a new version, and that the file is currently *active* in your workspace. (Active is an AccuRev term that means an element is a member of the workspace's default group. See *The Default Group of a Workspace or Stream* on page 134 for more information.)

This version is said to record a content change to the file. The File Browser also has commands for performing namespace changes. See *Renaming or Moving a File* on page 80 for more information.



Name	Status	Version
app_forms.vcxproj	(kept)(member)	app_forms_dfoster\2
app.rc	(kept)(member)	app_forms_dfoster\1

The Version column shows the version-ID of the newly created version. Note that this version is recorded in your private workspace (in this example, *app_forms_dfoster*); previously the Version column indicated that your workspace contained a version from the public backing stream (*forms*).

You can continue modifying the file and saving new versions in the depot with subsequent Edit and Keep commands. The file's status will alternate between **(modified)(member)** and **(kept)(member)**. The persistence of the **(member)** indicator reflects the fact that the file remains active in your workspace until you promote your changes to the backing stream or undo your changes.

How to Keep a File

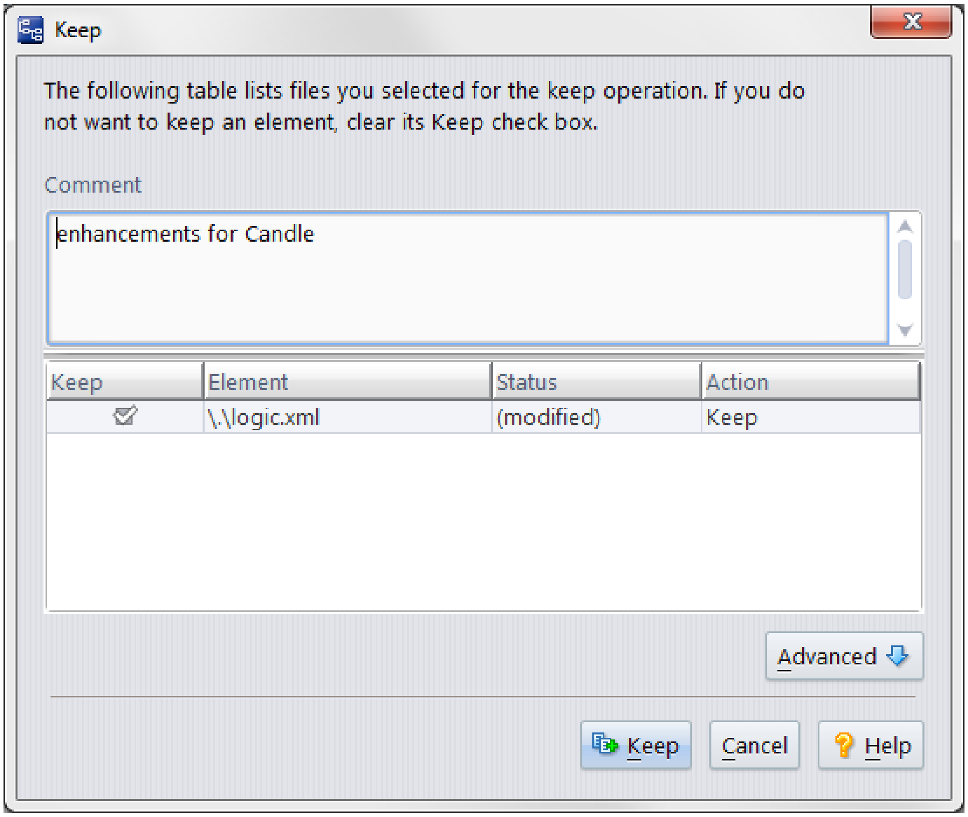
1. In the Outgoing Changes view of the workspace File Browser, select the files whose changes you want to save.

   Tip: Keep is also available in the Explorer mode. Consider using the Outgoing Changes mode because it presents all unsaved work in one place.

2. Click the **Keep** button () on the Outgoing Changes toolbar.

   Alternative: Right-click and choose **Keep** from the context menu.

   The Keep dialog box appears.



The following table lists files you selected for the keep operation. If you do not want to keep an element, clear its Keep check box.

Comment

enhancements for Candle

Keep	Element	Status	Action
☑	\.\logic.xml	(modified)	Keep

   Tip: The **Action** column lists the commands that AccuRev performs to save your changes.

3. In the **Comment** field, optionally enter a description about the version you are creating.

   Tip: AccuRev retains the comment you entered from your most recent keep.

4. Click the **Advanced** button if you want to specify a file type or enable locking.

5. If there are any files listed that you do not wish to keep at this time, clear that file's **Keep** checkbox.

6. Click the **Keep** button to save changes for the selected files.

See *Keep: Creating a New Version* on page 110 for more detailed information about **keep**.

Renaming or Moving a File

In addition to tracking changes to the contents of files, AccuRev tracks namespace changes:

- To rename an element within the same directory, invoke the *Rename* command. (You cannot use this command to rename **(external)** objects.)
- To move a file to a different directory in the depot, right-click the file and select *Cut* from the context menu. Then right-click the destination directory in the Explorer pane (not the Details pane) and select *Paste* from the context menu. You can cut-and-paste a multiple-file selection in the same way.

You can also use the Rename command to move a file to a different directory; specify a relative pathname (such as *..\otherdir\myfile.c*) as the new name for the file. See *Renaming an Element* on page 117 for more information.

AccuRev records a namespace change to a file in the same way it records a content change: by creating a new version of the file in the workspace stream. The file's status changes to **(kept)(member)**. Making a namespace change to a file activates it -- creates a new version in the workspace stream and makes it a member of the workspace's default group -- just like keeping a content change.

You can also rename and/or move a directory.

Note:

- *Status of a renamed file element: a special case*

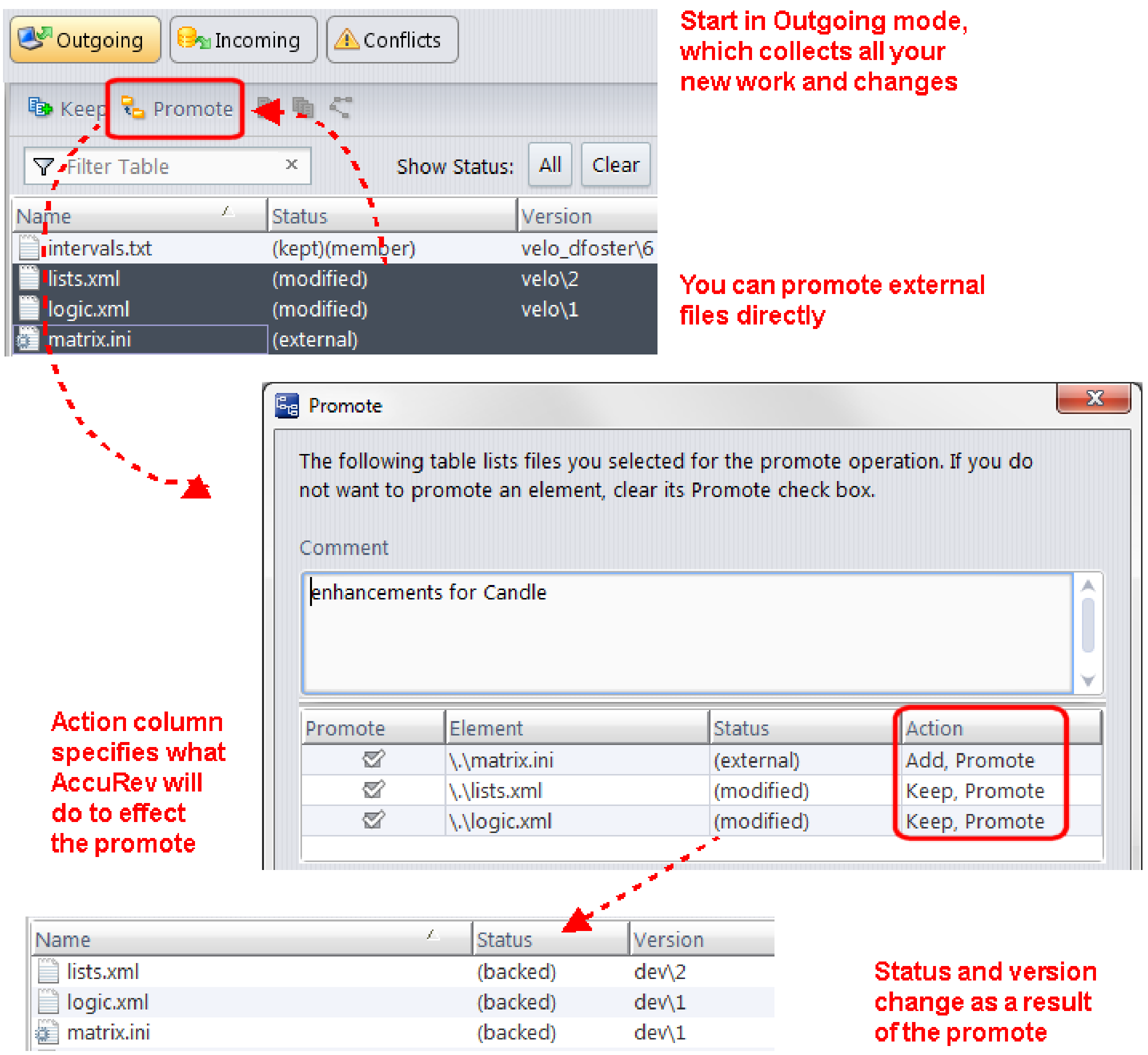
  If the file also has content changes that have not yet been saved with *Keep*, the status becomes **(modified)(member)**.

- *Reusing the old element name*

  After you rename an element, you are free to create a new element with the old name. If you do so, and then you decide to discard your renaming of the original element (*Revert to Basis* command), then the original element disappears from your workspace. (See also: *Caution on Reusing the Name of a Renamed Element* on page 118.)

Sharing Your Work with Others

Initially, the content changes and/or namespace changes you make to a file are recorded only in your workspace's private stream. This keeps your work isolated from your colleagues' work. When you are ready to share your changes with your colleagues, you *promote* the active version from your workspace stream to the backing stream. This makes your changes available to all workspaces that are based on the same backing stream.



Name	Status	Version
intervals.txt	(kept)(member)	velo_dfoster\6
lists.xml	(modified)	velo\2
logic.xml	(modified)	velo\1
matrix.ini	(external)	

Start in Outgoing mode, which collects all your new work and changes

You can promote external files directly

Promote

The following table lists files you selected for the promote operation. If you do not want to promote an element, clear its Promote check box.

Comment

enhancements for Candle

Promote	Element	Status	Action
☑	\.\matrix.ini	(external)	Add, Promote
☑	\.\lists.xml	(modified)	Keep, Promote
☑	\.\logic.xml	(modified)	Keep, Promote

Action column specifies what AccuRev will do to effect the promote

Name	Status	Version
lists.xml	(backed)	dev\2
logic.xml	(backed)	dev\1
matrix.ini	(backed)	dev\1

Status and version change as a result of the promote

Note how the version-ID in the Version column changes:

- Before the promotion, it indicates the particular version of the file that is active in your workspace. (In this example, version *velo\2 and velo\1* for **lists.xml** and **logic.xml**, respectively; **matrix.ini** has no version specified because it is external -- it has not been added to the workspace yet).
- After the promotion, it indicates that versions' newly created version-IDs in the backing stream (in this example, *dev*).

Having been promoted to the backing stream, the file is no longer active in the workspace (is no longer in the default group of the workspace stream). Accordingly, it loses its **(member)** status and returns to being **(backed)** -- its original status before you started working on the file. That is, the workspace returns to "inheriting" the version of the file currently in the backing stream -- which happens to be the version you just promoted there!

How to Promote Your Changes

You use Promote to send the version of a file containing your changes to your workspace's parent stream. This allows other users with workspaces on that same stream to get the changes in your version the next time they update their workspace. (Promote is also used to send versions from stream to stream. See

Promote: Sharing Your Work on page 113 for more information.)

To share the changes in your workspace with other users:

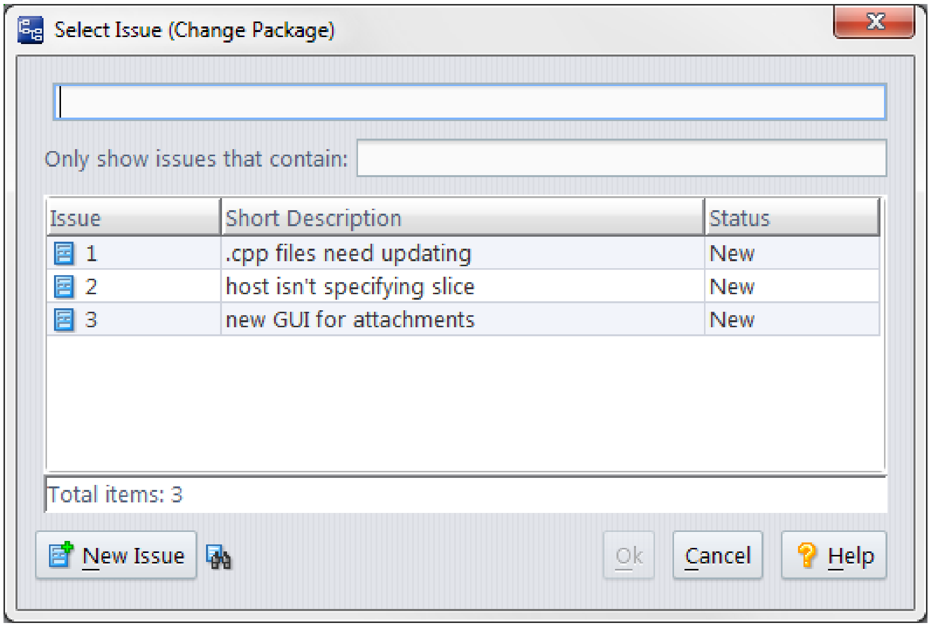
1. In the Outgoing Changes mode of the workspace File Browser, select the files whose changes you want to make available to other users.

   Tip: Promote is also available in the Workspace Explorer mode, but it can be more convenient to use the Outgoing Changes mode because this mode presents all changed work in one place.

2. Click the **Promote** button ( ) on the Outgoing Changes toolbar.

   Alternative: Right-click and choose Promote from the context menu.

   If you have enabled change packages, AccuRev displays the Select Issue (Change Package) dialog box, which allows you to associate the files you are promoting with one or more issues in your issue tracking system (AccuWork or a third-party ITS, for example).

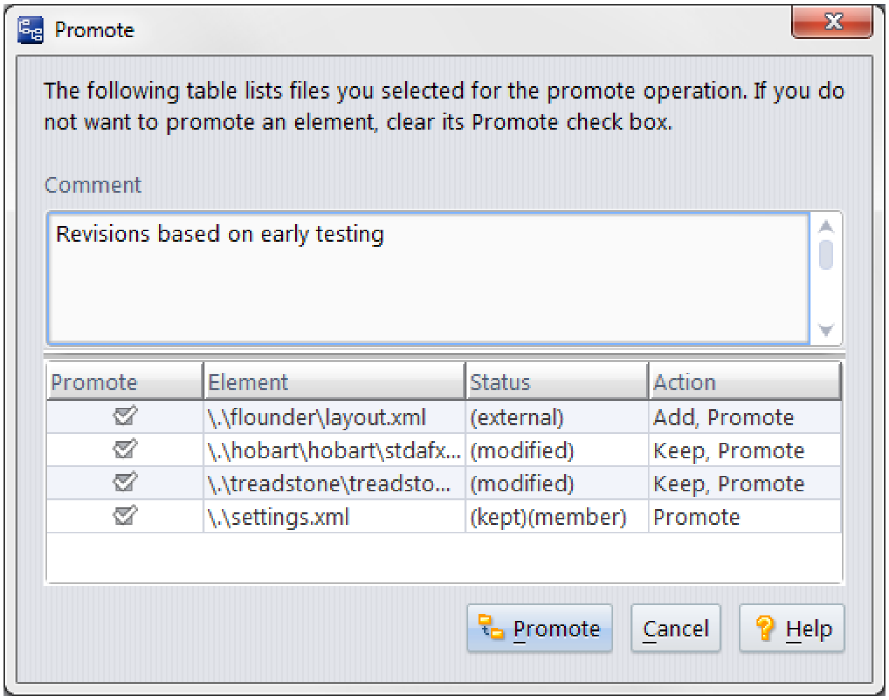


Issue	Short Description	Status
1	.cpp files need updating	New
2	host isn't specifying slice	New
3	new GUI for attachments	New

3. Select one or more issues from the list. You can also type numbers in the entry field, separated by a space. For example to specify issues 2 and 3, type *2 3*.

4. Click **Ok**.

   The Promote dialog box appears.



The following table lists files you selected for the promote operation. If you do not want to promote an element, clear its Promote check box.

Comment

Revisions based on early testing

Promote	Element	Status	Action
☑	\.\flounder\layout.xml	(external)	Add, Promote
☑	\.\hobart\hobart\stdafx...	(modified)	Keep, Promote
☑	\.\treadstone\treadsto...	(modified)	Keep, Promote
☑	\.\settings.xml	(kept)(member)	Promote

Tip: The **Action** column lists the commands that AccuRev performs to promote your version to the workspace's parent stream. Note that you can promote new work that has not yet been added to AccuRev -- the Promote operation performs the necessary Add action for you.

5. In the **Comment** field, optionally enter a description of the version you are creating.

   Tip: AccuRev retains the comment you entered from your most recent promote.

6. If there are any files listed that you do not wish to keep at this time, clear that file's **Promote** checkbox.

7. Click the **Promote** button to send the version with your changes to the parent stream.

Undoing the Changes

Inevitably, you sometimes decide *not* to share your changes with your colleagues -- instead, you can decide to discard the changes altogether (by using the *Revert to Basis* command) or to move the changes out of a parent stream but make them active in a child stream or workspace (by using the *Demote* command).

Revert to Basis

The *Revert to Basis* command (right-click the element, and then choose **Revert To... > Basis Version**) undoes all the content and/or namespace changes you have made to an active file. The file's status reverts to **(backed)**, and your workspace "rolls back" to using the version that it contained the last time the file's status was **(backed)**. It might be a version that you brought into your workspace with a recent *Update* command; or it might be a version that you created in your workspace, then *Promote*'d to the backing stream.

Notes:

- *Reverting namespace changes*

  If your changes to a file included renaming it or moving it to a different directory, invoking Revert to Basis causes the file to disappear from its new location and return to its original pathname.

- *Variant command: Revert to Most Recent Version*

  The variant command, Revert to Most Recent Version*,* is useful if you have modified a file's contents repeatedly, creating one or more intermediate versions in your workspace with the Keep command. Revert to Most Recent Version discards any content changes you have made since the most recent Keep. The file's status reverts from **(modified)** to **(kept)**. The file remains active in the workspace, so it retains its **(member)** status.

See *Revert: Undoing Changes* on page 126 for more information.

Demote

The *Demote* command moves an element version out of the default group of a stream, thereby rendering that version inaccessible in the stream. At the same time, that version is added to the default group of the child stream you specify as part of the demote process. In this way, the changes are not abandoned (as they are when you perform a *Revert to Basis* command) but, instead, are moved out of the parent stream and into a child stream.

The basic use of demote is to stop sharing changes so that the changes are no longer available to certain users. If, for example, a beta stream contains a new feature that causes the software to crash, you could demote the changes associated with that feature out of the beta stream so that your beta users no longer see those changes. The demote process enables you to remove problematic file versions from specific streams yet retain the code elsewhere so that it can be fixed and promoted again.

See *Demote: Moving Elements Out of a Stream* on page 99 for more information.

Changing a Directory

In addition to tracking changes to files, as discussed in the preceding sections, AccuRev tracks changes to directories. AccuRev's model for directory-level changes is simple, but somewhat different from the model used by the operating system (and by some other version-control systems). AccuRev considers the following to be changes to a directory:

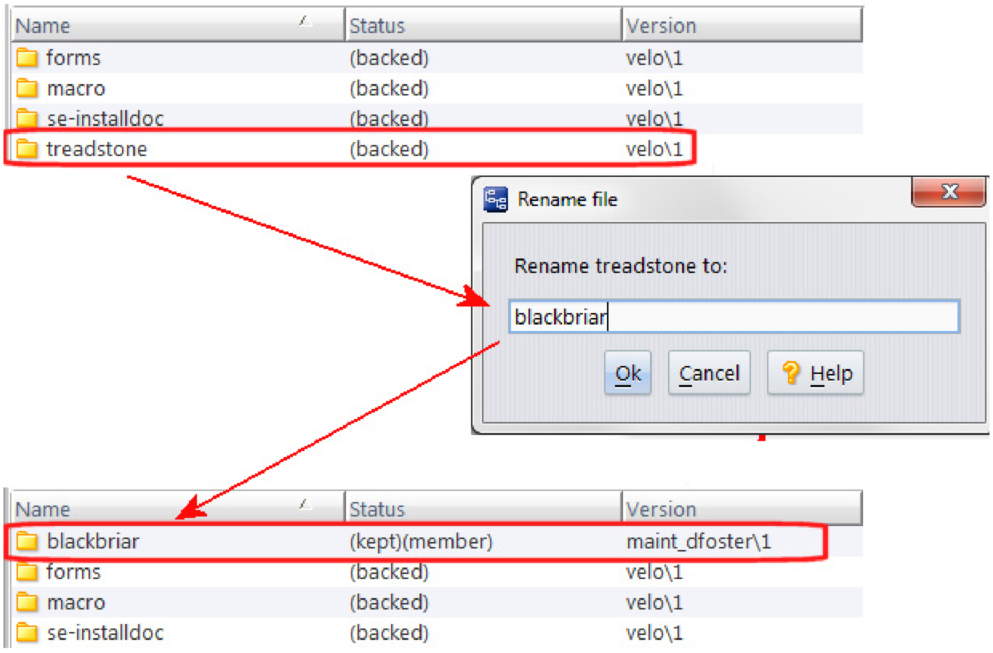
- Renaming a directory
- Moving a directory to another location in the depot's directory hierarchy
- Deleting a directory

The following are *not* changes to a directory:

- Creating a new file (it is a change to the file itself)
- Renaming an existing file (it is a change to the file itself)
- Deleting a file (it is a change to the file itself)

Note that it is only changes involving a directory's pathname that are considered to be changes to the directory itself. Changes to a directory's contents are not considered to be changes to the directory itself.

You change a directory's pathname in the same way you change a file's pathname -- with the Rename command (**Edit** > **Rename**) or with Cut and Paste commands. When you make such a change, AccuRev records a new version of the directory in the workspace stream.



Name	Status	Version
forms	(backed)	velo\1
macro	(backed)	velo\1
se-installdoc	(backed)	velo\1
treadstone	(backed)	velo\1

Rename file

Rename treadstone to:

blackbriar

Name	Status	Version
blackbriar	(kept)(member)	maint_dfoster\1
forms	(backed)	velo\1
macro	(backed)	velo\1
se-installdoc	(backed)	velo\1

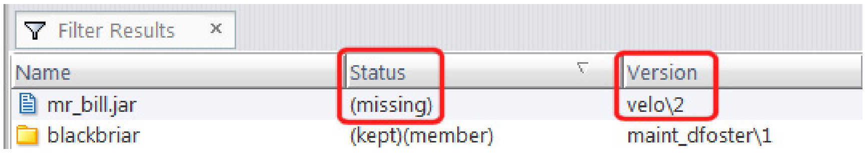
The new version-ID appears in the Version column of the Details pane (in this example, *maint_dfoster\1*). And the directory's status changes to **(kept)(member)** -- just as it does for a file's namespace change.

Deleting a File -- Accidentally or Temporarily

An AccuRev workspace tree is an ordinary directory tree, typically located on your machine's hard drive. Nothing prevents you (or perhaps, some rogue cleanup script) from using operating system commands to delete one or more of the files under version control. On occasion, you may even want to delete some files temporarily -- for example, to test the robustness of your build or installation procedure.

By definition, deleting a file at the operating system level makes it disappear from disk storage. Operating system tools, such as Windows Explorer or the UNIX *ls* command, will detect that the file no longer exists. But the file does *not* disappear from the File Browser display. AccuRev knows that the file *should* be in the workspace, because the file element still exists in the workspace's built-in stream. (The workspace stream is located in the AccuRev depot. It's unaffected by the operating system's delete-file commands.)

Accordingly, when a version-controlled file is deleted at the operating system level, the File Browser continues to list it, but indicates the file's status as **(missing)**.



Name	Status	Version
mr_bill.jar	(missing)	velo\2
blackbriar	(kept)(member)	maint_dfoster\1

Note that the Version column continues to indicate which version of the file should be in the workspace. To restore that version, invoke the Populate command from the missing file's context menu.

Deleting a directory at the operating system level is similar to (but potentially more destructive than) deleting a file. The entire directory subtree is deleted from disk storage, and the directory is listed by the File Browser as **(missing)**. To recover the entire directory tree, invoke the Populate command from the directory's context menu. Be sure to select the Recursive option from the Populate dialog.

See *Delete: Deleting Elements from a Workspace Tree* on page 98 for more information.

Deleting a File -- Intentionally and Permanently

Sometimes, you want to delete a file permanently (see note). That is, you want the file to disappear from your workspace, and from other users' workspaces, too. The file might be related to a product feature that was cancelled. Or perhaps a code reorganization rendered the file unnecessary.

Deleting a file at the AccuRev depot level (rather than simply at the operating system level) is called *defuncting*, and is implemented by the Defunct command.

Defuncting a file removes it from the workspace's disk storage (that is, deletes the file at the operating system level). In addition, the Defunct command is recorded in the depot. It may be surprising at first, but AccuRev manages the defuncting of a file in the same way as it manages the creation of new versions. To AccuRev, defuncting is just another kind of change that can happen to a file:

- Defuncting "activates" a file in the workspace stream, recording the fact that you've made a change to the file. In addition to getting **(defunct)** status, the file gets the **(kept)** and **(member)** statuses, just as if you had performed a *Keep* command.
- AccuRev records the change as a new version of the file in the workspace stream.
- At first, the defuncting of a file is isolated to your own workspace. The file continues to exist in other users' workspaces.
- To "share" the defuncting of a file, you promote the change to the backing stream. This causes the file to disappear from your workspace stream, and from the File Browser display. The file will disappear from other users' workspaces when they refresh their workspaces using the Update command.
- As always, you propagate the change -- in this case, removal of the file -- throughout the depot by promoting the defuncted file from the backing stream to the depot's higher-level streams.

Note: Defuncting is not really permanent (and the file is not even really gone)

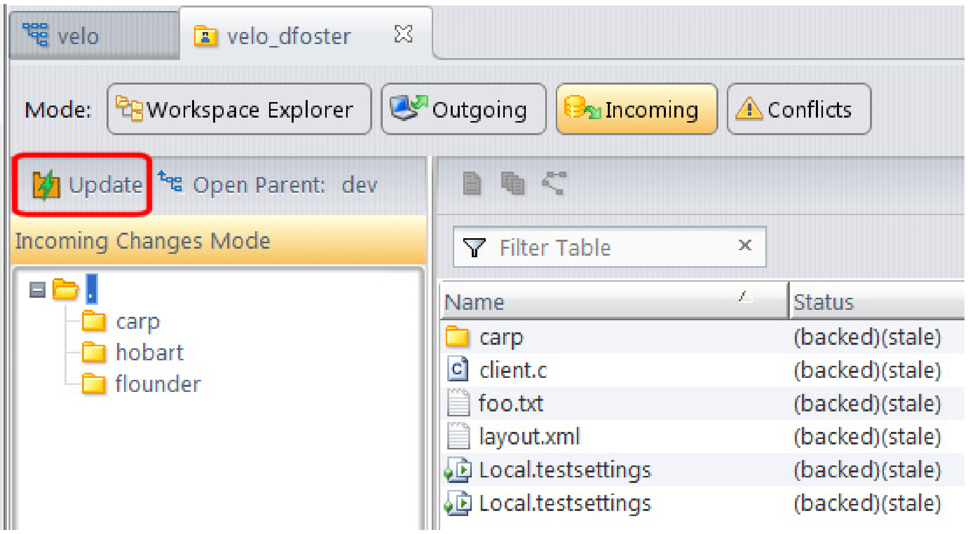
> The *Defunct* command does not actually remove the file element from the AccuRev repository. Because AccuRev is TimeSafe, old versions of the file continue to exist in the repository. You can see such versions in snapshots of streams, in History Browser displays, etc. You can restore such elements with the CLI command *undefunct*.

See *Defunct: Removing Elements from the Workspace* on page 97 for more information.

Incorporating Others' Changes into Your Workspace

As you and your colleagues make changes to a project's code base, the contents of your private workspaces diverge more and more. It's a software-development best practice to periodically incorporate others' changes into your workspace, effectively resynchronizing the workspaces (partially or fully). The *Update* command performs this task, by copying versions that your colleagues have promoted to the backing stream into your workspace.

This is an oversimplification. For some elements, you must use the *merge* operation to incorporate other's changes into your workspace. For a complete discussion of the update operation, see *Update: Keeping Your Workspace Current* on page 129.



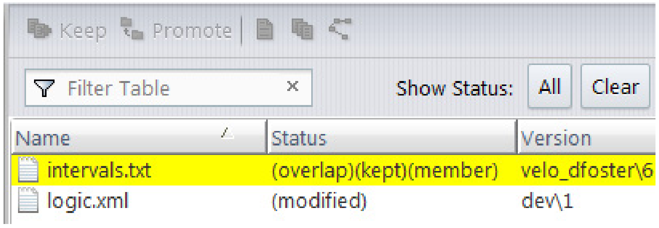
Name	Status
carp	(backed)(stale)
client.c	(backed)(stale)
foo.txt	(backed)(stale)
layout.xml	(backed)(stale)
Local.testsettings	(backed)(stale)
Local.testsettings	(backed)(stale)

Tip: The Update command is available in all File Browser modes. Consider first displaying the workspace using the Incoming mode to get a preview of the changes that will be made to your workspace. See *Incoming Changes Mode* on page 70 for more information.

Merging Your Changes with Someone Else's Changes

AccuRev supports concurrent development: two or more users can start with the same version of a file and make changes to that file independently -- both content changes and namespace changes. After one of the users promotes his changes to the backing stream, each of the others must merge her own changes with the newest version in the backing stream, using the ***Merge*** tool. We describe here how to initiate a merge operation.

As you work on a file, keeping intermediate versions in your workspace, the file's status alternates between **(modified)** and **(kept)**. At any point, you may notice that an additional indicator, **(overlap)**, appears in the Status column. To make sure you notice, the File Browser displays the entry with a yellow highlight.

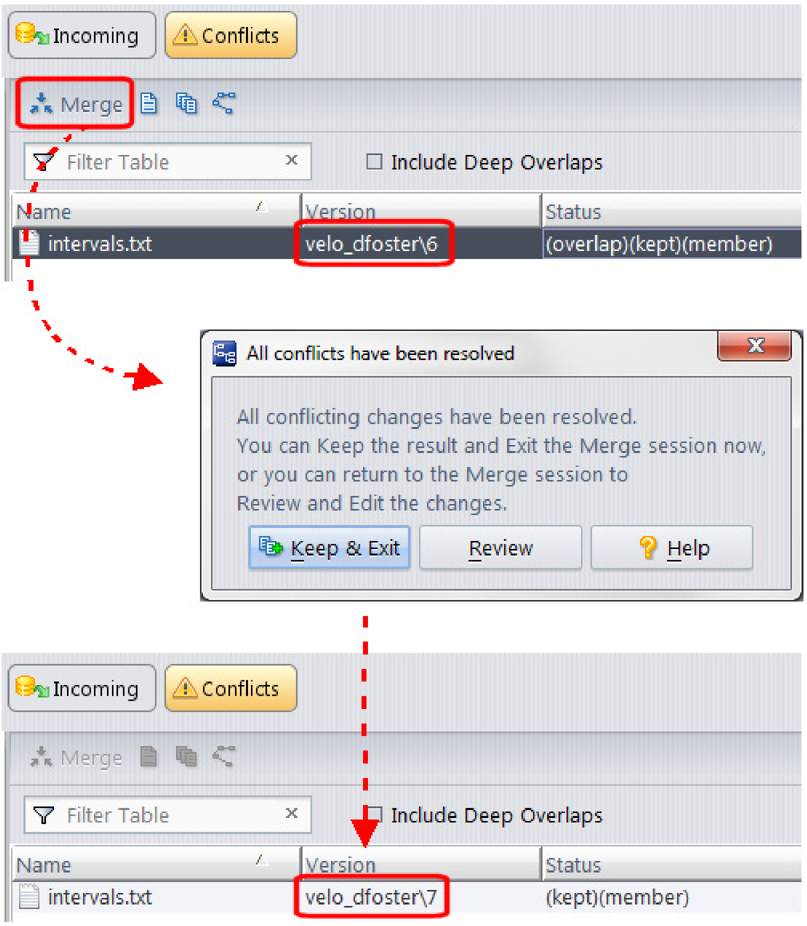


Name	Status	Version
intervals.txt	(overlap)(kept)(member)	velo_dfoster\6
logic.xml	(modified)	dev\1

This means that a new version of the file has been promoted to the backing stream. Typically, one of your colleagues has edited the file and promoted her version to the backing stream. It may also be that someone has promoted a new version of the file to a higher-level stream, and the backing stream dynamically inherits the new version from its parent stream.

Tip: Elements with a **(twin)** status are also highlighted in yellow. See *(twin)* on page 138 for more information. Consider using the File Browser Conflicts mode to display all overlap and twin elements in your workspace. See *Conflicts Mode* on page 73 for more information.

Whenever a file's status is **(overlap)**, the *Merge* command is enabled. The execution of a *Merge* command concludes with AccuRev prompting you to keep the new version in your workspace.



Name	Version	Status
intervals.txt	velo_dfoster\6	(overlap)(kept)(member)

All conflicting changes have been resolved.
You can Keep the result and Exit the Merge session now,
or you can return to the Merge session to
Review and Edit the changes.

Name	Version	Status
intervals.txt	velo_dfoster\7	(kept)(member)

When you promote this merged version to the backing stream, the file's status returns to **(backed)**.

See *Merge* on page 214 for more information.

Note: Underlap status

The status **(underlap)** is similar to **(overlap)**, in that an element has changed in your workspace and also in the backing stream. With **(underlap)**, the changes in your workspace version have already been promoted to the backing stream (from another workspace, or from a stream elsewhere in the depots stream hierarchy). You can resolve an underlap in the same way as an overlap -- by merging and promoting. But in many cases, it is more appropriate to use the *Revert to Basis* command to undo the changes in your workspace.

Revert: An Alternative to Merge

Sometimes you might simply wish to undo the changes that created the **(overlap)** condition. In AccuRev, you accomplish an "undo" using the *Revert* command. Revert lets you:

- Remove all changes that you have made to these elements in your workspace, or
- Remove the change you made since you last did a keep on the file in conflict

See *Revert: Undoing Changes* on page 126for more information.

Anchor: Preventing Files from Being Overwritten

The ***Anchor*** command makes the selected files *active* in the workspace (places them in the workspace's default group), without modifying them. Anchor creates a new version in the workspace stream. This new version, called a *virtual version*, simply records the fact that the file is officially active in the workspace. See *Real Versions and Virtual Versions* on page 136 for more information.

Typically, you anchor a file in your workspace to prevent it from being overwritten with a newer version by a subsequent ***Update*** command. (***Update*** overwrites inactive files only, not active ones.) However, there are some situations in which you *must* anchor a file before editing it:

- You are working in an anchor-required or exclusive file locking workspace, or
- A development lock has been placed on that particular file element.

The Send to Workspace command is a variant of ***Anchor***. Instead of activating the version that is currently in your workspace, ***Send to Workspace*** can activate any version of the element. See *Send to Workspace* on page 127 for more information.

Invoking the Anchor Command

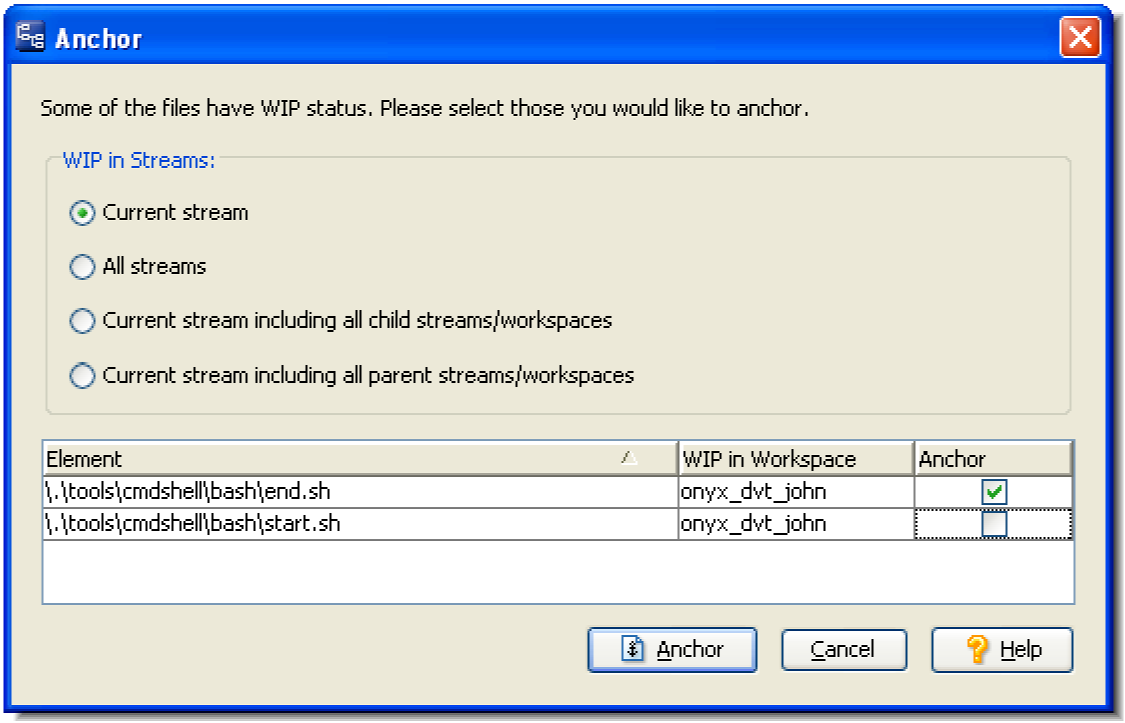
In the Details pane of the File Browser, select a file element whose current status is **(backed)**. Then, right-click and choose ***Anchor*** from the context menu.

In either of the must-anchor situations described above, if any element to be anchored is currently active in any sibling workspace, the ***Anchor*** dialog box appears, to help you complete the command.

Note: Anchoring an element link or symbolic link operates on the link itself.

Using the Anchor Dialog

When you anchor a set of elements in an anchor-required or exclusive file locking workspace, a dialog appears if one or more of them is active in a sibling workspace. This enables you to select/deselect individual elements to be anchored. Note that the elements that are active in the sibling workspace are initially deselected. This makes it easy to avoid the situation where multiple users are working on the same file(s) concurrently. The exclusive file locking feature guarantees that this situation won't occur



Some of the files have WIP status. Please select those you would like to anchor.

WIP in Streams:

● Current stream

○ All streams

○ Current stream including all child streams/workspaces

○ Current stream including all parent streams/workspaces

Element	WIP in Workspace	Anchor
\.\tools\cmdshell\bash\end.sh	onyx_dvt_john	☑
\.\tools\cmdshell\bash\start.sh	onyx_dvt_john	□

Annotate: Reviewing Transactions on an Element

The ***Annotate*** command lists the entire contents of the selected version of a text file, annotating each line with:

- The timestamp of the version in which that line was added to the file.
- The number of the transaction in which that version was created.
- The user who performed that transaction.

A version timeline makes it easy to view the contents of *any* version of the text file.

Opening an Annotate Tab

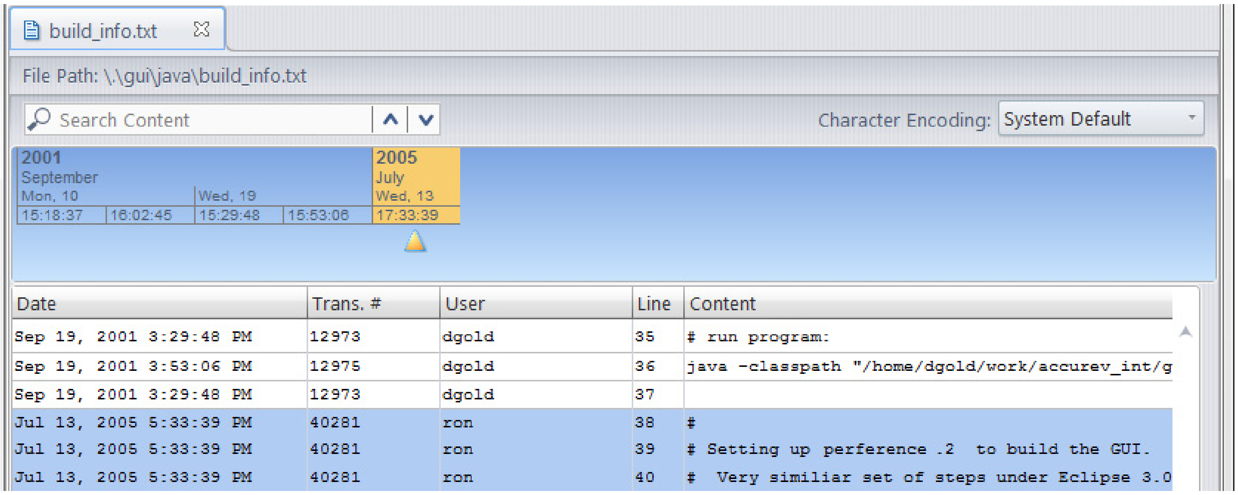
In the Details pane of the File Browser, select one text-file element. Then, invoke the command in any of these ways:

- Click the ***Annotate*** toolbar button.
- Right-click the selection to bring up its context menu, then select ***Annotate***.

Note: Annotating an element link or symbolic link operates on the target of the link.

Annotate Tab Layout

The contents of a text file are listed as a multiple-column table. Each row represents one line of the text file.



File Path: \.\gui\java\build_info.txt

Search Content

Character Encoding: System Default

Date	Trans. #	User	Line	Content
Sep 19, 2001 3:29:48 PM	12973	dgold	35	# run program:
Sep 19, 2001 3:53:06 PM	12975	dgold	36	java -classpath "/home/dgold/work/accurev_int/g
Sep 19, 2001 3:29:48 PM	12973	dgold	37	
Jul 13, 2005 5:33:39 PM	40281	ron	38	#
Jul 13, 2005 5:33:39 PM	40281	ron	39	# Setting up perference .2 to build the GUI.
Jul 13, 2005 5:33:39 PM	40281	ron	40	# Very similiar set of steps under Eclipse 3.0

The columns contain:

- **Date** column: the timestamp of the version in which that line was added to the file.
- **Trans. #** column: the number of the transaction in which that version was created.
- **User** column: the AccuRev username of the user who created that version.
- **Line** column: the line number of the text.
- **Content** column: the contents of the text line.

In a **(modified)**-status file, one or more lines have not yet been saved with the ***Keep*** command. For such lines, the **Date** and **User** columns are empty and the Trans # column displays *Modified*.

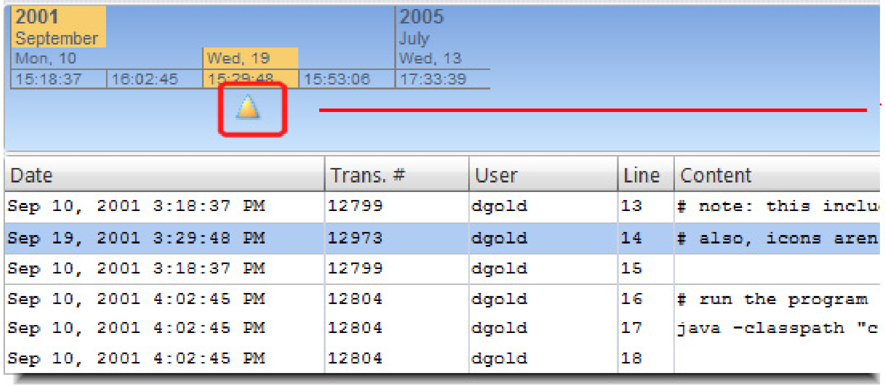
Working in an Annotate Tab

When you click on a cell in the Date, #, or User column, AccuRev highlights all the lines whose value in that column matches the value in the cell. This makes it easy to:

- Find all the lines added or changed in a particular version (click in the Date or Transaction # column).
- Find all the lines created by a particular user (click in the User column).

Using the Version Timeline

A timeline appears at the top of the Annotate tab, with an entry for each version of the text file element.



drag slider along the transaction timeline

Date	Trans. #	User	Line	Content
Sep 10, 2001 3:18:37 PM	12799	dgold	13	# note: this inclu
Sep 19, 2001 3:29:48 PM	12973	dgold	14	# also, icons aren
Sep 10, 2001 3:18:37 PM	12799	dgold	15	
Sep 10, 2001 4:02:45 PM	12804	dgold	16	# run the program
Sep 10, 2001 4:02:45 PM	12804	dgold	17	java -classpath "c
Sep 10, 2001 4:02:45 PM	12804	dgold	18	

You can use the timeline's slider control to switch among the file's versions. (Alternatively, just click a timeline entry to jump to that version.) Whenever you switch to a different version, AccuRev automatically highlights the lines that were added or changed in that version.

Creating a New File

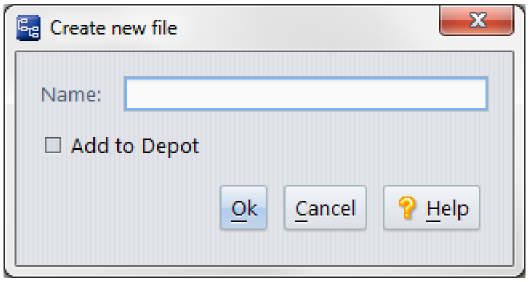
The ***New File*** command creates a new empty file in the current directory (the directory that's highlighted in the Workspace Explorer) of your workspace tree. It can also add the directory to the depot -- that is, place the directory under version control by converting it to an element.

If you choose not to place the file under version control, you can do so later, with the ***Keep*** command, for example.

To create a new file:

1. In the Workspace Explorer of the ***File Browser***, navigate to the directory in which you want to create the new file. Then, click the 🗋 ***New File*** button in the Details pane toolbar.

   The Create New File dialog box appears:

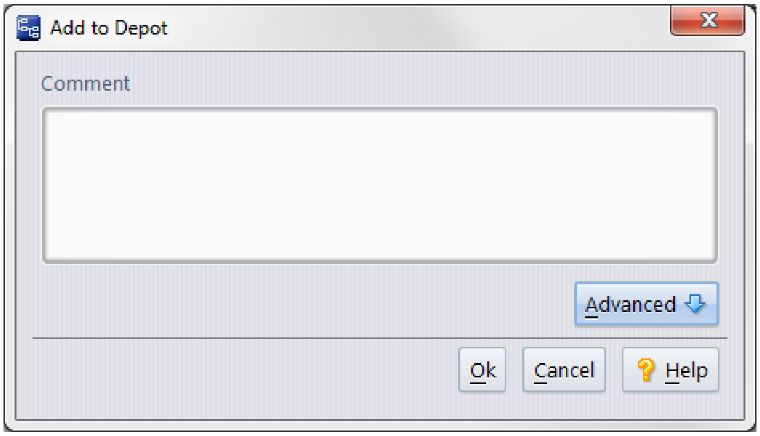


2. In the **Name** field, enter the simple name (leaf name) of the new file.
3. Do you want to place the new file under AccuRev control right now?

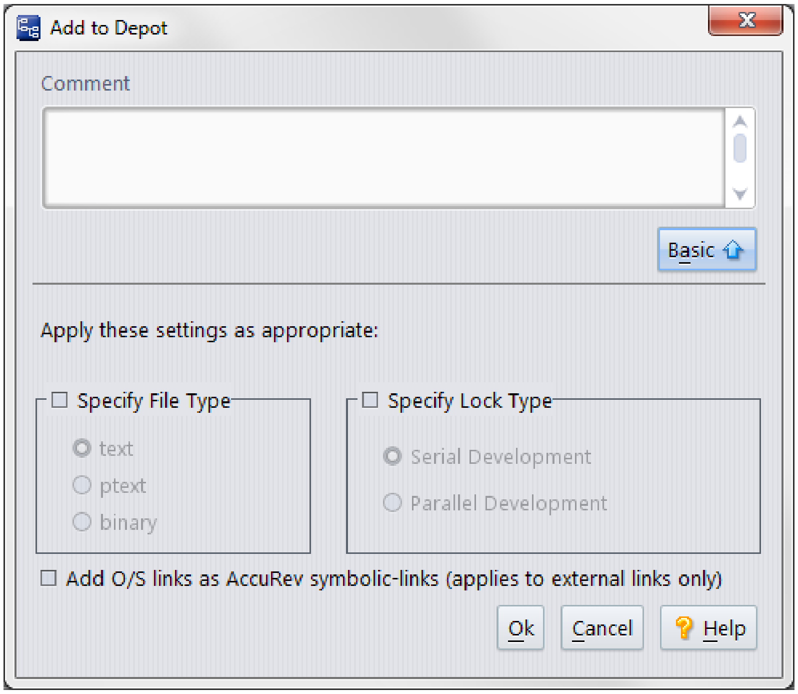
   If *no*, click **Ok** to just create the new file in your workspace. (You can always do this later using the Keep command, for example.)

   If *yes*, select the **Add to Depot** check box and click **Ok**.

   The Add to Depot dialog box appears.



4. Optionally enter a comment to document your reason for creating this file and placing it under AccuRev control. The comment string becomes a permanent annotation to the version you're creating. (AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a trigger on the Add to Depot command.)
5. Click the **Advanced** button to display an extended panel that lets you specify the file type, lock type, and link creation.



Apply these settings as appropriate:

☐ Specify File Type

● text
○ ptext
○ binary

☐ Specify Lock Type

● Serial Development
○ Parallel Development

☐ Add O/S links as AccuRev symbolic-links (applies to external links only)

- **File Type.** By default, AccuRev determines the element type for the newly created version automatically (**text** or **binary**). You can override the default by specifying either of these types or **ptext** (a variant of **text**). See *AccuRev Element Types* on page 111.
- **Lock Type.** Choosing **Serial Development** places a lock on the element to ensure that users in sibling workspaces won't have to merge their work on this element. The lock remains on the element until you (or someone else) changes this value to **Parallel Development** in a subsequent ***Keep*** command. See *The Locks Dialog Box* on page 52.
- **Symbolic Links.** By default, a symbolic link (UNIX/Linux) or junction point (Windows) is added to the depot as an AccuRev element link, not as an AccuRev symbolic link. The target of the link must be a directory element visible in your workspace. Use this checkbox to cause such objects to be converted to AccuRev symbolic links.

  Note: Windows junction points are not supported on FAT or FAT32 file systems.

6. Click **Ok** to add the new file to AccuRev with the options you specified.

Creating a New Folder

The ***New Folder*** command creates a new, empty folder (directory) in the current directory (the directory that's highlighted in the Workspace Explorer) of your workspace tree. It can also add the directory to the depot -- that is, place the directory under version control by converting it to an element.

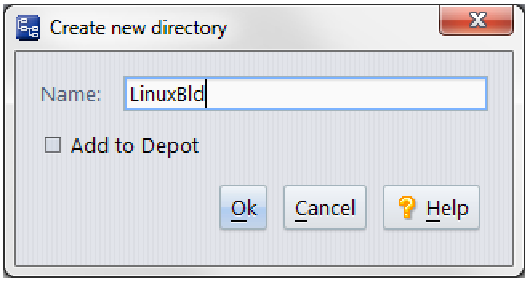
If you choose not to place the directory under version control, you can do so later ...

- explicitly, with the ***Keep*** command, for example
- implicitly, when you place any file or subdirectory below the new directory under version control

To create a new folder:

1. In the Workspace Explorer, navigate to the directory in which you want to create the new directory. Then, click the ***New Folder*** button in the Details pane toolbar.

   The Create New Directory dialog box appears.



Name: LinuxBld

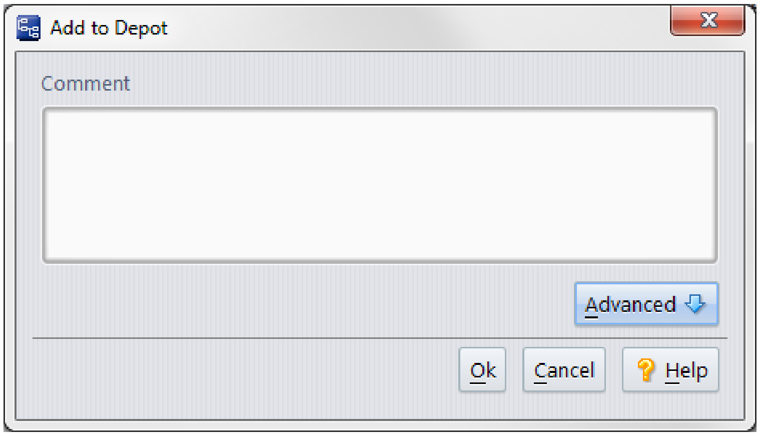
☐ Add to Depot

2. In the **Name** field, enter the simple name (leaf name) of the new directory.
3. Do you want to place the new directory under AccuRev control right now?

   If *no*, click **Ok** to just create the new directory in your workspace. (You can always do this later using the Keep command, for example.)

   If *yes*, select the **Add to Depot** check box and click **Ok**.

   The Add to Depot dialog box appears.



4. Optionally enter a comment to document your reason for creating this directory and placing it under AccuRev control. The comment string becomes a permanent annotation to the version you're creating. (AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a trigger on the Add to Depot command.)
5. Click the **Advanced** button to display an extended panel that lets you specify the lock type. Choosing **Serial Development** places a lock on the element to ensure that users in sibling workspaces won't have to merge their work on this element. The lock remains on the element until you (or someone else) changes this value to **Parallel Development** in a subsequent ***Keep*** command. See *The Locks Dialog Box* on page 52.

Cut, Copy, and Paste

This section describes how to use cut, copy, and paste operations on files, directories, and link. These operations are available in the workspace File Browser only.

Cut

Mark the currently selected element to be moved to another directory in the same workspace. To finish the relocation, right-click the destination directory (in either the Folders or Details pane) and select *Paste*.

Tip: You can use the Ctrl+x keyboard shortcut for *Cut*.

Copy

Designate the selected object (file, directory, or link) to be the target of an element link or a symbolic link. Use the *Paste* command to complete the creation of the link object.

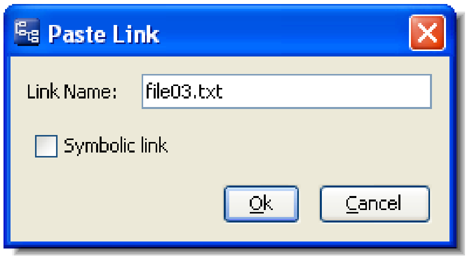
Tip: You can use the Ctrl+c keyboard shortcut for *Copy*.

Paste

The behavior of Paste varies slightly based on whether you used *Cut* or *Copy* to place the element on the clipboard.

- Paste after Cut -- Specify the destination for an element that has been marked for relocation with the *Cut* command.
- Paste after Copy -- Create an element link or symbolic link in the selected directory. The target of the link is the file, directory, or link element on which you previously invoked the ***Copy*** command.

  You can create a link that points to another link. A chain of links can be arbitrarily long, but must end at some file element or directory element. By default, the link object has the same name as the target object. But you can change the name in the Paste Link dialog, which appears after you perform the **Copy** operation.

  Paste Link

  Link Name: file03.txt

  □ Symbolic link

  If the name you specify does not currently exist, AccuRev creates a new link element (**add** transaction). If the name you specify is an existing link element, AccuRev changes the target of that link element (**keep** transaction).

Tip: You can use the Ctrl+v keyboard shortcut for *Paste*.

Defunct: Removing Elements from the Workspace

The *Defunct* command removes elements from active use in your workspace. That is, for each element you specify, it:

- removes the file, directory, or link from your workspace tree
- marks the element as **(defunct)** in the workspace stream
- makes the element active in the workspace, by adding it to the workspace's default group

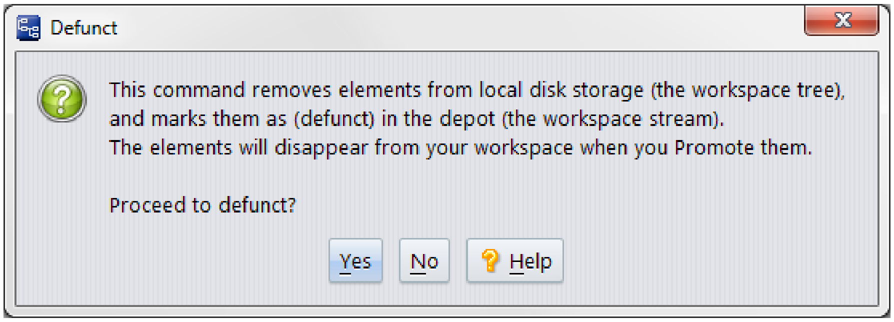
Operating system commands won't find a defunct element (because it really is gone from the workspace tree), but the File Browser can see the element in your workspace stream, where it is both **(defunct)** and a **(member)**.

Defunct does not remove an element from the depot altogether (no operation removes an element altogether -- that would violate AccuRev's TimeSafe property). And it does not make an element disappear for all users. *Defunct* just removes an element from a particular workspace. The element remains visible in other streams and workspaces -- at least until you Promote it.

Invoking the Defunct Command

In the Details pane of a File Browser that is open on a workspace, select one or more file elements. Then right-click the selection and choose ***Defunct*** from the context menu.

The ***Defunct*** dialog appears, prompting you to confirm your intention to remove the element(s).



This command removes elements from local disk storage (the workspace tree),
and marks them as (defunct) in the depot (the workspace stream).
The elements will disappear from your workspace when you Promote them.

Proceed to defunct?

Defuncting a Directory

If any of the elements you specify is a directory, D*efunct* works recursively: it removes the directory itself and elements under that directory. Only the specified directory itself becomes **(defunct)**; the objects below it are simply removed from the workspace tree, but do not become **(defunct)** in the workspace stream.

The precise result depends on whether any elements located below the defuncted directory are active:

- If no element below the defuncted directory is active, the entire directory tree is removed from your workspace tree.
- If one or more elements below the defuncted directory is active, those elements become stranded. A stranded element is *not* removed from your workspace tree, but its status becomes **(external)**, since the workspace stream no longer has a valid pathname to the object.

  Caution: In all cases of defuncting a directory, a file below the defuncted directory that you have edited -- but never preserved with ***Keep*** -- will be removed from the workspace tree. (Such files are not officially "active" in the workspace.) This removes data for which there might be no other copy.

Promoting a Defunct Element

After defuncting an element, you use *Promote* to propagate its removal to your workspace's backing stream. The element won't appear in a File Browser that is open on your workspace, but it will appear, with **(defunct)** status, in a File Browser that is open on the backing stream.

Bringing Back a Defunct Element

You can usually recover the backed version of the **(defunct)** element by invoking the ***Revert to Basis*** command on it. However, if either of the following conditions apply, the correct way to recover the version is to invoke the ***accurev undefunct*** command from the CLI.

- If you have kept changes in your workspace before defuncting. Remember that if you ***Revert to Basis*** an element which has not been promoted, you eliminate the version of that element in your workspace.
- If you have promoted the **(defunct)** element, so that it no longer appears in the File Browser.

The Past and Future of a Defunct Element

When a user in a sibling workspace performs an ***Update***, the element will disappear from that workspace.

With each successive promotion up the stream hierarchy, a defunct element disappears from the source ("promote from") stream, and becomes active, with **(defunct)** status, in the destination ("promote to") stream.

AccuRev's TimeSafe property means that you cannot change the past. This means that a defunct element remains in old snapshots of streams. You can always get information about the element (if you know which stream it still exists in) using the ***History Browser***.

Delete: Deleting Elements from a Workspace Tree

The ***Delete*** command removes an object -- file, directory, or link -- from a workspace tree. It's equivalent to using an operating system command or tool (e.g. Windows Explorer, UNIX ***rm*** command) to delete the object. This command does not make any change in the AccuRev repository.

If the object is under version control (that is, it's an AccuRev element), its status becomes **(missing)**. If not, the object is simply removed from your workspace's local disk storage.

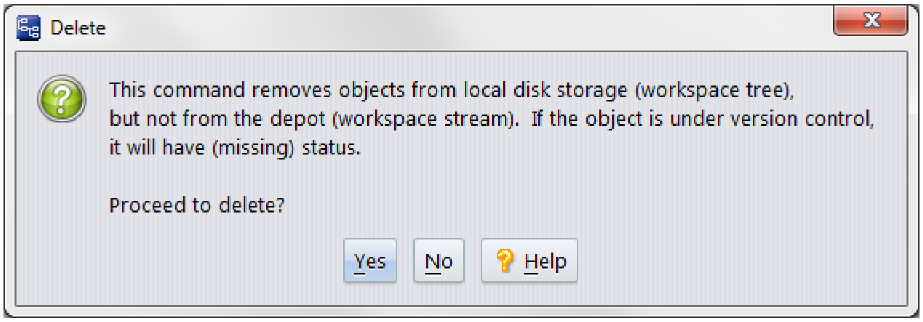
Tip: To remove an element from the depot's stream hierarchy, so that its removal will be reflected in other users' workspaces, use the *Defunct* command instead of *Delete*. See *Defunct: Removing Elements from the Workspace* on page 97 for more information.

Invoking the Delete Command

In the ***Details*** pane of a ***File Browser*** that is open on a workspace, select one or more objects. Then:

- Choose **Edit** > **Delete** from the main menu, or
- Right-click the selection and choose **Delete** from the context menu.

The **Delete** dialog appears, prompting you to confirm your intention to remove the object(s).



This command removes objects from local disk storage (workspace tree), but not from the depot (workspace stream). If the object is under version control, it will have (missing) status.

Proceed to delete?

Demote: Moving Elements Out of a Stream

The ***Demote*** command moves one or more element versions out of a parent stream (the *source stream*), while making those versions accessible in a child stream (the *destination stream*) -- and in workspaces and streams below it.

The basic use of demote is to stop sharing changes in a given stream by moving those changes out of the stream. A demote operation is different from a *Revert to Basis* operation in that the changes are not completely abandoned. Instead, the current file versions are retained in the child stream that you specify during the demote process.

Suppose, for example, you made some changes to the matrix.ini file and promoted your changes to the Dev stream, with the result that your changes were causing software builds to fail. You could demote those changes from the Dev stream to a child stream, which would enable your builds to succeed while you fixed the problematic file in another location. Once you had made the appropriate corrections, you could promote the matrix.ini file to the Dev stream once more.

The Demote option is available in several places in the AccuRev GUI, such as the StreamBrowser and in the Conflict mode of the Stream View Explorer.

When considering a demote operation, keep the following points in mind:

- The destination stream must be an immediate child of the source stream. You cannot demote to a non-child stream nor can you demote more than one level beneath the source stream.
- You cannot demote from a workspace, since it has no valid destination streams beneath it.
- You can only demote the head version of an element.
- You cannot demote elements that have dependencies or that have an overlap status in the destination stream.
- A demote operation does not overwrite the element version in the destination stream if that version is newer than the version contained in the source stream.
- Just as you can promote by file, transaction, or issue, you can also demote by file, transaction, and issue.
- AccuRev recommends demoting by transaction as a best practice, since you can retain previous versions of the element in the source stream which you cannot do when demoting by file.

Demote and Entity Types

The following table indicates whether it is possible to demote to or from various AccuRev entities.

AccuRev Entity	Demote from?	Demote to?
Dynamic stream	Yes	Yes
Time-based stream	Yes	Yes
Snapshot stream	No	No
Pass-through stream	No	No
Gated stream	No	No
Workspace	No	Yes
Depot (root)	Yes	No
Reference tree	No	No

Demote by Transaction

Rather than demoting individual files, you can also demote a set of element versions by demoting a transaction. When you demote by transaction, the specific versions of the files in that transaction are no longer available in the source stream but they are now accessible in the destination stream.

The process of demoting by transaction enables you to leave previous versions of demoted elements active in the source stream. You cannot do this when demoting by file.

For example, suppose that you make three successive changes to a file and promote it after each change. If the file is problematic, demoting it by file means that the source stream now contains the version of the file that is in the backing stream. If, instead, you demote by transaction, you can demote the most recent transaction which leaves the previous version of the file active in the source stream.

The ability to demote a single transaction while leaving previous transactions in the source stream means that you can exercise a finer granularity of control over the intermediary versions of a file contained in the source stream than when demoting by file. For this reason, AccuRev recommends demoting by transaction as a best practice.

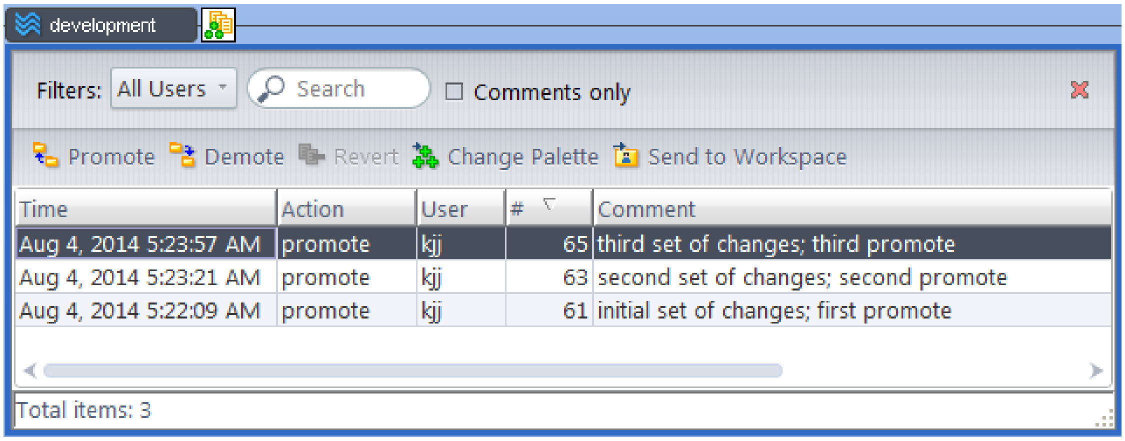
How to Demote by Transaction

To demote a transaction:

1. In the StreamBrowser, specify Active Transactions as the Active Group Mode.
2. Click the Show Active Transactions button to the right of the source stream.

   Alternative: Right-click the source stream and choose **Show Active Transactions** from the context menu.

AccuRev displays a list of transactions:

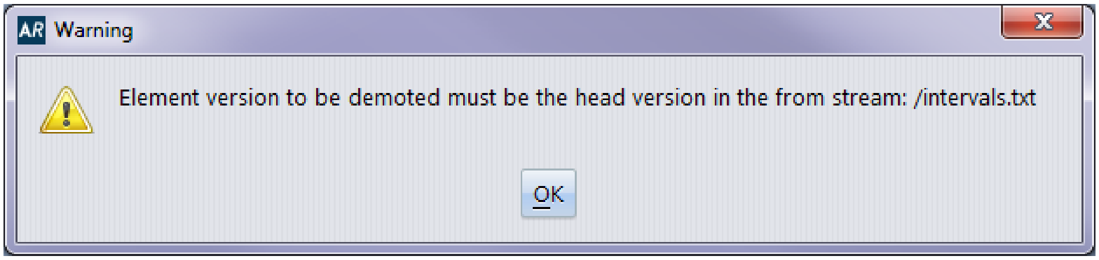


Time	Action	User	#	Comment
Aug 4, 2014 5:23:57 AM	promote	kjj	65	third set of changes; third promote
Aug 4, 2014 5:23:21 AM	promote	kjj	63	second set of changes; second promote
Aug 4, 2014 5:22:09 AM	promote	kjj	61	initial set of changes; first promote

Total items: 3

3. To demote a transaction, select the transaction (say, for example, #65) and click **Demote**.

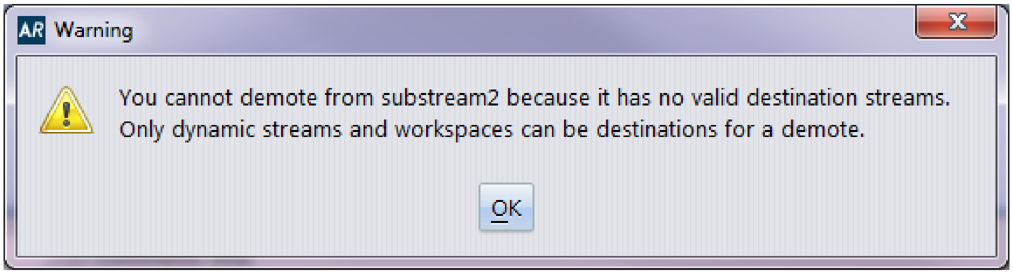
   Note: You can demote multiple transactions at once, but the transactions you demote must be consecutive for an element and they must contain the most recent transaction. If, for example, you attempted to demote just transaction #63, AccuRev displays the following warning:

   Element version to be demoted must be the head version in the from stream: /intervals.txt

4. If the current stream has more than one child stream or workspace, AccuRev displays a dialog box that lets you choose the destination for the demote. Select the destination stream or workspace and click **OK**.

   Note: If you attempt to demote from a stream that does not have a child stream or workspace, AccuRev displays the following warning:

   You cannot demote from substream2 because it has no valid destination streams.
   Only dynamic streams and workspaces can be destinations for a demote.

When the demote operation has succeeded, transaction #65 and the file versions it contains have been moved out of the source stream (in this case, the Dev stream) and into the destination stream.

Demote by File

When you demote a file, you move it out of the default group of active files in the source stream and you move all active versions of it to a child stream (the destination stream) where it is added to that stream's default group if there is no newer version of the file.

Note that after demoting a file, the source stream inherits the version of the file from its backing stream. Consequently, other versions of the file are no longer present in the source stream, although they exist in the destination stream.

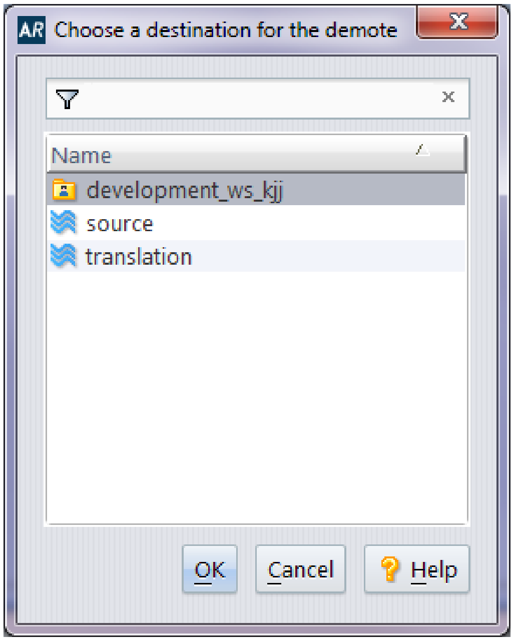
How to Demote by File

To demote a version by file:

1. In the StreamBrowser, specify Active Files as the Active Group Mode and select the elements you want to demote.
2. Click the **Demote** button ( ) on the StreamBrowser toolbar.

   Alternative: Right-click and choose **Demote** from the context menu.

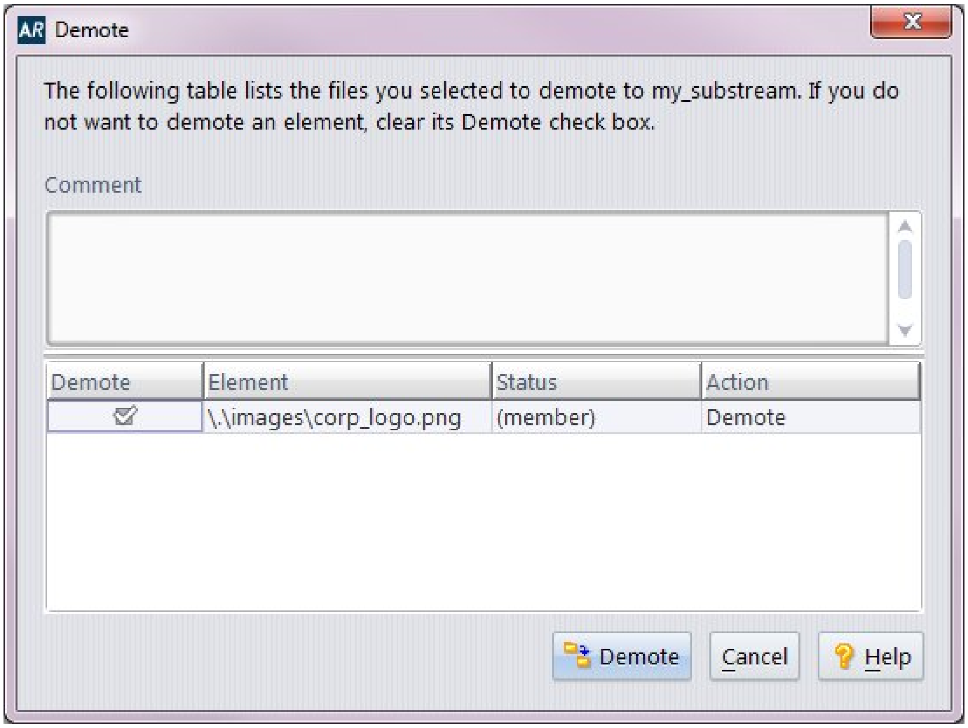
   If the current stream has more than one child stream or workspace, AccuRev displays the Choose a destination stream for the demote dialog box:

   Select the destination stream or workspace and click **OK**.

   Note: The dialog box displays only those workspaces belonging to the current user.

The Demote dialog box appears.



The following table lists the files you selected to demote to my_substream. If you do not want to demote an element, clear its Demote check box.

Comment

Demote	Element	Status	Action
☑	\.\images\corp_logo.png	(member)	Demote

3. In the **Comment** field, optionally enter a reason for the demote operation.

   Tip: AccuRev retains the comment you entered from your most recent demote.

4. If there are any files listed that you do not wish to demote at this time, clear that file's **Demote** checkbox.

5. Click the **Demote** button.

   The version is no longer in the source stream but it is now active in the destination stream.

Demote by Issue

Just as you can promote issues, you can also demote issues using the StreamBrowser or the Command Line Interface. For more information about the process of demoting by issue, see *Demote by Issue* on page 351.

Including and Excluding Files

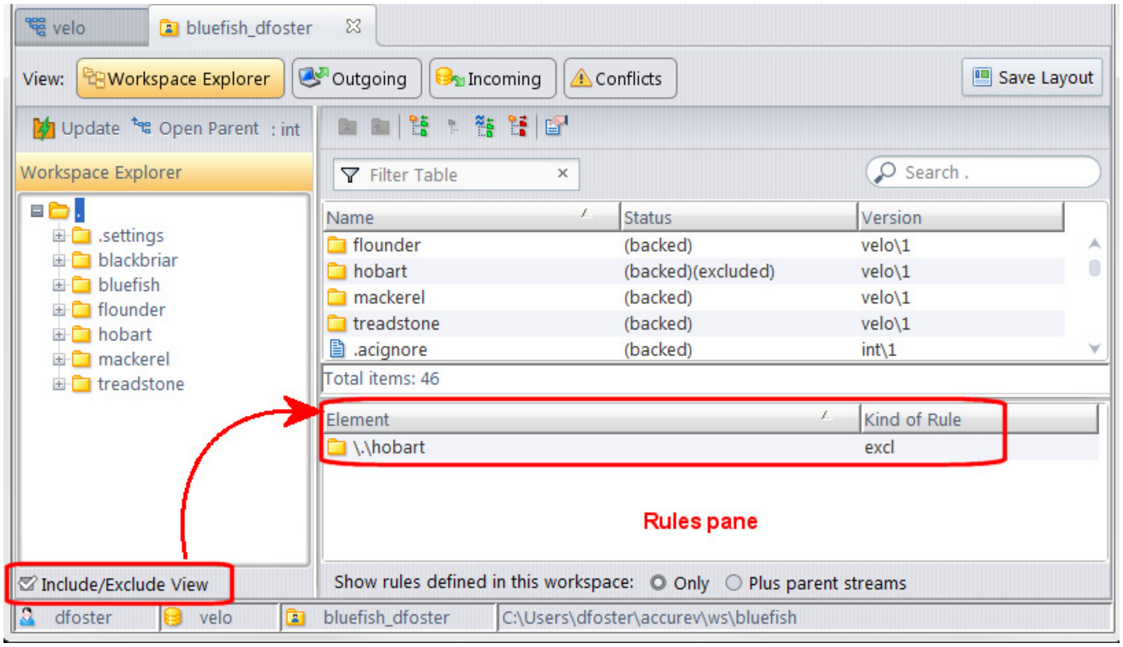
By default, workspaces and streams contain a copy of each file in the parent stream when they are created. A stream might contain hundreds or thousands of files, however, of which you might need or want only a subset. If you are a developer working with, say, core libraries, you might not want source files used by your team's GUI developers and technical writers in your workspace, for example.

This section describes how to use the Explorer mode's **Include/Exclude View** to choose the subset of directories and files you want to make available in a given workspace.

Displaying the Include/Exclude View

To display the Include/Exclude View, select the **Include/Exclude View** check box at the bottom of the Explorer pane.

When the Include/Exclude View is active, AccuRev adds another pane to the Details pane. The *Rules* pane displays any elements for which an include/exclude rule has been defined, as shown in the following illustration. If you are viewing a workspace or stream for which no rules have been defined, the Rules pane is empty.



Name	Status	Version
flounder	(backed)	velo\1
hobart	(backed)(excluded)	velo\1
mackerel	(backed)	velo\1
treadstone	(backed)	velo\1
.acignore	(backed)	int\1

Total items: 46

Element	Kind of Rule
\.\hobart	excl

Show rules defined in this workspace: ● Only ○ Plus parent streams

☑ Include/Exclude View

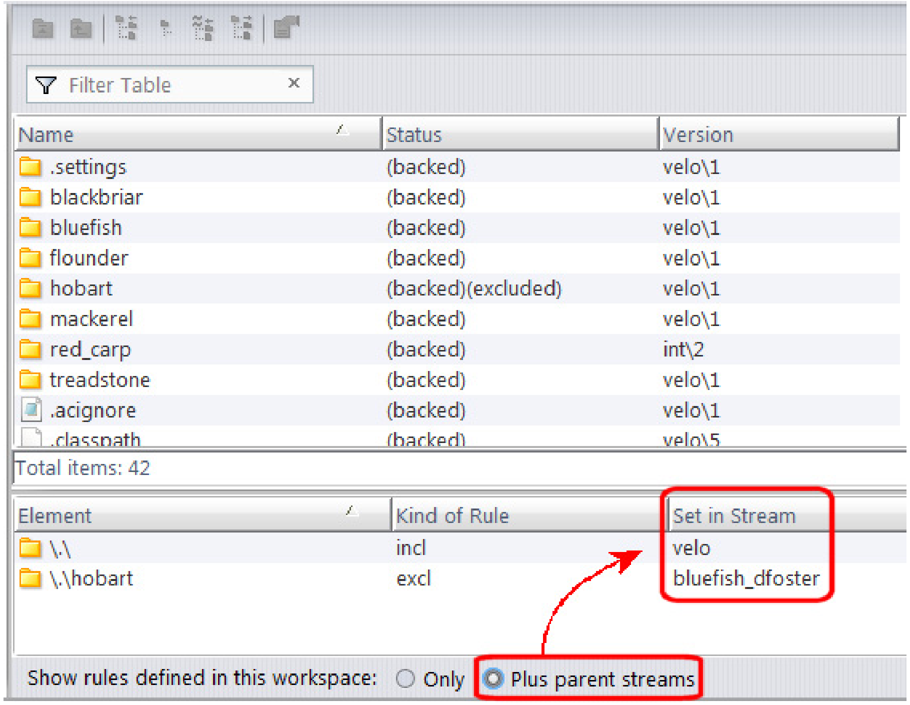
C:\Users\dfoster\accurev\ws\bluefish

Tip: The **Include/Exclude View** check box setting is saved as a user preference in AccuRev's **preferences.xml** file. The setting is applied to all workspaces and streams you subsequently open in the File Browser, though you can change it any time you wish.

By default, AccuRev displays all include/exclude rules defined directly in the current workspace or stream. If you want you can also display include/exclude rules:

- Inherited from streams higher up in the stream hierarchy, if any. This option is not applicable to the root stream.
- Defined in child streams, if any. This option is not applicable to workspaces.

When you select one of these display options, AccuRev adds a Set in Stream column to the table in the Rules pane to help you see where a given include/exclude rule was set, as shown in the following illustration.



Filter Table

Name	Status	Version
.settings	(backed)	velo\1
blackbriar	(backed)	velo\1
bluefish	(backed)	velo\1
flounder	(backed)	velo\1
hobart	(backed)(excluded)	velo\1
mackerel	(backed)	velo\1
red_carp	(backed)	int\2
treadstone	(backed)	velo\1
.acignore	(backed)	velo\1
.classpath	(backed)	velo\5

Total items: 42

Element	Kind of Rule	Set in Stream
\.\	incl	velo
\.\hobart	excl	bluefish_dfoster

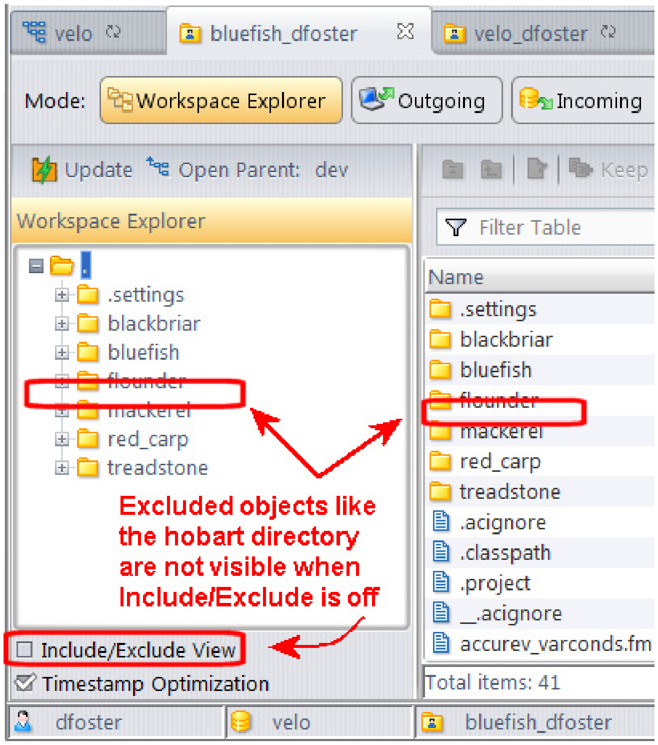
Show rules defined in this workspace: ○ Only ● Plus parent streams

See *Setting Include/Exclude Rules* on page 106 for more information.

How Include/Exclude View Affects the File Browser Display

In Include/Exclude View, both the Explorer and Details panes show *all* elements in the workspace or stream. For example, if the top-level directory *hobart* is excluded, it still appears in both the Explorer and Details panes in Include/Exclude View, as shown in the preceding illustration. You can navigate into an excluded directory as you would any other. Its elements will have a status of **(backed)(excluded)**.

Excluded directories and files are *not* displayed in the File Browser when Include/Exclude View is turned off, however -- notice that the directory *hobart* does not appear in the following illustration:



Excluded objects like the hobart directory are not visible when Include/Exclude is off

☐ Include/Exclude View

☑ Timestamp Optimization

Changes to Toolbar and Context Menu

In the Include/Exclude View, the Details pane toolbar and context menu change to support operations associated with this view -- setting and removing include/exclude rules. A few of the standard toolbar buttons remain, to aid you in navigating the depot and determining element history. Commands for typical AccuRev operations -- *Keep*, *Promote*, and *Merge*, for example -- are not applicable in this view, so they are not displayed.

Setting Include/Exclude Rules

This section describes include/exclude rules and how to add and remove them.

Types of Rules

AccuRev supports the following include/exclude rules:

Include

Make the selected elements appear in the workspace or stream. If you specify a directory element, the entire subtree below it will appear in the workspace or stream, too.

As usual, the particular versions that will appear in the workspace or stream are the ones inherited from the backing stream.

Include Directory Only

Make the selected *directory* elements, but not any of the elements *within* the directory, appear in the workspace or stream. You can set additional Include or Include Directory Only rules to make part of the directory's contents appear.

The Include Directory Only rule enables you to include files and directories that are deep in a depot's directory hierarchy, without having to include lots of other, unwanted elements. If you include a file or directory that is several levels below an include-only directory, AccuRev automatically creates the necessary Include Directory Only rules required to "dig down" to the level of the desired data.

Example:

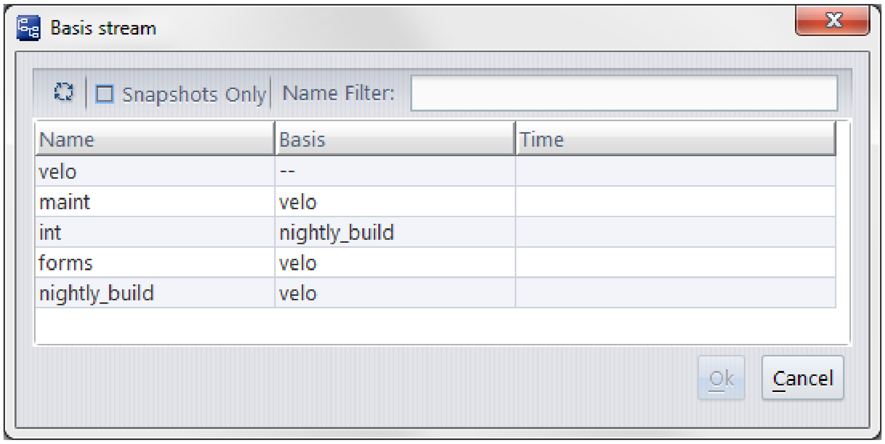
A depot contains a **tools** subdirectory, which contains a **src** subdirectory, which contain a **scripts** subdirectory, which contains two subdirectories, **perl** and **python**. If you need only the **python** subdirectory, not the rest of the tools subtree, use these commands:

- Set the Include Directory Only rule on **tools**
- Set the Include rule on **python**

In the Rules pane, note that AccuRev automatically creates "incldo" (Include Directory Only) rules for the intermediate directories, **src** and **scripts**.

Include from Stream

Establish a cross-link, making the selected elements appear in the workspace or stream. Instead of inheriting versions of the elements from the backing stream (as with the Include rule), inherit the versions from the stream you specify in the Basis Stream dialog.



Name	Basis	Time
velo	--	
maint	velo	
int	nightly_build	
forms	velo	
nightly_build	velo	

Use the toolbar to filter the streams shown in the dialog.

Note: For cross-linking to succeed, the element must have been present in the selected Basis Stream at the time your workspace was last updated. In a time-based stream, you cannot create a cross-link to another time-based stream (or to a snapshot) whose basis time is 'in the future' with respect to your stream.

- The **Refresh** button contacts the AccuRev Server to refresh the list of streams displayed in the Basis Stream dialog box.
- The **Snapshots Only** checkbox restricts the streams shown to snapshot streams only. This setting (as well as the sort order of the columns in the dialog) is stored as a preference.

- The **Name Filter** field filters the streams shown using the text you enter in the field. If the stream name contains the text entered in this field, the stream appears. Otherwise, it is filtered out and does not appear.
- If you specify a directory element, the entire subtree below it will appear in the workspace or stream, too.

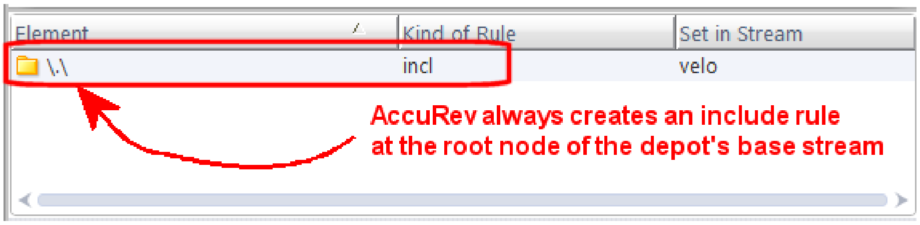
Note: In a stream, you cannot promote to cross-linked elements; in a workspace, you cannot keep (or anchor, or defunct, and so on) cross-linked elements.

Exclude

Remove the selected elements from the workspace or stream. For a workspace, the elements are deleted from the workspace tree. If you specify a directory element, the entire subtree below it will be removed.

Default Include Rule

AccuRev hard-codes a single include rule in each depot's base stream that specifies the depot's top-level (root) directory. This rule makes the depot's entire directory hierarchy visible in the depot's base stream.



Element	Kind of Rule	Set in Stream
\.\	incl	velo

AccuRev always creates an include rule at the root node of the depot's base stream

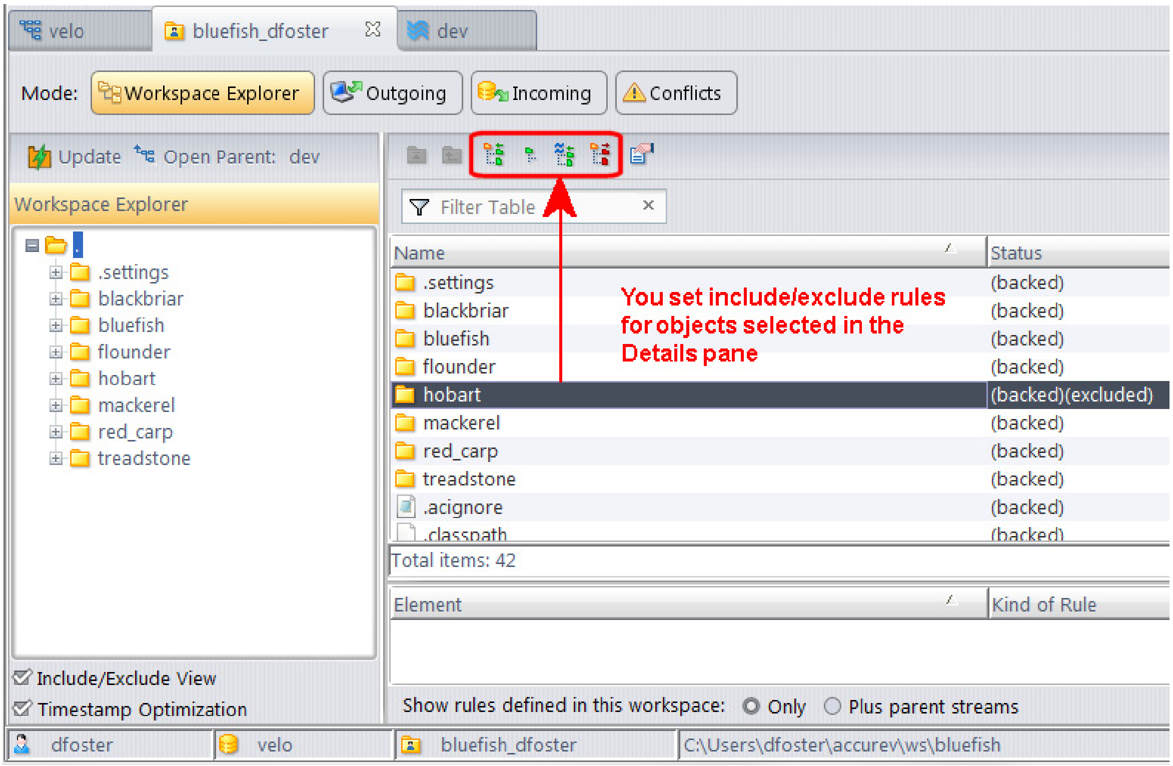
You cannot remove this rule.

Adding Rules

You can add any number of rules throughout the workspace or stream. Each rule that you add applies to a particular pathname within the depot's directory hierarchy (as shown in the "Element" column), and applies at a particular level in the depot's stream hierarchy (as shown in the "Set in Stream" column). A rule set in a dynamic stream gets inherited by lower-level streams; but a rule for the *same element* in a lower-level stream or workspace overrides a rule in a higher-level stream.

To add a rule:

1. In the workspace or stream Explorer, select the **Include/Exclude View** checkbox.
2. In the Explorer pane, select the *parent* directory to expose in the Details pane the directory or files for which you wish to specify an include/exclude rule. For example, if you want to set a rule for the *hobart* directory, select the root node in the Explorer pane to display all children of the root node in the Details pane, as shown in the following illustration.



You set include/exclude rules for objects selected in the Details pane

Name	Status
.settings	(backed)
blackbriar	(backed)
bluefish	(backed)
flounder	(backed)
hobart	(backed)(excluded)
mackerel	(backed)
red_carp	(backed)
treadstone	(backed)
.acignore	(backed)
.classpath	(backed)

Total items: 42

Element	Kind of Rule

Show rules defined in this workspace: ● Only ○ Plus parent streams

3. In the Details pane, select the directory or file for which you want to specify the rule. You can select multiple rows to set rules for more than one object at a time.

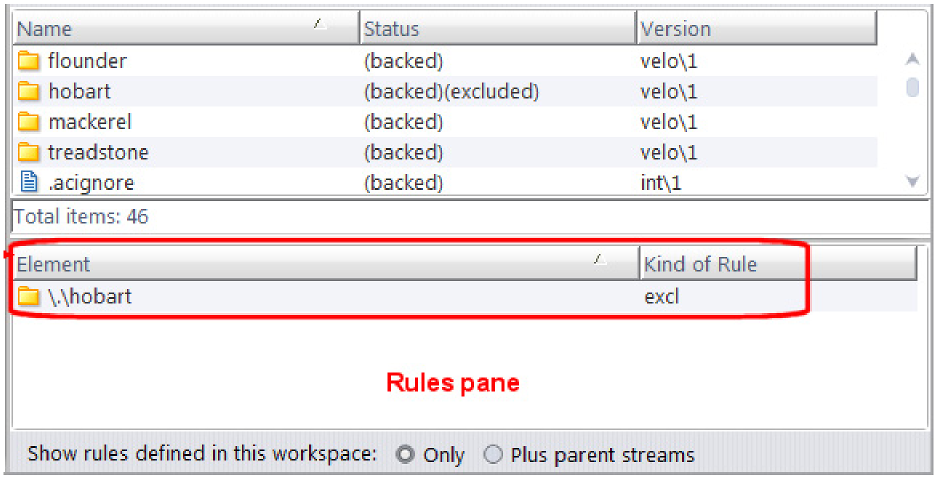
4. Select the rule you want to apply from the toolbar. See *Types of Rules* on page 106 for detailed information on the implication and effect of setting include/exclude rules.

   Alternative: Right-click the selected objects and choose the rule from the context menu.

   AccuRev displays a dialog box asking you to confirm setting the rule.

5. Click **Yes** to create the rule.

   The new rule appears in the rules pane:



Name	Status	Version
flounder	(backed)	velo\1
hobart	(backed)(excluded)	velo\1
mackerel	(backed)	velo\1
treadstone	(backed)	velo\1
.acignore	(backed)	int\1

Total items: 46

Element	Kind of Rule
\.\hobart	excl

Rules pane

Show rules defined in this workspace: ● Only ○ Plus parent streams

Note: If you set the rule in a dynamic stream, remember that elements are not removed from workspaces below that stream until they are updated. Element exclusion, however, is instantly inherited by *streams* below that stream.

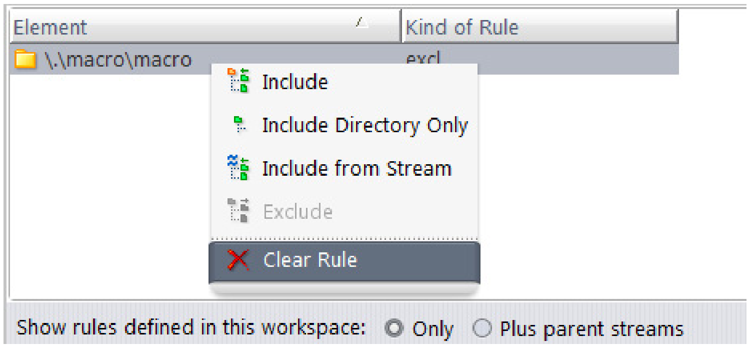
Removing Rules

When you remove a rule from a stream, the effect is immediate on the stream itself and on streams below it. The effect does not take place on workspaces below the stream until they are updated.

When you remove a rule from a workspace, the effect is immediate on the workspace itself: files are copied into the workspace tree if you remove an exclude rule; files are deleted from the workspace tree if you remove an include rule.

To remove an existing rule:

1. In the workspace or stream Explorer, select the **Include/Exclude View** checkbox.
2. Display the workspace or stream in which the rule was explicitly set (as shown in the *Set In Stream* column).
3. Right-click the rule, and select **Clear Rule** from the context menu.



Leaving Include/Exclude View

To leave Include/Exclude View in a workspace or stream, clear the checkbox at the bottom of the Explorer pane. The Rules pane disappears, and the Details pane reverts to displaying data for the current workspace or stream. Note that any elements that have been excluded are not displayed. See *How Include/Exclude View Affects the File Browser Display* on page 105 for more information.

Keep: Creating a New Version

Video Tip	Click *this link* to view an introductory video on checking in your work.

The ***Keep*** command creates a new version in the workspace stream for each selected file. Both the contents of the version and its identity are stored permanently in the AccuRev repository. For each file you ***Keep***:

- AccuRev copies the contents of the file currently in your workspace tree to the associated depot in the repository.
- AccuRev assigns a version-ID to the new version, consisting of the workspace's name and an integer. For example, the version-ID

  ```
  talon_dvt_mary/17
  ```

... identifies the 17th version of the file element that was created in the workspace named *talon_dvt_mary*.

Typically, you invoke the *Keep* command on files that you've been working on, and thus have **(modified)** status. But this is not a requirement. If you *Keep* a file that you have not modified, a new version is created with identical contents.

Invoking the Keep Command

In any workspace File Browser mode except Incoming Changes, select one or more file elements. Then:

- Click the ***Keep*** toolbar button, or
- Right-click the selection and choose ***Keep*** from the context menu.

The ***Keep*** dialog appears, which you use to complete the command.

The Keep Dialog

The ***Keep*** dialog includes basic options (always visible) and advanced options (visibility controlled by a Basic/Advanced button).

Basic Options

Comment

You can enter a comment string in the text box. (AccuRev "seeds" the text box with your most recent **Keep** comment made during the current AccuRev GUI session.) The comment string becomes a permanent annotation to the version you're creating.

AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a trigger on the ***Keep*** command.

Advanced Options

Specify File Type

By default, AccuRev determines the element type for the newly created version automatically (**text** or **binary**). You can override the default by specifying either of these types or **ptext** (a variant of **text**). See *AccuRev Element Types* on page 111.

Specify Lock Type

Choosing **Serial Development** places a lock on the element to ensure that users in sibling workspaces won't have to merge their work on this element. The lock remains on the element until you (or someone else) changes this value to **Parallel Development** in a subsequent ***Keep*** command. See *The Locks Dialog Box* on page 52.

AccuRev Element Types

Each version of a file element has an *element type*, which identifies the type of data contained in that version. Each time you create a new version (each time you keep your changes, for example), you can change the element type, using the **Advanced** section of the command's dialog box.

AccuRev supports the following element types. For a newly created version of any type, AccuRev creates a new storage file in the file storage area of the AccuRev repository.

binary

When a new binary version is created, the storage file is an exact copy of the file in the workspace tree. When you retrieve a binary version from the repository (for example, with ***Send to Workspace*** or ***Update***), AccuRev makes an exact copy of the storage file in the workspace tree.

text

When a new text version is created, the storage file is created with a single **NL** character (hex character code **x0A**) at the end of each text line. This means that a version's storage file may have different line terminators than the workspace-tree file you submitted to the command.

When you retrieve a text version from the repository (for example, with ***Send to Workspace*** or ***Update***), the file created in the workspace tree uses either the UNIX/Linux line terminator (**NL** / hex **x0A**) or the Windows line terminator (**CR-LF** / hex **x0D-x0A**) -- depending on the EOL setting for your workspace (see *The New Workspace Command: Screen 3 of 3—Configuring the Workspace)* on page 29).

ptext

ptext versions are handled like binary versions: exact copies are transferred between the workspace tree and the file storage area. The difference between ptext and binary files is in how they are handled by the ***Merge*** and ***Patch*** commands.

Results of a Keep Operation -- the Details

The ***Keep*** command does the following:

- **Creates a new version.** A new version object is created in the workspace stream. It has a unique version-ID, such as talon_dvt_mary/13 ("the 13th version of this element created in workspace talon_dvt_mary"). A version object is immutable, and cannot be removed from the repository. The version created by ***Keep*** is termed a real version, because it represents an actual change to an element.

  AccuRev allows you to invoke the Keep command on a file whose contents you haven't modified. So the new version might actually have the same contents as its predecessor.

- **Preserves the contents of the new version.** The contents of the new version are preserved permanently in the repository. AccuRev copies the file in your workspace tree to a container file in the depot. No AccuRev operation modifies or removes the container file (*exception*: archive command).
- **Sets the element status.** The file element gets the status flag **(kept)** in your workspace. It also gets the **(member)** flag, indicating that the element is active in your workspace. (It's possible that the element already had one or both these status flags, from previous activity in your workspace.) If the element previously had **(modified)** status, this flag is removed.

Notes:

Content changes and namespace changes. The ***Keep*** command preserves content changes only, not namespace changes. Suppose you edit a file, so that it has **(modified)** status, then give it a new name with the ***Rename*** command. The new version that ***Rename*** creates does not record your content changes. The file will still have **(modified)** status until you preserve the contents changes with ***Keep***.

"Undoing" a Keep operation. There is no true "undo" operation in AccuRev version control, since that would violate the TimeSafe principle. The Revert to Basis command provides a close approximation. If you want to "undo" some, but not all of several intermediate keeps, perform a ***Revert to Basis*** command followed by a Send to Workspace of the intermediate version.

Populate: Restoring Deleted Elements

For one or more selected elements, the ***Populate*** command copies the version that is currently in the workspace *stream* to the workspace *tree*. This has the effect of undoing an accidental deletion, or discarding edits that you have not yet preserved with ***Keep***.

Invoking the Populate Command

Select one or more file and/or directory elements in the Details pane of the File Browser. Then, choose ***Populate*** from the context menu of the selection.

Using the Populate Dialog

You can select one or both of these options:

Recursive

For each selected element that is a directory, perform a *Populate* on that element and on all elements below it.

Overwrite

For each file element, replace the file (if any) currently in the workspace tree. By default, *Populate* silently declines to overwrite an existing file.

The Populate Progress Box

The ***Populate*** command displays a progress box, much like the one used by the ***Update*** command.

Promote: Sharing Your Work

Video Tip	Click *this link* to view an introductory video on promoting your changes to the parent stream.

The ***Promote*** command sends one or more versions from one stream to another, making the versions accessible in the destination stream -- and in workspaces and streams below it. A single transaction in the AccuRev depot records the creation of the new version(s) in the destination stream.

The basic use of ***Promote*** is to take the private versions that you've created in your workspace stream and make them public by sending them to the workspace's backing stream.

Promote can also send versions from a dynamic stream to any other dynamic stream. You can do such higher-level promotion using the File Browser or StreamBrowser. Promoting from a stream to a non-parent ("cross-promotion") can be as simple as drag & dropping the version(s) to the destination stream. However, if there are any conflicts that must be resolved, AccuRev automatically displays the Change Palette.

Note: Promoting an element or symbolic link operates on the link itself, not on the target of the link.

Promoting New Work

Generally speaking, you use promote to share changes to elements that are already under AccuRev control. Note, however, that you can use the promote operation on external files -- when you promote a file with an

(external) status, AccuRev automatically performs the actions required to place the file under AccuRev control.

Similarly, you can directly promote elements you have modified -- AccuRev will automatically save your changes to the depot before promoting.

Real Versions and Virtual Versions

The ***Promote*** command creates a virtual version in a dynamic stream. No data is copied from your machine to the repository by this command; it just creates new entries in the repository database. By contrast, the **Keep** command (as well as **Anchor**, **Send to Workspace**, **Revert**, **Rename**, **Defunct**, and the CLI-only command **undefunct**) creates a real version in a workspace stream. See *Real Versions and Virtual Versions* on page 136.

Inheritance of Promoted Versions

After you promote a version, it can be inherited by all dynamic streams and workspaces below the backing stream. Inheritance by dynamic streams is automatic; inheritance by a workspace occurs when the user ***Update***'s it.

Overlap Status and Merging

If an element has **(overlap)** status, you cannot promote it. You must merge your version with the version in the backing stream, to create a new, merged version. You can then promote the merged version. See *Merging Your Changes with Someone Else's Changes* on page 87 for more information.

Underlap Status

The status **(underlap)** is similar to **(overlap)**, in that an element has changed in your workspace and also in the backing stream. With **(underlap)**, the changes in your workspace version have already been promoted to the backing stream (from another workspace, or from a stream elsewhere in the depot’s stream hierarchy). In many cases, it is most appropriate to use the ***Revert to Basis*** command to 'undo' the changes in your workspace. In other cases, you can merge and promote, as with **(overlap)**-status elements.

Invoking the Promote Command

To promote one or more files:

1. In the workspace File Browser, select one or more files from the Details pane.

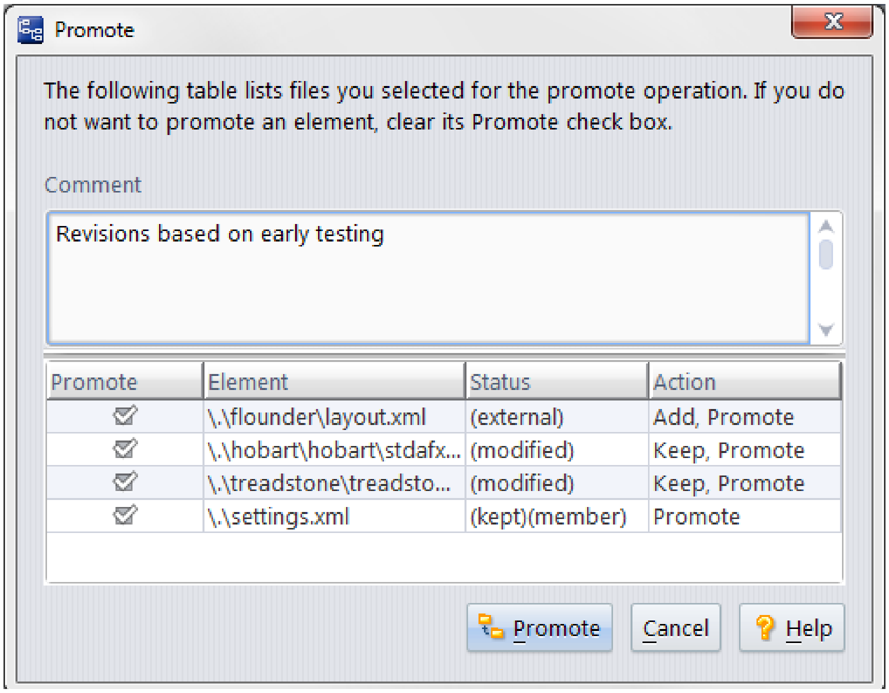
   Tip: Although you can promote files from the Explorer, Outgoing Changes, and Conflicts modes, consider using the Outgoing Changes mode to see all the new work and changes you might want to promote to the parent stream.

2. Click the button in the Details pane toolbar, or right-click and choose **Promote** from the context menu.

   What happens next depends on whether or not AccuRev is integrated with an issue-tracking system (ITS), such as AccuWork or a third-party system.

   If AccuRev is integrated with an ITS, the Select Issue (Change Package) dialog box appears. Choose the issues against which you want to promote your changes and click **Ok** to continue. See *Promoting Changes When AccuRev is Integrated with an ITS* on page 115 for more information.

If AccuRev is not integrated with an ITS, the Promote dialog appears.



The following table lists files you selected for the promote operation. If you do not want to promote an element, clear its Promote check box.

Comment

Revisions based on early testing

Promote	Element	Status	Action
☑	\.\flounder\layout.xml	(external)	Add, Promote
☑	\.\hobart\hobart\stdafx...	(modified)	Keep, Promote
☑	\.\treadstone\treadsto...	(modified)	Keep, Promote
☑	\.\settings.xml	(kept)(member)	Promote

3. Optionally, enter a comment describing the changes you are promoting. (AccuRev "seeds" the text box with the most recent **Promote** comment made during your current AccuRev GUI session.) The comment string becomes a permanent annotation to the transaction that creates the new version(s) in the destination stream.

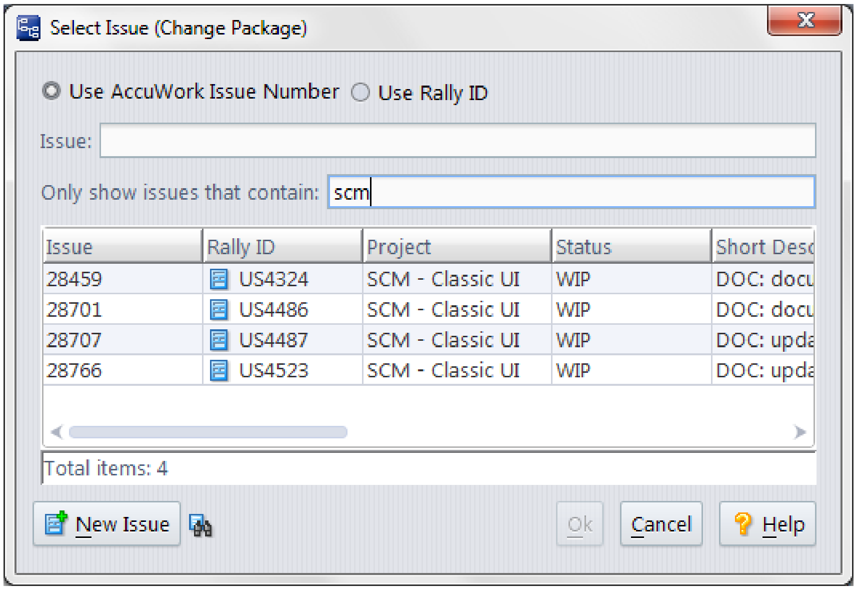
   Tip: AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a trigger on the ***Promote*** command.

4. Optionally, clear the **Promote** checkbox for any versions that you do not want to promote.
5. Click the **Promote** button.

   The elements' status changes to **(backed)**. If you performed the promote operation from in Outgoing Changes mode, the element is removed from the Details pane.

Promoting Changes When AccuRev is Integrated with an ITS

If you are using AccuWork, or if you have integrated AccuRev with a third-party issue tracking system (ITS), when you promote AccuRev executes the default AccuWork query and displays its results in the Select Issue (Change Package) dialog box. You use this dialog box to select one or more issues with which you want to associate the changes you are promoting:


Select Issue (Change Package)

● Use AccuWork Issue Number ○ Use Rally ID

Issue:

Only show issues that contain: scm

Issue	Rally ID	Project	Status	Short Desc
28459	US4324	SCM - Classic UI	WIP	DOC: docu
28701	US4486	SCM - Classic UI	WIP	DOC: docu
28707	US4487	SCM - Classic UI	WIP	DOC: upda
28766	US4523	SCM - Classic UI	WIP	DOC: upda

Total items: 4

You can select one or more of the issue records and click **Ok**; or you can type one or more numbers (*SPACE*-separated) in the **Issue** input field. When you promote, the integration(s) record information about the *Promote* transaction in the issue record(s) you designate.

Note: If you have enabled the use of third-party ITS keys in the AccuWork schema, the Select Issue (Change Package) dialog displays additional controls that let you specify whether you want to use AccuWork issue numbers or third-party keys when selecting issues to promote against. See *Using Third-Party Keys in the AccuRev GUI* on page 408 for more information.

Promoting Dependent Elements

A *dependent element* is an element that has a relationship with another element. Dependent elements must be promoted as a group in order to resolve the dependency. This section describes how dependencies can be created and how to promote them.

How Dependencies Can Be Created

A dependency can be created when you rename an element in the workspace using the name of another element that has also been renamed and promoted to the backing stream. For example, imagine that you **add** the following files to your workspace, and then **keep** them and **promote** them to the backing stream:

```
proposed_tracker.c
beta_tracker.c
ref_tracker.c
```

Next you rename these elements in the workspace as follows:

`proposed_tracker.c` is renamed as `accepted_tracker.c`
`beta_tracker.c` is renamed as `proposed_tracker.c`
`ref_tracker.c` is renamed as `beta_tracker.c`

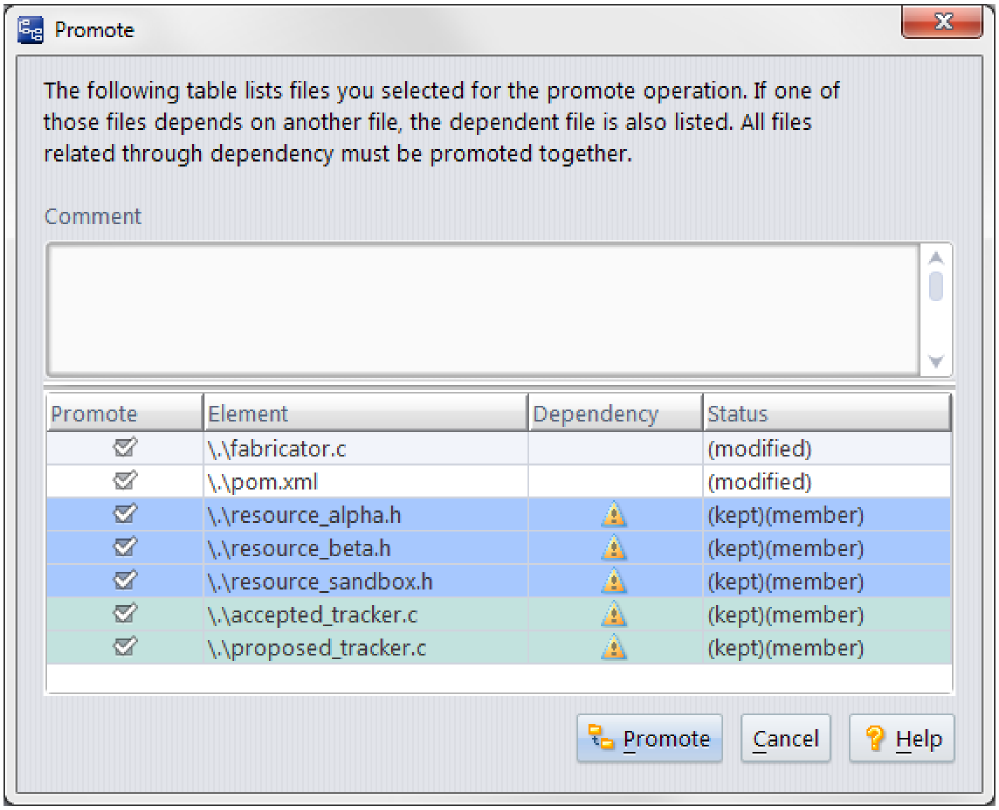
The element `accepted_tracker.c` can be promoted freely -- it has simply been renamed using a new name (that is, the name of an element that has not already been promoted to the backing stream). Notice, however, that elements `beta_tracker.c` and `ref_tracker.c` were renamed with names of elements that have already been promoted to the backing stream and now have dependencies (`proposed_tracker.c` and `proposed_tracker.c`, respectively), creating a dependency -- `proposed_tracker.c` cannot be promoted from the workspace without `accepted_tracker.c`. Similarly, `beta_tracker.c` requires that both `proposed_tracker.c` and `accepted_tracker.c` be promoted with it.

Element dependencies can also be created during the twin resolution process.

How to Promote Dependent Elements

When AccuRev detects a dependency during a **promote** operation, it displays a message alerting you to the dependency and the requirement to promote dependent elements together, as a group.

When you dismiss the message, the Promote dialog box appears. Dependent elements are included in the table, along with any other files you have selected for **promote**, as shown in the following illustration:



The following table lists files you selected for the promote operation. If one of those files depends on another file, the dependent file is also listed. All files related through dependency must be promoted together.

Comment

Promote	Element	Dependency	Status
☑	\.\fabricator.c		(modified)
☑	\.\pom.xml		(modified)
☑	\.\resource_alpha.h	⚠	(kept)(member)
☑	\.\resource_beta.h	⚠	(kept)(member)
☑	\.\resource_sandbox.h	⚠	(kept)(member)
☑	\.\accepted_tracker.c	⚠	(kept)(member)
☑	\.\proposed_tracker.c	⚠	(kept)(member)

Dependent elements are displayed with a warning icon in the **Dependency** column. Elements that share a dependency are highlighted using the same color. Because dependent elements must be promoted together, clearing the **Promote** check box for one element in a group clears the check box for *all* elements in that group.

Renaming an Element

The ***Rename*** command changes the pathname at which an element is accessed. You can use this command to perform a simple renaming, to move an element to another directory, or both. You cannot move an element to another depot.

Invoking the Rename Command

Select an element in the Details pane of the File Browser. Then choose ***Rename*** from the selection's context menu.

In the Rename dialog box, type a new pathname for the element:

- Type a simple name to rename the element, leaving it in the same directory.
- Type a relative pathname to move the element to another directory in the depot (and possibly rename the element, too).

(You cannot type a depot-relative pathname.)

In either case, a single **move** transaction records the changing of the element's pathname in your workspace. The renamed object is activated (it is included in the workspace's default group), and the **(kept)** and **(member)** flags are added to its status.

Renaming a Modified File Before Keeping It

You can create a new version of a file element in your workspace, recording either a content change or a namespace change. But you can't do both at once. For example, suppose you:

1. Edit the contents of file *color.java*, so that it gets **(modified)** status.
2. Use the Rename command to change the filename to *hue.java*.

This creates a new version of the file element in your workspace, recording the name change, but not the content change. To preserve the content change, you must use a separate ***Keep*** command to create a second new version.

Caution on Reusing the Name of a Renamed Element

The ability to reuse the name of a renamed element provides significant flexibility for project refactoring tasks. But it also introduces a complication: what happens if you rename an element, create a new element at the same pathname, then invoke the ***Revert to Basis*** command on the renamed element?

The renamed element cannot revert to its old pathname, because there’s a new element at that pathname. The original element simply disappears from your workspace. You might wonder “Why does the element not get **(stranded)** status?” The ***Revert to Basis*** command makes an element inactive in the workspace. The **(stranded)** status applies only to active elements.

At this point, your workspace contains a new element at the given pathname, and the parent stream contains the original element at that pathname. Attempting to promote the new element would produce a 'name already exists in parent stream' error. Use one of the following procedures to return this pathname to a consistent state:

If you want to return to using the original element

1. Rename the new element, to a name like *myfile.java.DISCARDED*.
2. Defunct the new element.
3. Promote the new element to the backing stream.
4. At this point, the original element reappears in the Details pane, with **(missing)** status.
5. Invoke the Populate command on the original element.

If you want to discard the original element and use the new element

These steps must be performed with the AccuRev CLI, which supports defuncting of the "disappeared" element using its *element-ID*.

1. Defunct the original element in the workspace, using the command **accurev defunct –e** ***<element-ID>***.
2. Promote the defuncted version to the backing stream.

Resolving (twin) Status

An element has a (twin) status when it and another element have the same pathname in a dynamic stream. Elements with (twin) status are highlighted in yellow in the File Browser, Change Palette, and the default group to make them easy to spot. Both directories and files can have (twin) status.

Elements with (twin) status cannot be promoted until their (twin) status is resolved. To resolve an element's (twin) status, you can:

- Defunct one of the elements in conflict. For example, you might decide to defunct the version of the element that is in the workspace, implicitly keeping the one that is in the backing stream.
- Rename one of the elements in conflict. This action lets you keep both versions in AccuRev.

You can perform defunct and rename operations manually using commands in either the AccuRev GUI or the AccuRev CLI, but the AccuRev GUI provides enhanced features -- a twin status resolution wizard, and one-step defunct operations -- that help simplify the process of resolving an element's (twin) status. The remainder of this section describes these features and when to use them.

Locating Twin Elements

Elements with a (twin) status are displayed in all File Browser modes, but consider using Conflicts mode to locate them quickly and easily. See *Conflicts Mode* on page 73 for more information.

The Twin Resolution Wizard

AccuRev provides a wizard you can use to resolve an element's (twin) status. You should consider using the twin resolution wizard if:

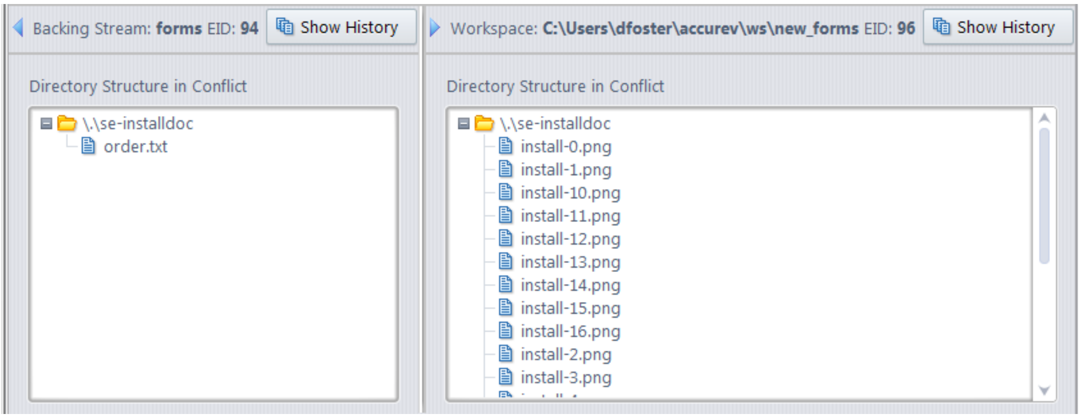
- **You want to compare the elements that are in conflict**. Before deciding whether to defunct or rename an element that is in conflict, it can be useful to compare the versions side-by-side. The twin resolution wizard provides a Diff view for directories and files. The information that is displayed varies by type -- colors are used to show changed, added, and deleted text in text files, for example. See *Diff View Varies Based on Element Type* for more information. AccuRev also displays the element ID (EID) of the versions in conflict.
- **You want to view element history**. A **Show History** button for both versions of the element that is in conflict opens the History Browser, allowing you to review the transactions and changes associated with each version.
- **You want to merge the elements' contents**. The twin resolution wizard's Merge Contents page includes the same tools as AccuRev's Merge view, allowing you to review and accept or reject proposed changes. When you choose the element in which you want to save the merge changes, AccuRev automatically defuncts the other version. Merge is available for text files only.

Note that you can use the twin resolution wizard on individual elements only. You can, however, defunct multiple elements at one time. See *Defuncting Twin Elements* for more information.

Diff View Varies Based on Element Type

The following list summarizes how the twin resolution wizard's Diff view varies for directories and different types of files.

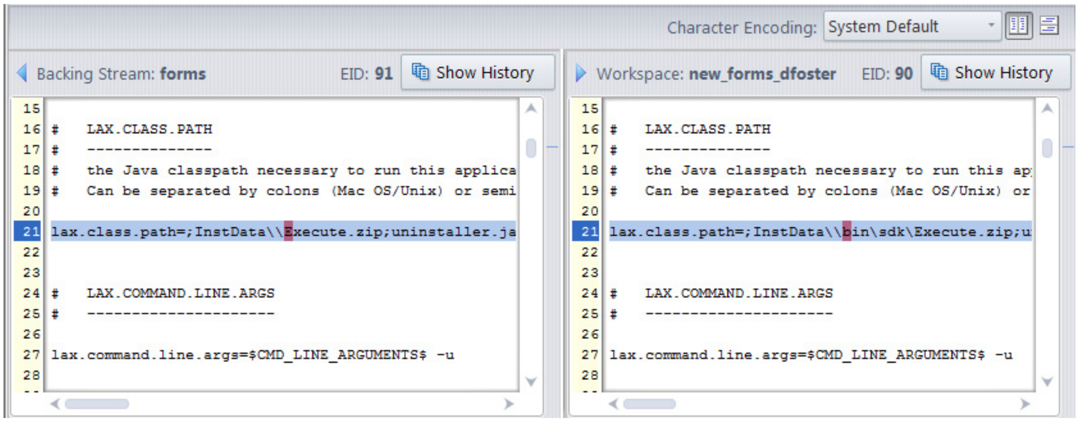
- **Directories.** When you resolve a directory with a (twin) status, the twin resolution wizard displays navigable trees for the directories that are in conflict, allowing you to easily compare directory contents.


Backing Stream: forms EID: 94
Show History
Workspace: C:\Users\dfoster\accurev\ws\new_forms EID: 96
Show History
Directory Structure in Conflict
Directory Structure in Conflict

```
\.\se-installdoc
    order.txt
```

```
\.\se-installdoc
    install-0.png
    install-1.png
    install-10.png
    install-11.png
    install-12.png
    install-13.png
    install-14.png
    install-15.png
    install-16.png
    install-2.png
    install-3.png
```

- **Text files.** When you resolve a text file with a (twin) status, the twin resolution wizard displays a Diff view with color-coded text to show you where the files in conflict differ.

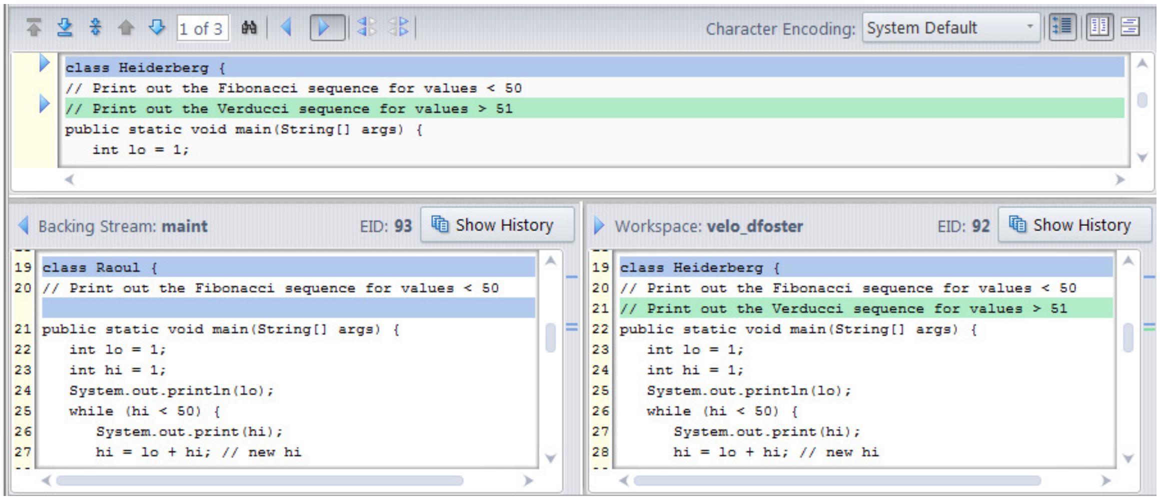

Character Encoding: System Default
Backing Stream: forms
EID: 91
Show History
Workspace: new_forms_dfoster
EID: 90
Show History

```

#   LAX.CLASS.PATH
#   --------------
#   the Java classpath necessary to run this applica
#   Can be separated by colons (Mac OS/Unix) or semi

lax.class.path=;InstData\\Execute.zip;uninstaller.ja


#   LAX.COMMAND.LINE.ARGS
#   ---------------------

lax.command.line.args=$CMD_LINE_ARGUMENTS$ -u

```

```

#   LAX.CLASS.PATH
#   --------------
#   the Java classpath necessary to run this ap
#   Can be separated by colons (Mac OS/Unix) or

lax.class.path=;InstData\\bin\sdk\Execute.zip;u


#   LAX.COMMAND.LINE.ARGS
#   ---------------------

lax.command.line.args=$CMD_LINE_ARGUMENTS$ -u

```

For details about using the Diff view, see *The Diff Tool* on page 203.

With text files, you can optionally choose to merge contents; in this case, the twin resolution wizard displays a full-featured Merge view, as shown here:



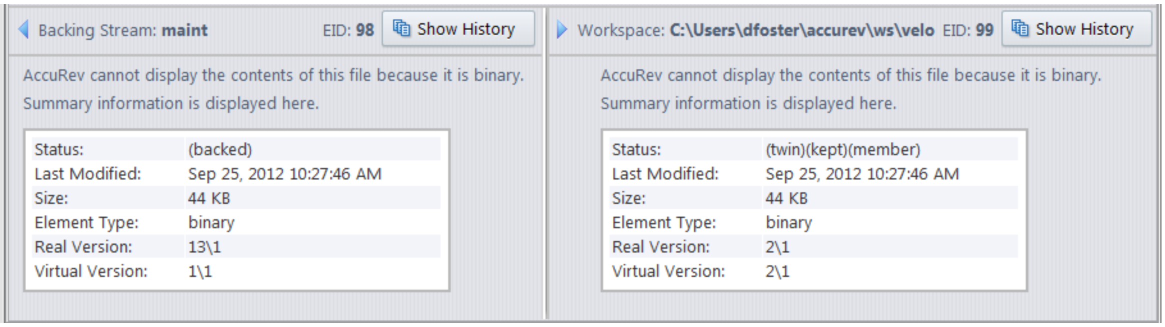
```
class Heiderberg {
// Print out the Fibonacci sequence for values < 50
// Print out the Verducci sequence for values > 51
public static void main(String[] args) {
   int lo = 1;
```

```
class Raoul {
// Print out the Fibonacci sequence for values < 50

public static void main(String[] args) {
   int lo = 1;
   int hi = 1;
   System.out.println(lo);
   while (hi < 50) {
      System.out.print(hi);
      hi = lo + hi; // new hi
```

```
class Heiderberg {
// Print out the Fibonacci sequence for values < 50
// Print out the Verducci sequence for values > 51
public static void main(String[] args) {
   int lo = 1;
   int hi = 1;
   System.out.println(lo);
   while (hi < 50) {
      System.out.print(hi);
      hi = lo + hi; // new hi
```

For details about using the Merge view, see *The AccuRev Merge Tool* on page 216.

- **Binary files.** If the file type of the element with (twin) status is binary, the twin resolution wizard displays a table that summarizes information about each version (the date the file was last modified, real and virtual version numbers, and so on).



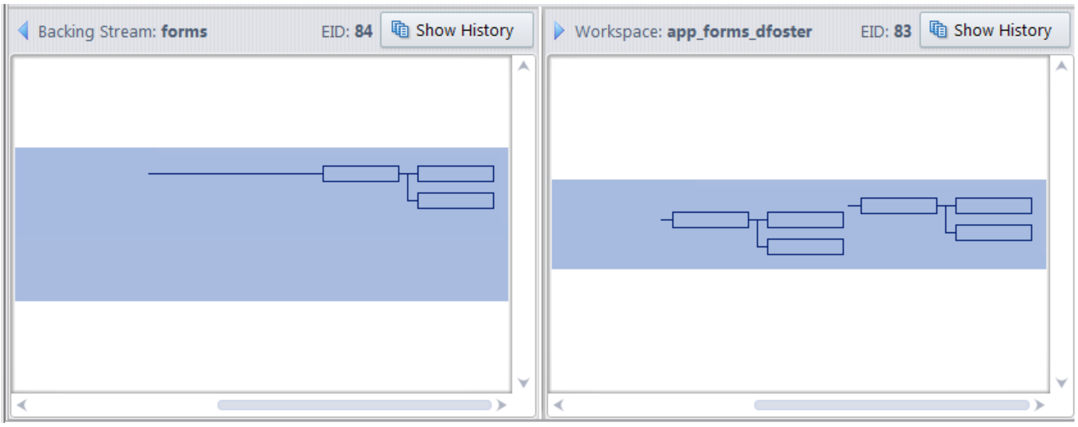
AccuRev cannot display the contents of this file because it is binary. Summary information is displayed here.

Status:	(backed)
Last Modified:	Sep 25, 2012 10:27:46 AM
Size:	44 KB
Element Type:	binary
Real Version:	13\1
Virtual Version:	1\1

AccuRev cannot display the contents of this file because it is binary. Summary information is displayed here.

Status:	(twin)(kept)(member)
Last Modified:	Sep 25, 2012 10:27:46 AM
Size:	44 KB
Element Type:	binary
Real Version:	2\1
Virtual Version:	2\1

- **Image files.** To allow simple visual inspection of versions that are in conflict, the twin resolution wizard renders images for .jpeg, .jpg, .gif, and .png file types, as shown in the following illustration.



Images of other types are treated as plain binary files.

Where You Can Run the Wizard

You can run the twin resolution wizard from the:

- File Browser
- Default group
- Change Palette

The basic procedure for running the wizard is the same regardless of where you start. However, there are slight differences to consider when running the wizard from the Change Palette. These differences are identified in the following procedure, *How to Run the Wizard*.

How to Run the Wizard

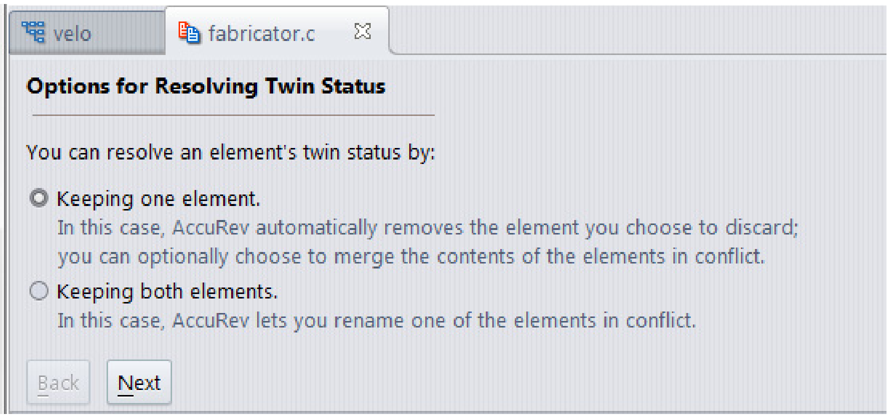
To run the twin resolution wizard:

1. Right-click the element whose (twin) status you want to resolve.

   Note: The twin resolution wizard can be used on individual elements only.

2. Choose **Resolve Twin Status** > **Use Wizard** from the element's shortcut menu.

   AccuRev displays the first page of the twin resolution wizard. Instructions and actions appear at the top of the view, as shown here:



Options for Resolving Twin Status

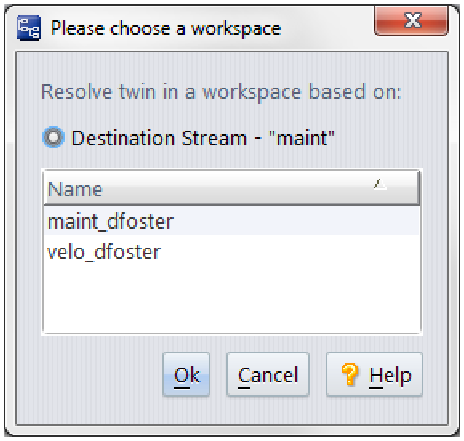
You can resolve an element's twin status by:

● Keeping one element.
In this case, AccuRev automatically removes the element you choose to discard; you can optionally choose to merge the contents of the elements in conflict.

○ Keeping both elements.
In this case, AccuRev lets you rename one of the elements in conflict.

   Versions of the element in conflict appear in a panel beneath the instructions as described in the section *Diff View Varies Based on Element Type*.

   Note for Change Palette users: If you are cross-promoting from one stream to another (not from a stream to a workspace), AccuRev first displays the Please Choose a Workspace dialog box. The Please Choose a Workspace dialog box displays a list of workspaces off the destination stream you selected when creating the Change Palette. AccuRev needs this workspace in order to resolve the twin elements. For example:



Please choose a workspace

Resolve twin in a workspace based on:

● Destination Stream - "maint"

Name
maint_dfoster
velo_dfoster

Choose the destination stream workspace you want to use and click **Ok** to continue.

If the destination stream workspace you select in the Please Choose a Workspace dialog box does not contain a version of the twin element with a (backed) status, AccuRev displays a warning message. AccuRev will not start the twin resolution wizard until you have addressed the situation described in the warning message.

3. In the Diff view, review the versions of the element that are in conflict and then specify whether you want to keep one or both elements:

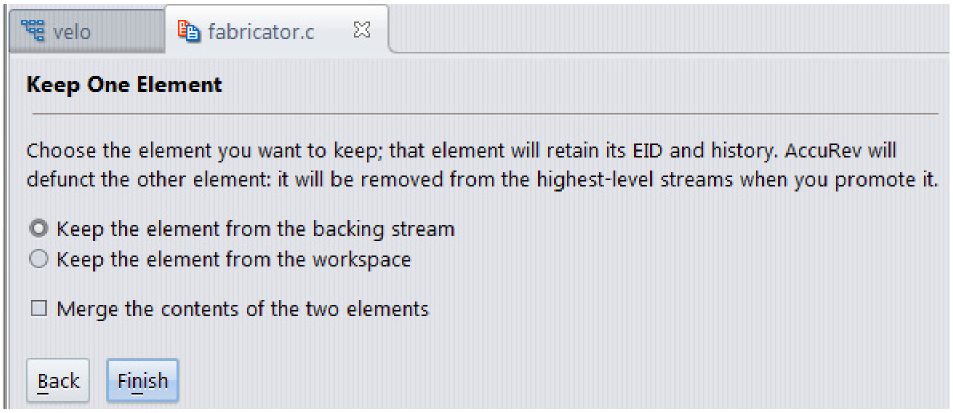
   - **Keep one element.** Choose this option if you know which element you want to keep; AccuRev will automatically defunct the other. Then, click **Next** and go to *Step 4*.

     Note: If you want to merge the contents of a text file that is in conflict, choose this option.

   - **Keep both elements.** Choose this option to keep both elements. Then, click **Next** and go to *Step 6*.

   Note that if you click the **Show History** button for an element, you must return to the twin resolution wizard to complete the resolution process. The wizard and the History Browser views are independent.

4. On the Keep One Element page, select the version of the element you want to keep.



Keep One Element

Choose the element you want to keep; that element will retain its EID and history. AccuRev will defunct the other element: it will be removed from the highest-level streams when you promote it.

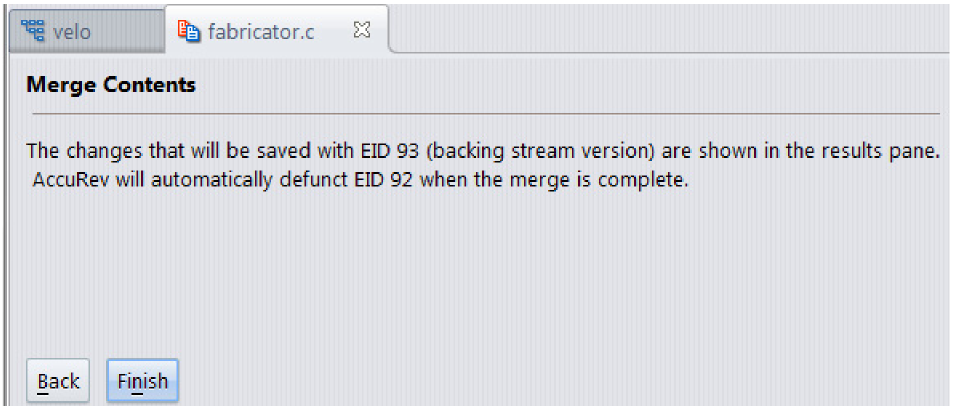
● Keep the element from the backing stream
○ Keep the element from the workspace

□ Merge the contents of the two elements

   If you are resolving the (twin) status of a directory, binary file, or image file, or of a text file whose contents you do not wish to merge, go to *Step 7*.

   If you are resolving the (twin) status of a text file and you want to merge file contents, select the **Merge the contents of the two elements** check box. Then, click **Next** and go to *Step 5*.

5. In the Merge view on the Merge Contents page, review the results panel, which displays the changes that will be saved to the file you chose in *Step 4*. You can use the tools of the Merge view to keep or reject individual changes. See *The AccuRev Merge Tool* on page 216 for more information.

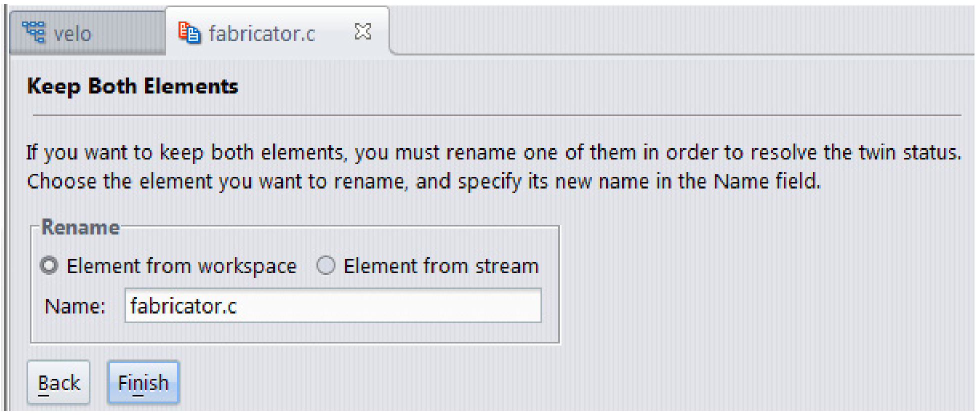


Merge Contents

The changes that will be saved with EID 93 (backing stream version) are shown in the results pane. AccuRev will automatically defunct EID 92 when the merge is complete.

When you are satisfied with the proposed merge results, go to *Step 7*.

6. On the Keep Both Elements page, select the version of the element you want to rename, and then enter the new name in the **Name** field.



Keep Both Elements

If you want to keep both elements, you must rename one of them in order to resolve the twin status. Choose the element you want to rename, and specify its new name in the Name field.

Rename

● Element from workspace ○ Element from stream

Name: fabricator.c

Note that you can change only the element's name, and not its path; the element will remain in the same directory and same stream (workspace or backing, for example) that you choose in this step.

7. Click **Finish**.

AccuRev resolves the element's (twin) status as you specified in the wizard.

Note for Change Palette users: If you are cross-promoting changes from one stream to another (not from a stream to a workspace), AccuRev displays the Promote dialog box, identifying elements in the destination stream workspace that need to be promoted in order to complete the twin resolution process. Names for these elements are appended with the suffix **.old_*<EID>***, where *<EID>* is the element ID (EID) of the version you chose to defunct in *Step 4*. For example, if resolving the twin element resulted in defuncting an element with the name **fabricator.c** and an EID of 92, the element name displayed in the Promote dialog box would be **fabricator.c.old_92**.

Defuncting Twin Elements

AccuRev lets you explicitly defunct one or more elements with a (twin) status directly from the shortcut menu. That process is described here.

When to Use Defunct

Consider using defunct to resolve an element's (twin) status when:

- You know which of the elements in conflict you want to defunct (the version in the workspace, and not the version in the backing stream, for example)
- You do not need to compare or merge the elements' contents before defuncting
- You want to resolve multiple elements with a (twin) status at the same time

How to Defunct Twin Elements

To defunct one or more twin elements:

1. Select the elements whose (twin) status you want to resolve.

   Tip: Use SHIFT+ and CTRL+ to select multiple elements.

2. Right-click one of the selected elements and choose **Resolve Twin Status** from the shortcut menu.

3. Next, choose the version you want to defunct (workspace or backing stream, for example).

   Note: The same version is defuncted for *all* of the elements you selected in *Step 1*.

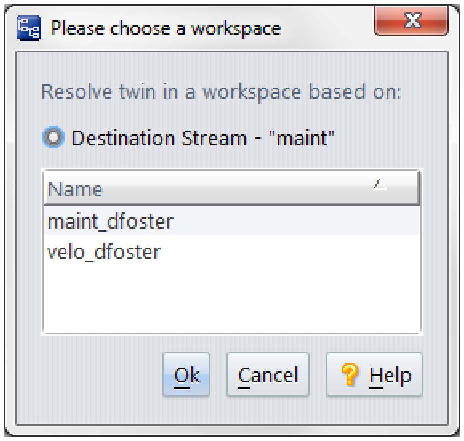
   AccuRev displays a message asking you to confirm the defunct operation.

4. Choose **Yes** to defunct the version of the selected elements.

   Choose **No** to cancel the defunct operation.

   If you are defuncting the twin element from the File Browser, you are done.

   Note for Change Palette users: If you are defuncting the twin element from the Change Palette, AccuRev displays the Please Choose a Workspace dialog box. The Please Choose a Workspace dialog box displays a list of workspaces off the destination stream you selected when creating the Change Palette. AccuRev needs this workspace in order to resolve the twin elements. For example:



Please choose a workspace

Resolve twin in a workspace based on:

● Destination Stream - "maint"

Name
maint_dfoster
velo_dfoster

   Choose the destination stream workspace you want to use and click **Ok** to continue.

   If the destination stream workspace you select in the Please Choose a Workspace dialog box does not contain a version of the twin element with a (backed) status, AccuRev displays a warning message.

AccuRev will not start the twin resolution wizard until you have addressed the situation described in the warning message.

Revert: Undoing Changes

The ***Revert*** command implements an "undo change" capability for dynamic streams. It doesn't actually remove any versions from the stream or transactions from the repository; that never occurs, since it would be a violation of AccuRev's TimeSafe principle. Instead, ***Revert*** creates a new version of one or more elements in a workspace based on the stream.

> **Note:** There is one exception to this workspace requirement: the ***Revert Change Package*** command option to continue without using a workspace, available in the StreamBrowser. (See *“Revert Change Package”* on page 361.)

The contents of the new version are the result of "subtracting out" a certain set of changes from the stream's current version:

- the changes that were added to the stream in a specified promote transaction, or
- the changes contained in a change package that is currently in the stream.

The "subtracting out" of content changes is performed by the Merge tool. Submitting a different set of versions to this tool effectively implements the ***reverse patch algorithm*** (see *Reverse Patch: Removing Content Changes* on page 238).

After the ***Revert*** command completes, you can verify the correctness of its work (and make further changes, if appropriate) in the workspace. Then, you can complete the "undo change to stream" process by promoting the new versions.

> **Note:** Again, the ***Revert Change Package*** command available in the StreamBrowser (*Revert Change Package*) has an option to continue without using a workspace. The results of this revert are placed directly in the stream itself. However, AccuRev recommends that you still build and test these changes in an updated workspace to verify the results.

Invoking the Revert Command

There are several ways to invoke the ***Revert*** command:

- **Reverting a Transaction**. Open a ***History Browser*** tab, showing the transactions for the dynamic stream where you want to perform the "undo". To do this, choose **View > Streams** from the GUI main menu (and, if prompted, choose the depot containing your stream). Right click your stream, and choose **Show History** from the context menu. Then right-click the transaction to be reverted, and choose ***Revert*** from the context menu. Only transactions whose action is **promote** can be reverted.
- **Reverting a Change Package**. Open a ***Stream Issues*** tab, showing the issues whose change packages are in the dynamic stream where you want to perform the "undo". To do this, choose **Show Issues** from the context menu of any dynamic stream in the StreamBrowser. Or choose **Show Diff by Issues** from the context menu of a dynamic stream, then select another stream to compare it with.

Choosing a Workspace for the Revert Command

AccuRev displays the workspaces that both:

- Belong to you, and
- Are children of the stream whose transaction or change package you wish to revert

The workspace you select will be used to perform a reverse patch operation. AccuRev invokes the Merge tool in a special mode to perform this operation, with the file in the selected workspace as one of the contributors. It refuses to proceed if the file is currently active in the workspace.

Revert to Basis

For each selected element, this command removes all changes you've made since the last time you *Promoted* it, or since the last *Update* of the workspace -- whichever is more recent:

- File elements: this includes both content changes and namespace changes.
- Directory elements: this includes namespace changes only.
- Link elements: this command operates on the link itself, "undo"ing changes you've made to the specification of the link target. For example, you might have changed the link with the **defunct** or **rename** commands. Or you might have used **Copy** / **Paste** to point the link at a different target directory.

The selected elements become inactive in the workspace (they are removed from the workspace's default group). The status of the elements becomes **(backed)**.

Tip: When you invoke this command in a dynamic stream, the depot's *pre-promote-trig* trigger fires. This is because, like a *Promote*, the command changes which version of the element appears in the stream.

For a detailed discussion of the ***Revert to Basis*** command, see *Chapter 10 Notes on Revert to ... and Diff Against... GUI Commands* in the AccuRev *Technical Notes*.

Revert to Most Recent Version

(This command is enabled only in a workspace, and only for file elements with **(modified)** status. Link and directory elements cannot have **(modified)** status.)

This command replaces (by invoking the *Populate* command) each selected file with the most recent version you created with Keep. (This command does not undo any namespace changes you made with ***Rename*** or with ***Cut***-and-***Paste***.) The selected elements remain active in the workspace. That is, they are *not* removed from the workspace's default group. The **(modified)** status of the elements changes to **(kept)**.

Use this command when you have saved one or more intermediate versions of the files, and you want to discard further changes you have made since a Keep.

If you have modified a file but not yet performed a Keep on it, this command works like the *Revert to Basis*.

Send to Workspace

The ***Send to Workspace*** command activates the specified version of an element (or set of elements) in a particular workspace. All of the elements must currently be inactive that is, have a **(backed)** status -- in the workspace. This command:

- Copies the contents of the specified version from the repository to the workspace tree.
- Changes the status of elements to **(member)**, indicating that it is in the workspace's default group.

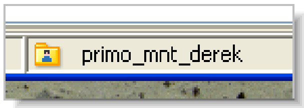
This command is a more generalized variant of the ***Anchor*** command, which always operates on the version of an element that is currently in the workspace.

Invoking the Send to Workspace Command

You can invoke ***Send to Workspace*** in any of these contexts:

- In the ***Version Browser***, choose ***Send to Workspace*** from the context menu of any version.
- In the ***Versions pane*** of the History Browser (displaying the history of an individual element), select one or more versions and choose ***Send to Workspace*** from the selection's context menu.
- In the ***Stream Version Browser***, choose ***Send to Workspace*** from the context menu of any version.

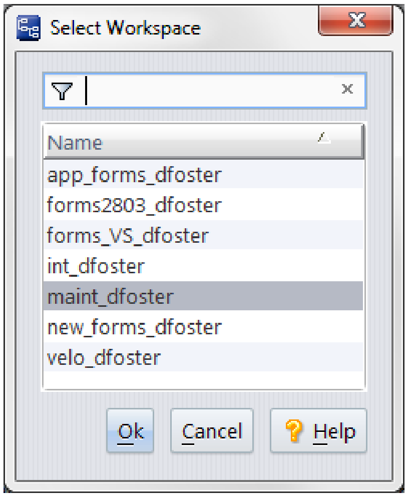
In each of the above cases, you have a current workspace context -- the workspace name is displayed at the bottom of the GUI window. AccuRev sends the selected versions to this workspace.



You can also invoke this command from the ***Summary pane*** of the History Browser. Select a single transaction and choose ***Send to Workspace***. In this case, you don't necessarily have a current workspace context, AccuRev prompts you to select one of the workspaces belonging to you. All the versions in the selected transaction are sent to the workspace you select.

Using the Send to Workspace Dialog

Specify a workspace by:



- double-clicking it, or
- selecting it and clicking ***Ok***

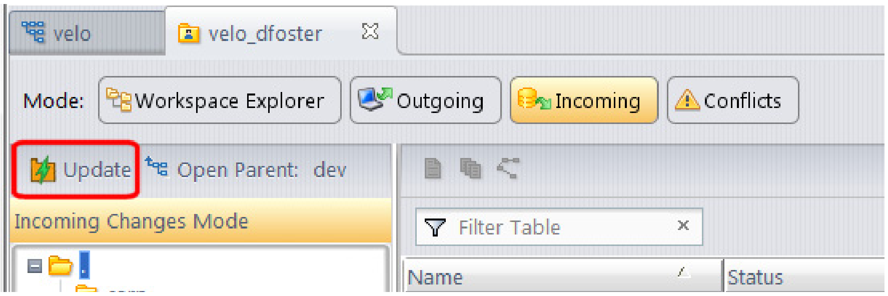
You can click the **Name** column header to reverse the sort order of the workspace names.

Update: Keeping Your Workspace Current

Video Tip	Click *this link* to view an introductory video on updating your workspace.

The principal purpose of an AccuRev workspace is to provide an isolated location for performing development tasks. The changes you make do not affect your colleagues until you decide to make them public, using the *Promote* command. Likewise, the changes that others make do not immediately affect your workspace. You must execute an *Update* command to bring versions created (and then promoted) by your colleagues into your workspace.

You invoke the *Update* command in the File Browser, using the *File* > *Update* command or a button above the Workspace Explorer.



During execution of the ***Update*** command, a progress window appears (See *"The Update Progress Box"* on page 132.). A user preference controls how this window is used.) See *"AccuRev Preferences (Tools > Preferences Command)"* on page 39.

Tip: Prior to executing the ***Update*** command, use Incoming Changes mode (Incoming) to see what the Update will do. See *Incoming Changes Mode* on page 70 for more information.

Kinds of Changes Involved in an Update

At its simplest, an update just copies versions of some file elements from your workspace's backing stream into the workspace. For example, your colleagues may have edited the contents of files *colors.java* and *main_menu.java*, created new versions of them in their workspaces, then promoted the versions to the common backing stream. When you invoke *Update*, the new versions of those two files are copied from the depot to your workspace, overwriting the older versions of the file.

In addition to incorporating such content changes into your workspace, *Update* incorporates namespace changes:

- renaming of a file
- moving of a file to another directory
- creation of new files and directories
- defuncting of existing files and directories

AccuRev tracks namespace changes in the same way as content changes -- by saving each change as a new version in your workspace. If you *Rename* file *colors.java* to *colours.java*, the change is recorded as a new version of the file. Changing the name again, to *hues.java*, creates another new version.

Similarly, defuncting file *rgb.java* creates a new version of the file in your workspace.

> **Note:** See *"Renaming a Modified File Before Keeping It"* on page 118.

How Update Works

The following sections provide an overview of how *Update* works.

Deciding Which Elements to Update

The basic *Update* strategy is to leave the files you're working on undisturbed, and to copy any new versions of other files into the workspace. This enables the workspace to provide the safety of isolation, while still "keeping in touch" with other users' progress.

Roughly speaking, ***Update*** partitions the files in your workspace tree into two categories, to determine which are candidates for updating:

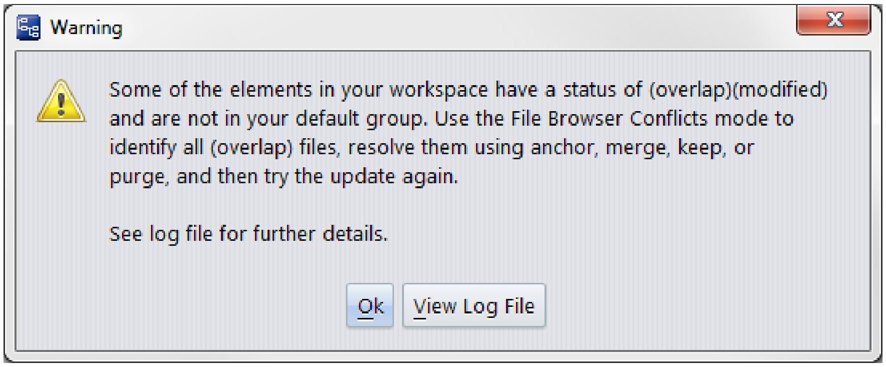
- **Files that you're currently working on.** AccuRev uses the concept of default group to keep track of the files you're working on. Files are placed in the default group when you process them with such AccuRev commands as ***Send to Workspace***, ***Keep***, and *Rename*. These files are not candidates for updating, because it would overwrite your changes.
- **All the other files.** The files that you're not currently working on are candidates for updating. AccuRev will update all such files for which a more recent version exists in the backing stream.

But things are not quite this simple. You might have edited other files, too, without preserving the changes with the ***Keep*** command. Such files have **(modified)** status.

In many cases, no one else has been working on those files. If no new versions of the **(modified)**-status files have been created in the backing stream since your most recent ***Update***, the update proceeds, leaving those files alone.

But suppose one or more of the files *does* have a new version in the backing stream. In this case, the file's status is **(modified)(overlap)**. ***Update*** won't simply overwrite the file with the backing-stream version, but it can attempt to merge the file with the backing-stream version:

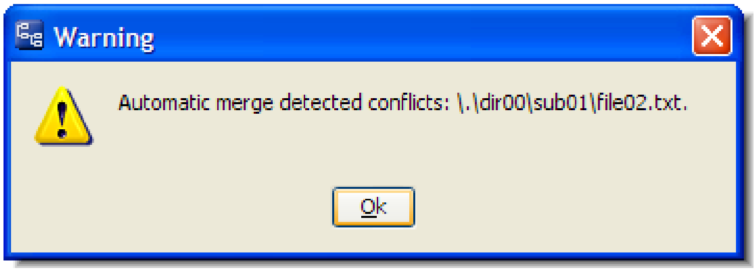
- Unless you have set the ***Update Resolves Trivial Merges*** user preference, AccuRev displays a warning message like this:



Some of the elements in your workspace have a status of (overlap)(modified) and are not in your default group. Use the File Browser Conflicts mode to identify all (overlap) files, resolve them using anchor, merge, keep, or purge, and then try the update again.

See log file for further details.

- If user preference ***Update Resolves Trivial Merges*** is selected, ***Update*** determines whether a merge can be performed automatically on all files with **(modified)(overlap)** status. If so, the update proceeds. If not, the update terminates with an error box:



Automatic merge detected conflicts: \.\dir00\sub01\file02.txt.

Notes:

1. *Members of the default group are "active" elements.*

   AccuRev documentation often uses the term *active* to describe the elements that are members of the default group.

2. *Improving the performance of workspace searches.*

   To improve performance, the Timestamp Optimization algorithm and ignore patterns enable *Update* to avoid examining every file in the workspace tree. (See the AccuRev *Administrator's Guide* for more information on these topics.)

3. *Anchor-required workspaces and exclusive file locking.*

   By default, AccuRev allows you to edit any file in a workspace at any time -- it doesn't require you to perform a "check out" operation on a file before editing it. This provides convenience and flexibility, but the edited files (with "non-member" status) can abort an ***Update***. You can use the anchor-required workspace and/or exclusive file locking feature to ensure files never get "non-member" status, thus guaranteeing that *Update* will always proceed. See *"File Locking"* on page 273.

4. *Determining which elements need to be updated: performance note.*

   This step is efficient and speedy. *Update* needs to consider only the elements that were involved in transactions recorded since the workspace's previous update. Only these transactions can contain changes that have not yet been incorporated into the workspace.

Transferring Data from the Repository

Update next applies both content changes and namespace changes to elements in the workspace:

- It applies content changes to file elements by copying the backing-stream versions from the repository to the workspace tree. For **(modified)(overlap)-**status files, it merges the backing-stream version with the file in the workspace tree.
- It handles namespace changes to file and directory elements by creating, removing, renaming, and/or moving objects in the workspace tree.

Recording the Update

After AccuRev has completed updating versions in the workspace, the workspace is "up to date as of transaction *N* "; *N* is the workspace's update level. The higher this value, the fewer transactions your next invocation of *Update* will need to examine, in order to determine which elements need to be updated.

The Update Progress Box

The ***Update*** command displays the command's progress in a pop-up window. This window displays the filenames being processed one-at-a-time or in a scrolling text pane, depending on the setting of the ***Show Progress Log*** user preference.

You can use the following command buttons in the progress box:

View Full Log

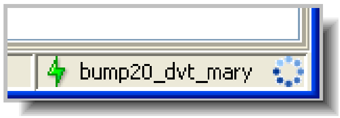
(enabled after command completes) Display the entire ***Update*** log in a separate text-editor window.

Save Log As

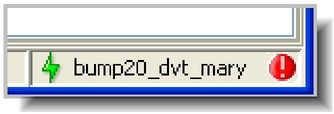
(enabled after command completes) Create a text file containing the entire ***Update*** log.

Run in Background

Close the progress box, but continue command execution. You can proceed to perform other work in the AccuRev GUI window. A progress indicator appears in the lower right corner of the GUI window.



The indicator disappears when the command completes. If an error occurs during command execution, the progress indicator changes to a flashing red error indicator:



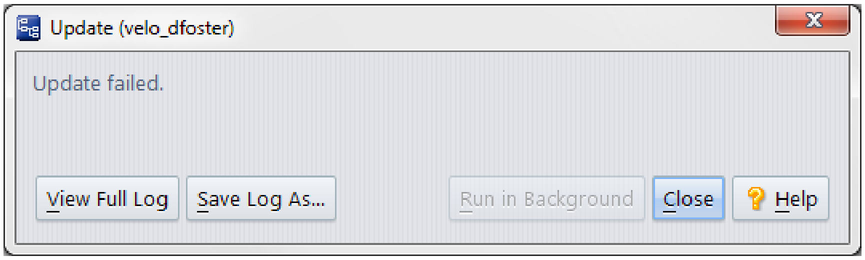
Click on the indicator to restore the progress box. The update log will include a description of the error.

Cancel (while command is in progress)
Close (after command completes)

Close the progress box. If the ***Update*** command is still executing, cancel it.

When Update does not Work

Sometimes AccuRev is not able to complete an *Update* operation, typically because a file cannot be overwritten due to permissions, or because it is open in another application. In this case, you will see an "Update Failed" warning. Use the **View Log File** button on this dialog to display the details of the failure, which should allow you to troubleshoot and correct the issue.


Update (velo_dfoster)

Update failed.

More on Update ...

For a more detailed description of how *Update* does its work, see *The Update Algorithm* on page 26 in *AccuRev Technical Notes*.

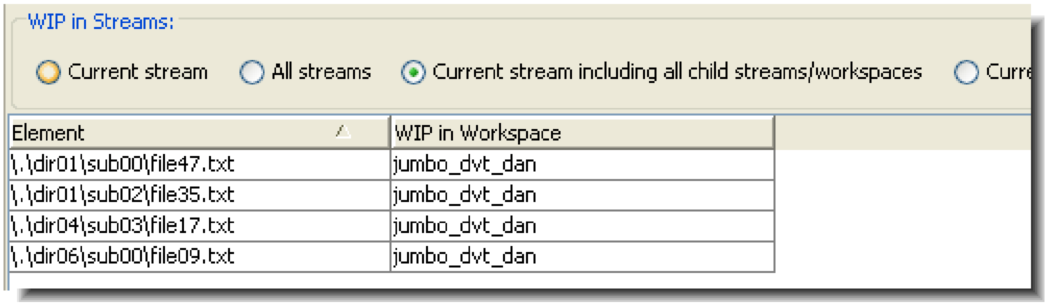
WIP: Identifying Work in Progress

The WIP ("Work in Progress") tab shows elements that are currently active in the workspaces below a specified set of one or more streams. (A stream's own default group is not included in this listing.)

Opening a WIP Tab

Select a dynamic stream in the StreamBrowser, and choose ***WIP*** from its context menu.

WIP Tab Layout



WIP in Streams:

○ Current stream ○ All streams ● Current stream including all child streams/workspaces ○ Curre

Element	WIP in Workspace
\.\dir01\sub00\file47.txt	jumbo_dvt_dan
\.\dir01\sub02\file35.txt	jumbo_dvt_dan
\.\dir04\sub03\file17.txt	jumbo_dvt_dan
\.\dir06\sub00\file09.txt	jumbo_dvt_dan

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

WIP in Workspace

The workspace in which the element is active.

Working in a WIP Tab

Use radio buttons to specify which stream -- or set of streams -- whose workspaces are to be searched for active elements:

Current stream

The stream you selected when invoking the ***WIP*** command.

All streams

All streams in the depot.

Current stream including all child streams/workspaces

The entire subhierarchy of streams below the one you selected.

Current stream including all parent streams/workspaces

The entire backing chain of the stream you selected.

A workspace is searched for active elements if it is a direct child of one of the streams you've indicated with your radio-button choice. (But pass-through streams are ignored when determining the "direct child" relationship. Workspaces whose parent is a pass-through stream are treated as if they were backed but their "grandparent" stream.)

The WIP tab display is informational only. You cannot change the status of elements from this tab.

See also: *Working with Tables* on page 14.

The Default Group of a Workspace or Stream

For each dynamic stream and workspace (but not for pass-through streams, snapshots, or reference trees), AccuRev keeps track of the elements that are currently under active development in that context. This set of elements is called the *default group* of the stream or workspace.

> **Note:** For some purposes, the default group is regarded as containing particular versions of the elements -- for example, when you promote a stream's entire default group.

How Elements Become Members of the Default Group

Workspace

The default group is a property of the workspace stream, maintained in the AccuRev repository. There are no "hidden files" or other administrative structures in the workspace tree to keep track of which elements are active.

The following commands cause an element to become a member of the workspace's default group. Note that these commands can be executed *only* in a workspace, not in a stream.

- Keep
- Rename (or cut-and-paste)
- Defunct
- undefunct (CLI only)
- Merge (ends by performing a Keep)
- Revert (ends by performing a Keep)
- Paste (after a copy, creating a link; CLI command: **ln**)
- Send to Workspace / Anchor

Dynamic Stream

When a version is promoted to a dynamic stream, the element becomes a member of the stream's default group.

If an element is already in the default group of a workspace or stream when one of the commands listed above is executed, the element simply remains a member. In other words, the default group's membership doesn't change when it's viewed as a set of elements. But it does change if you view the default group as a set of particular versions of those elements.

How Elements are Removed from the Default Group

For both a workspace and a dynamic stream, the following commands remove an element from the default group:

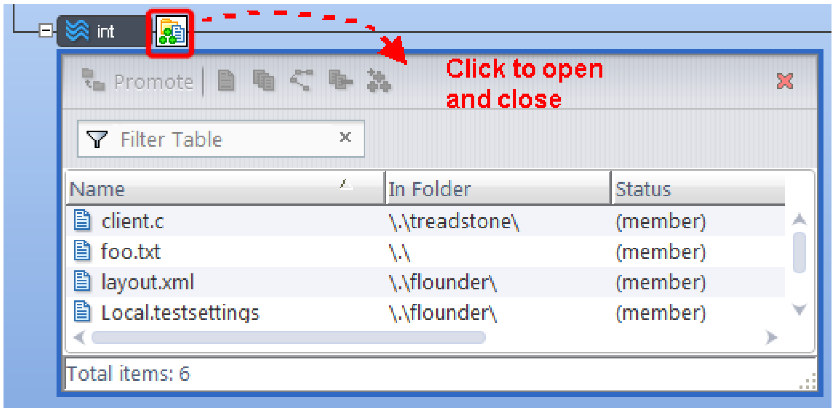
- Promote (to the immediate parent stream)
- Revert to Basis

The element returns to a passive state in the workspace or stream. A stream returns to inheriting a version of the element from its parent stream; a workspace returns to acquiring versions of the element from the backing stream through the Update command.

Cross-promoting an element to a stream that is *not* the immediate parent stream does not remove the element from the default group of the "from" stream.

Viewing the Contents of the Default Group

The StreamBrowser can show the development activity currently taking place in each stream or workspace.



Click to open and close

Name	In Folder	Status
client.c	\.\treadstone\	(member)
foo.txt	\.\	(member)
layout.xml	\.\flounder\	(member)
Local.testsettings	\.\flounder\	(member)

Total items: 6

A control below the stream or workspace opens or closes a subwindow that displays the details of the development activity. The activity details can appear in several ways -- by element, by transaction, or by issue record.

For a dynamic stream, the by-element display is a listing of the stream's default group. For a workspace, this subwindow displays those members of the default group that are displayed in the File Browser’s Outgoing Changes mode (see discussion below).

Default Group vs. Outgoing Changes in a Workspace

A workspace's set of active elements (that is, its default group) is not necessarily the same as the set of elements that have changed in the workspace. In practice, the sets will often be the same. But in theory, the sets are overlapping, but not identical:

- The Default Group shows only those elements with a status of **(member)**. This status can be the result of "checking out" elements using Send to Workspace or Anchor commands, or simply the result of adding a new file to the workspace.
- The Outgoing Changes mode includes elements in the Default Group, but it can also show external objects and missing elements.

Real Versions and Virtual Versions

The difference between the ***Keep*** and ***Promote*** commands highlights an important aspect of the way that AccuRev organizes and manages development data. It also highlights the difference between *backing streams* and *workspace streams*.

Real Versions

All 'real' development -- both content changes and namespace changes -- takes place in workspaces, not in dynamic streams or reference trees. The following commands create a new **real version** of some element in the built-in workspace stream of some AccuRev workspace.

- Keep
- Rename (or cut-and-paste)
- Defunct
- undefunct (CLI only)
- Merge (ends by performing a Keep)
- Revert (ends by performing a Keep)
- Paste (after a copy, creating a link; CLI command: **ln**)

Virtual Versions

Unlike the commands listed above, the ***Promote*** command does not record a change to the contents or pathname of any element. Rather, it makes an existing real version available in a new location: a particular dynamic stream. The version that ***Promote*** creates in a dynamic stream is called a **virtual version**; a new virtual version is a pointer to, or alias for, an existing real version in some workspace stream.

Virtual Versions in Workspace Streams

The commands ***Send to Workspace*** and ***Anchor*** are special. They create a new version in a workspace stream, but the new version is *virtual*, not real. Why? It's for consistency with the above descriptions of real and virtual versions:

- These commands do not preserve a development change, so it doesn't make sense to create a new real version.
- These commands make an existing real version available in a new location: a particular workspace -- so it makes sense to record the action as a virtual version.

AccuRev Element Status Reference

Development work involves making changes to elements -- both files and directories. As files get edited in workspaces, new versions get created in workspace streams, and existing versions get promoted to higher-level streams, AccuRev shows you the current status of each element. Keep in mind that the status of an element in your workspace may be different from its status in other workspaces and streams.

An element's **file status** is indicated by a set of one or more status indicators (or flags), each in the form of a parenthesized word. For example:

```
    (overlap)(kept)(member)
```

Even though we use the term "file status", each directory element also has a status in each workspace and stream. A subset of the status indicators is used to report directory statuses.

These controls affect the File Browser's display of elements and their file statuses:

- User Preferences
  - Show External Objects
  - Include Ignored Objects
- Timestamp Optimization (File Browser checkbox; see *Chapter 5 Timestamp Optimization (TSO)* in the *AccuRev Administrator's Guide* for more information)

File Status Indicators

The following sections detail the AccuRev status indicators, organized by category:

Link-Related Indicators

(elink)

The element is an element link. (The link's icon is 📄.)

(slink)

The element is a symbolic link. (The link's icon is 📄.)

(missing-target)

For an element link, the target element is not present in the workspace or stream. This can occur if the target element is removed from the workspace tree by an operating system command. It can also result from an ***Include from Stream*** or ***Exclude*** command. For a symbolic link, there is no object at the target pathname.

(modified-target)

For an element link, the target element has been modified (either a content change or a namespace change) in the workspace or stream.

(corrupted)

For an element link in a workspace, AccuRev and the operating system disagree on the link target. That is, the target element recorded in the AccuRev repository differs from the target in the operating system's instantiation of the link in the workspace tree. This can occur if you modify or replace a link using operating system commands instead of AccuRev commands.

A cross-linked element (see **(xlinked)** below) gets **(corrupted)** status if AccuRev does not overwrite the element during execution of the Include from Stream command, because the element has **(modified)** or **(kept)** status in the workspace. This should not occur during normal operation.

Presence of the Element in the Workspace or Stream

(defunct)

The element has been marked for removal from the workspace or stream with the *Defunct* command.

Workspace: the element has already been removed from the workspace tree (local disk storage). It will be removed from the workspace stream (in the depot) when you *Promote* the element to the parent stream.

Dynamic stream: the element will be removed from the stream (in the depot) when you *Promote* the element to the parent stream.

(external)

(workspace only) The file or directory has not been placed under version control. (There's an object in the workspace tree, but no element has been created in the workspace stream.)

(ignored)

(workspace only) An external file or directory that matches a pattern specified in the **.acignore** file or the **--ignore** option for the **add**, **stat**, and **files** CLI commands.

(excluded)

The element does not appear in the workspace because it has been excluded, using the ***include/exclude facility***. For file elements, it's likely that the exclusion was explicitly set on the directory in which the file resides, or on a higher-level directory that includes the file.

(xlinked)

This version of the element appears in the workspace or stream by virtue of a cross-link ***(Include from Stream*** command) — either on the element itself or on its parent directory or a higher-level directory.

(missing)

(workspace only) The workspace "should" include a version of this element, but doesn't. This occurs when you delete version-controlled files from the workspace tree using operating system commands, or using the AccuRev *Delete* command. If an operation causes the target of an element link to be removed from a workspace, AccuRev removes the element link, also, causing it to have **(missing)** status.

(no such elem)

This status can appear under two circumstances:

***Incoming Changes Mode*:** The element does not currently exist in your workspace, but will be created if you proceed with the update.

Element security: An AccuRev administrator has changed your access to an element or namespace with the **eacl** command during your session, and you are now denied access to something that was already displayed in your File Browser.

(twin)

The element is one of multiple elements in the workspace that exist at the same pathname. At most one of these elements can be accessed through the pathname; the other(s) can be accessed through their unique *element-IDs*. Elements with this status appear with a yellow highlight.

See *Resolving (twin) Status* on page 119 to learn different ways to resolve an element's **(twin)** status.

Note: Elements with **(overlap)** status are also highlighted in yellow. See *(overlap)* on page 139 for more information.

(stranded)

The element is active in the workspace or stream, but there currently is no pathname to the element.

See Chapter 8 Version Control of Namespace-Related Changes in *AccuRev Technical Notes* for more information on stranded elements.

Changes to the Element

(modified)

(*workspace only*) The file has been modified in the workspace since the most recent *Update* or *Keep*. You can control how AccuRev determines whether a file has this status: see Chapter 5 Timestamp Optimization (TSO) in the AccuRev *Administrator's Guide* for more information.

(kept)

A new version of the element has been created with *Keep*, *Rename*, or *Defunct* (or the CLI command *undefunct*) and the file has not subsequently been modified, promoted to the backing stream, or purged (*Revert to Basis*).

(member)

The element is *active* in the workspace. Commands that create a new version in the workspace or stream also make the element active there (if it isn't already).

Relationship to the Version in the Parent Stream

(backed)

The version in the workspace stream is the same as the version in the backing stream. And you have not changed the element since the last time you promoted it or purged it with *Revert to Basis*, or since the most recent *Update* of your workspace.

(stale)

(workspace only) The element needs to be updated, because the version in the backing stream has changed since the workspace's latest *Update*. And since you have not changed the element in your workspace, it can be safely updated (no unpreserved changes will be overwritten).

(overlap)

The element has changed both in this workspace or stream, and in the parent stream. This indicates that a merge is required before you can promote your changes to the parent stream. Elements with this status appear with a yellow highlight.

Note: Elements with **(twin)** status are also highlighted in yellow. See (twin) on page 138for more information.

(underlap)

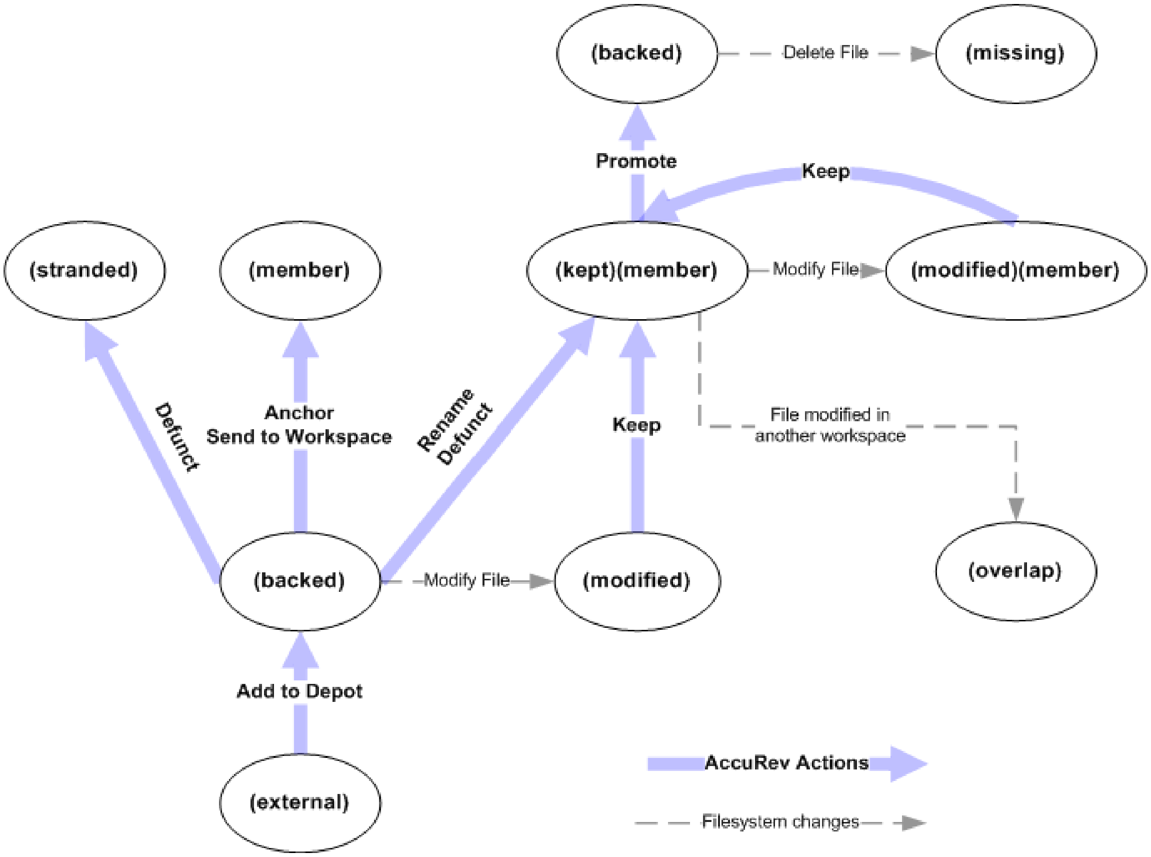
The element has changed both in this workspace or stream, and in the parent stream -- but the parent stream already contains the changes in this workspace or stream. In many cases, it is most appropriate to use **Revert to Basis** to remove your changes from this workspace stream. In other cases, you can merge and promote, just as with an **(overlap)**-status element.

Note: In a workspace, **Revert to Basis** restores the version of an element that was in the parent stream at the time of your workspace's most recent **Update**. You might need to perform another **Update** to bring the version causing the (underlap) status into the workspace.

File Status Diagram

This diagram shows many of the file statuses, illustrating how an element's changing status indicates its development progress. This diagram suggests that as more work is performed on an element, it "rises to a higher level" of development. (That's the idea behind having a command named "promote".) The diagram also depicts the fact that AccuRev tracks changes to an element in two "dimensions":

- It records changes made by AccuRev commands as new versions in the depot.
- It detects that a file's contents have changed at the operating system level.



This diagram captures the following facts:

- Before you start working on an element, it's unchanged along both the AccuRev dimension and the operating-system dimension. Its status is **(backed)**.
- When you modify the contents of a file, it changes along the operating-system dimension only, and becomes **(modified)**.
- The following involve changes along the AccuRev dimension. This makes the element active in the workspace, so its status indicates that it is a **(member)** of the default group.
- When you *Keep* a file that you've modified, its status becomes **(kept)**.
- An element can also achieve **(kept)** status through a namespace change -- *Rename* or *Defunct* command. This is a change along both the AccuRev and operating-system dimensions.

Note: The CLI command **undefunct**, which reinstates a previously defuncted object, works similarly. This command is not in the GUI.

- The *Anchor* command change a file along the AccuRev dimension only. The file's status becomes **(member)**.
- You can create any number of intermediate versions of a file in the workspace, by repeatedly modifying the file then keeping it. Throughout this process, the file remains a **(member)** of the workspace's default group. In addition, its status toggles between **(kept)** and **(modified)**.
- You can *Promote* an element whose status is **(kept)**. This returns the element's status to **(backed)** -- the workspace now uses the newly promoted version in the backing stream.

Details Pane Table Columns Reference

The table in the Details pane typically displays Name, Status, and Version columns. This section describes these and other columns that can appear in the Details pane.

Tip: You can add and remove columns and change their order. See *Working with Tables* on page 14 for more information.

Name

The name of the object, either a file or a directory, in the workspace or stream. AccuRev allows you to rename and relocate objects, so a file might have a different name and/or a different directory location in another workspace or stream.

When the Details pane is displaying the contents of a single folder, the *Name* column shows the name of each object in the folder. The highlight in the Explorer pane indicates which folder’s contents are being displayed.

The Display of Element Names in Tables user preference lets you specify whether you want to display element pathnames as a single column (*Element*) or two columns (*Name* and *In Folder*). See *Controlling the Display of Element Names* on page 57 for more information.

Element

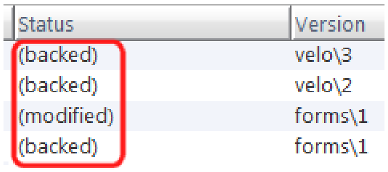
(*Available only in Outgoing Changes, Incoming Changes, and Conflicts modes*) The full path and name of the element.

In Folder

(*Available only in Outgoing Changes, Incoming Changes, and Conflicts modes in the workspace File Browser and in Conflicts mode in the stream File Browser*) The full path of the element.

Status

One or more keywords that indicates the AccuRev status of the element in this workspace or stream. As shown in the following illustration, status keywords are displayed in parentheses.



Status	Version
(backed)	velo\3
(backed)	velo\2
(modified)	forms\1
(backed)	forms\1

file status indicators

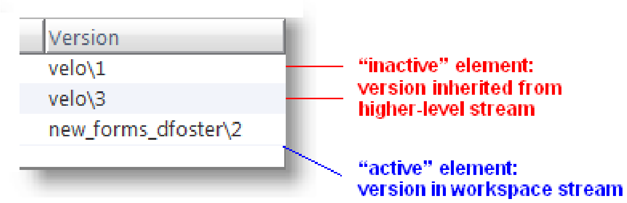
See *AccuRev Element Status Reference* on page 136 for more information.

Version

A version-ID that indicates the version of the element that currently appears in this workspace or stream.

- **Workspace**: If you have modified an element, but not yet promoted it to the backing stream, the version-ID indicates a version in the workspace stream. These are described as "active elements", or as "members of the workspace's default group". For inactive elements (those you have not modified or activated with *Anchor* or *Send to Workspace*), the Version column indicates a version from some higher-level stream, which your workspace stream inherits.

  If the Version column is blank, the file or directory has not yet been placed under version control. Such objects have the status *external*.



- **Stream**: If the element is currently active in the stream (is in the stream's default group), the version-ID includes the name of this stream. Otherwise, the version-ID includes the name of the higher-level stream from which this stream inherits the version.

Action

(*Available only in Incoming Changes mode in the workspace File Browser and in Outgoing Changes mode in the stream File Browser*) The actions AccuRev will perform on an element when updating your workspaces with changes from the parent stream. For example, if the element’s status is **(no such elem)** -- meaning this element does not exist in your workspace -- the value displayed in the Action column might be *create*.

Real

(*Available only in Incoming Changes view in the workspace File Browser and in the Explorer mode in the stream File Browser*) A version-ID that indicates the basis version whose changes will be incorporated in the workspace when the workspace is updated.

New Name

(*Available only in Incoming Changes mode in the workspace File Browser*) The name AccuRev will use for an element that has been moved or renamed.

ElemType

File element: the element type of this version of the element: **text**, **ptext**, or **binary**.

Directory element: the keyword **dir**.

Link element: the keyword **link**.

Id

The element-ID of the element.

Last Modified

A date-timestamp, indicating the last time this element was modified. In a workspace, this is the same as the timestamp reported by the operating system for the object in the workspace tree.

Size

The size of this version of the file (or the file version referenced by this version of an element link or a symbolic link). The size of a directory is reported as "0KB".

Lock Type

Either **parallel** or **serial**, indicating the element's current development mode. See *The Locks Command* on page 52.

Parent EID

(*Available only in Outgoing mode*) The element-ID of the element’s parent stream.

Overlap Stream

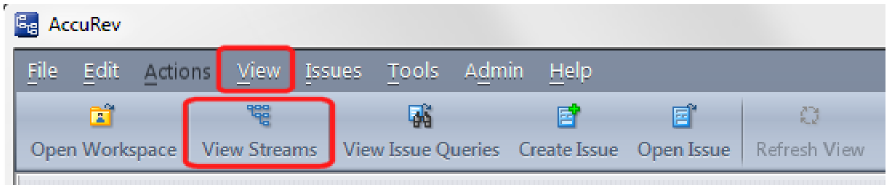
(*Available only in Conflicts mode*) The workspace or stream that contains a version with **(overlap)** status.

4. The StreamBrowser

AccuRev's unique StreamBrowser enables graphical control over the entire configuration management environment, in a way that is simple, flexible, and powerful. Using the StreamBrowser, you can view and manipulate all the streams in a depot, including workspace streams and snapshots.

Opening a StreamBrowser Tab

You can open a StreamBrowser tab to display the streams of the current depot in any of these ways:

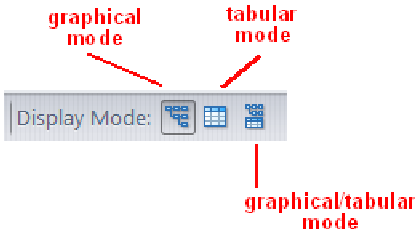


- Choose *View* > *Streams* from the GUI main menu.
- Click the *View Streams* button on the GUI main toolbar.
- Choose *View* > *Streams* from the context menu of a depot on a *Depots* tab.

If there is no current depot, AccuRev prompts you to select one.

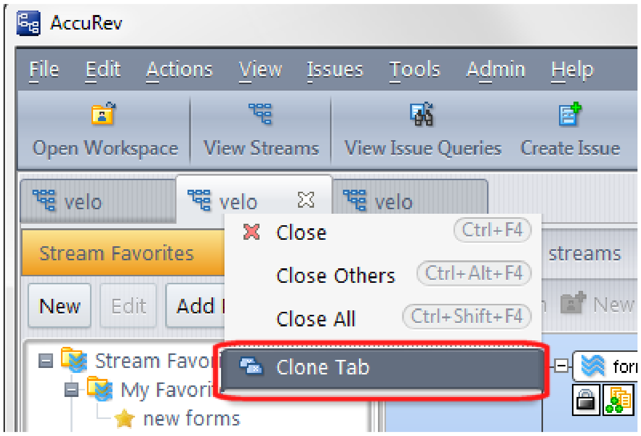
StreamBrowser Tab Layout

You can use the StreamBrowser in any of the following display modes, which you select using buttons on the *StreamBrowser toolbar*:



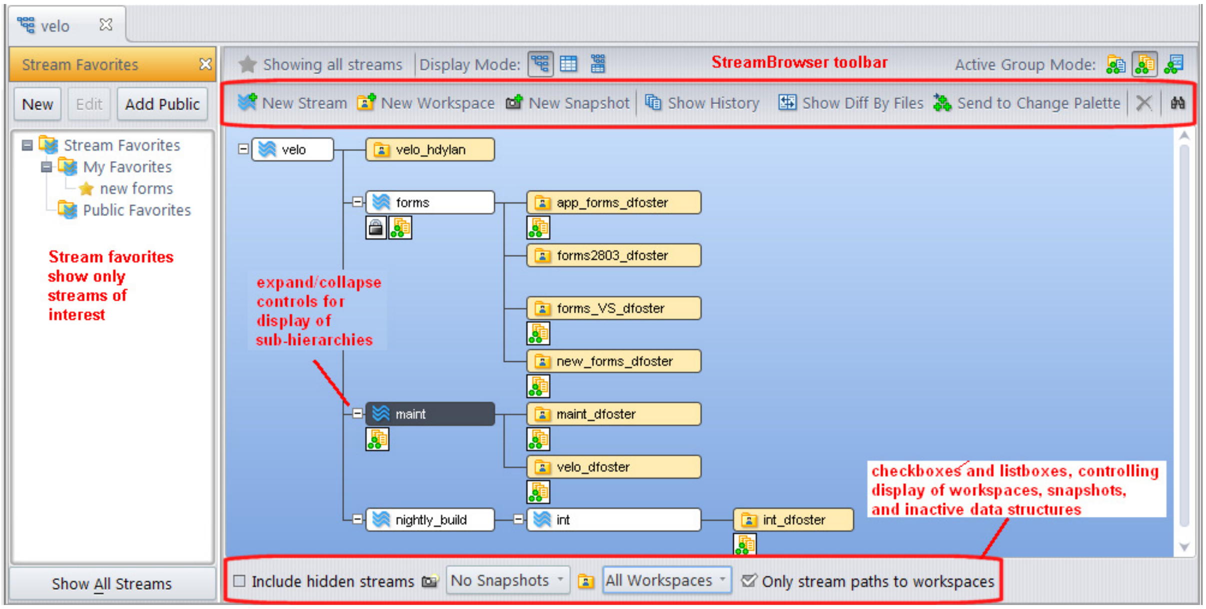
- *Graphical mode*: displays the depot's stream hierarchy as a tree (root node at the upper left)
- *Tabular mode*: displays a table containing the depot's streams, snapshots, and workspaces, initially sorted by stream name.
- *Graphical/tabular mode*: both of the above -- the tab is divided into a graphical pane (above) and a tabular pane (below). A movable separator bar enables you to adjust the space allocation to the panes.

You can have any number of StreamBrowser tabs open at the same time, each for a different depot. You can also open multiple StreamBrowser tabs on the *same* depot, by invoking the *Clone Tab* command on the context menu of the tab control.



Graphical StreamBrowser Display

The graphical display of a depot's stream hierarchy is organized as follows:



- The depot's root stream (top-level) is at the left edge.
- A given stream's children appear to its right; the children are arranged vertically, in this order:
  - Workspaces, in alphabetical order
  - Dynamic streams with no basis time and pass-through streams, in alphabetical order (capitalized stream names appear first, followed by stream names starting with the characters **[**, ****, **]**, **^**, **_**, and **`**, followed by non-capitalized stream name)
  - Dynamic streams with basis times and snapshots, ordered by basis time (most recent first)
- Each stream is annotated with an icon that indicates the kind of stream:

  dynamic ("normal") stream

  time-based dynamic stream

workspace stream

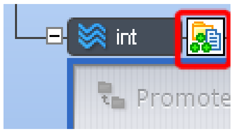
snapshot (static stream)

pass-through stream

gated stream

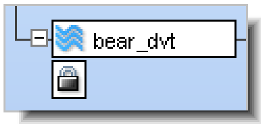
staging stream

- Each stream or workspace that has active elements (that is, a non-empty default group*)* is displayed with a special control that opens and closes a *subwindow showing the activity's details*.



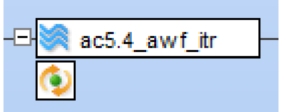
See *The Default Group of a Workspace or Stream* on page 134 for a general discussion of the default group.

- Each stream can have a lock, which prevents certain operations involving that stream from being performed.

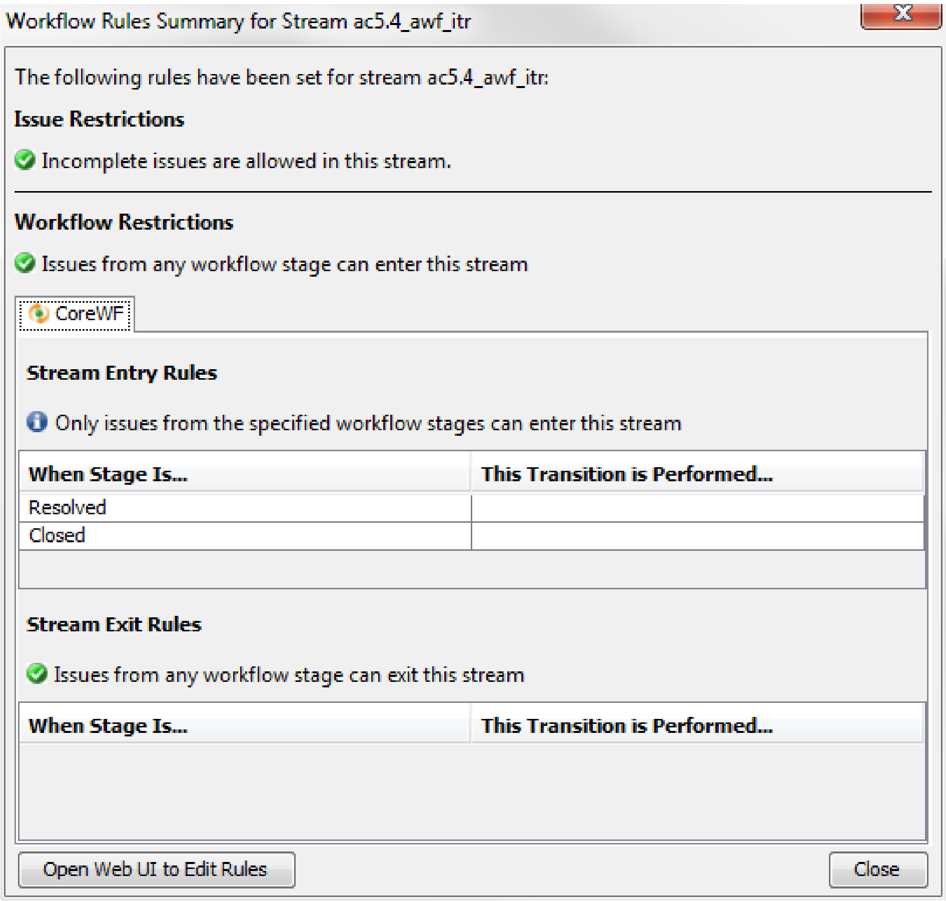


If a lock exists on a stream, a special control appears, which provides access to *The Locks Dialog Box* on page 52.

- Each stream may be associated with a *workflow*, available through the AccuRev *workflow* feature. If workflow rules have been set for a stream, a workflow icon appears.



Clicking the workflow icon displays the Workflow Rules Summary dialog box, from which you can review the rules that have been set for that stream. Clicking the **Open Web UI to Edit Rules** button opens the AccuRev Web UI, from which you can edit existing rules or create new ones.



Workflow Rules Summary for Stream ac5.4_awf_itr

The following rules have been set for stream ac5.4_awf_itr:

Issue Restrictions

✓ Incomplete issues are allowed in this stream.

Workflow Restrictions

✓ Issues from any workflow stage can enter this stream

Stream Entry Rules

Only issues from the specified workflow stages can enter this stream

When Stage Is...	This Transition is Performed...
Resolved	
Closed	

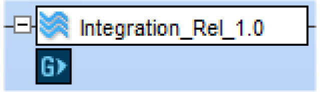
Stream Exit Rules

✓ Issues from any workflow stage can exit this stream

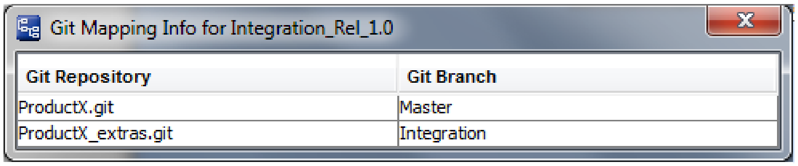
When Stage Is...	This Transition is Performed...

Workflow rules are defined in the AccuRev Web UI. See "Using Workflow to Manage Issues" in the *AccuRev® Web Interface User's Guide*, Version 2012.1 or later for information about workflow.

- Each stream may be mapped to a branch in a Git repository, via the optional AccuRev GitCentric™ product. If a stream is mapped to a Git branch, a GitCentric icon appears on the stream.



Clicking the GitCentric icon displays information about any Git branches and their repositories that are mapped to this stream via GitCentric.



Git Mapping Info for Integration_Rel_1.0

Git Repository	Git Branch
ProductX.git	Master
ProductX_extras.git	Integration

See the AccuRev GitCentric documentation for more information

See *Controlling the Display* on page 149 to learn about the ways you can filter the StreamBrowser display.

Tabular StreamBrowser Display

In tabular display mode, the StreamBrowser tab contains a table listing some or all of the current depot’s streams (including snapshots and workspaces). If you have applied a Stream Favorite to restrict the display, only the stream paths and children (if any) for the selected streams appear in the table. (A stream path is a subset of the stream hierarchy that starts at the root stream and ends at the stream being referenced.)

The checkboxes and list box at the bottom of the tab work the same way as in graphical mode to control which streams are included in the table. The StreamBrowser’s tabular mode offers the *same commands* (toolbar and context menus) as the graphical mode.

See also: *Working with Tables* on page 14.

Working in the StreamBrowser

The following sections describe working in the StreamBrowser. Note that many of these operations can be performed only in the graphical display, not in the tabular display.

Controlling the Display

Initially, the StreamBrowser displays all of the depot's currently active dynamic streams, along with workspaces for the current user. It does not display any snapshots, nor any item that you have deactivated with the *Remove* command.

Note: If you have applied a Stream Favorite, only a subset of the streams in the depot appear.

Applying a Stream Favorite

A *Stream Favorite* is a way to limit the stream hierarchy shown in the StreamBrowser to the stream path of a specific stream and all its children. See *Using Stream Favorites* on page 160 for more information.

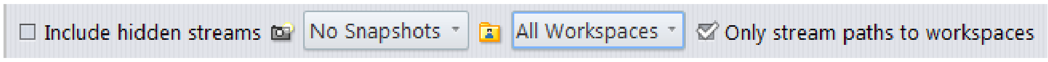
Filtering Users/Groups

The **View** > **Users/Groups** menu option allows you to restrict the set of users and groups displayed in the StreamBrowser and other locations in the AccuRev GUI. See *Filter Users/Groups (View Users/Groups Command)* on page 38 for more information.

Tip: You can also create Stream Favorites to limit the stream hierarchy shown in the StreamBrowser and any list of streams available in the AccuRev GUI. See *Using Stream Favorites* on page 160 for more information.

Filtering Streams, Snapshots, and Workspaces

You can adjust the numbers and types of streams shown by using the controls at the bottom of the StreamBrowser tab:



Include hidden streams

This checkbox is a toggle switch: it reveals/hides the data structures that have been *Remove*'d. (The *Remove* command doesn't actually delete anything from the depot; the data structure just becomes invisible and inactive.)

Snapshots

This drop-down list controls how many snapshots are visible in the StreamBrowser. It contains the choices *No Snapshots* (the default), *All Snapshots*, *Past week*, *Past 2 weeks*, and *Past 4 weeks*. (Check the *Include hidden streams* checkbox if you want to see snapshots that have been *Remove*'d.) No matter what the setting, a snapshot remains visible if there are any workspaces or streams below it.

Workspaces

This drop-down list controls which of the depot's workspaces are visible in the StreamBrowser. It contains the choices *All Workspaces*, *No Workspaces*, *Current User* (the default), *Unassigned*, and the name of each registered AccuRev user and group. (Check the *Include hidden streams* checkbox if you want to see workspaces that have been *Remove*'d.) To control which users and groups appear in this list, use the *Filter Users/Groups (View Users/Groups Command)* on page 38 command.

Only stream paths to workspaces

This checkbox is a toggle switch. Select it to filter the streams so that only the stream paths to workspaces -- the streams directly connecting the root stream to the workspace -- are visible in the StreamBrowser. Unselect it to remove the filter. Use the *Workspaces* list to control which workspaces are shown.

Expanding and Collapsing Subtrees

In the graphical display mode, each stream that has "children" (workspaces and/or snapshots) is displayed with an expand/collapse control. Collapsing causes the entire hierarchy below the stream to disappear from the screen. This affects the StreamBrowser display only. It does not affect the operation of the stream in any way.

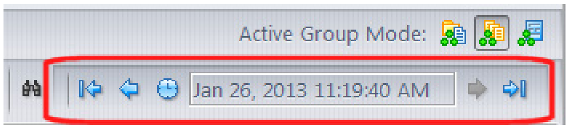
Changing the StreamBrowser View

If your depot has a large or complex stream hierarchy, not all of its streams and workspaces will be visible on your display. To change your view of the stream hierarchy, you can:

- Use the StreamBrowser's horizontal and vertical scroll bars
- Use your mouse to drag the stream hierarchy -- place the pointer on the StreamBrowser background, press and hold mouse button one, and then move the mouse to change the view.

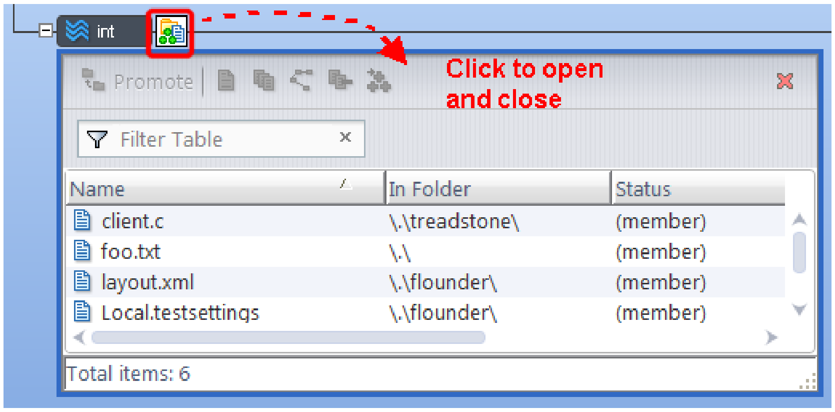
Historical View of the Stream Hierarchy

If the *Enable StreamBrowser History preference* is checked (*AccuRev Preferences (Tools > Preferences Command)* on page 39), a set of *history controls* are added to the toolbar. This enables you to view the stream hierarchy as it existed at any point in the past.



Viewing a Stream's Current Development Activity

The StreamBrowser can show the development activity currently taking place in each stream or workspace.

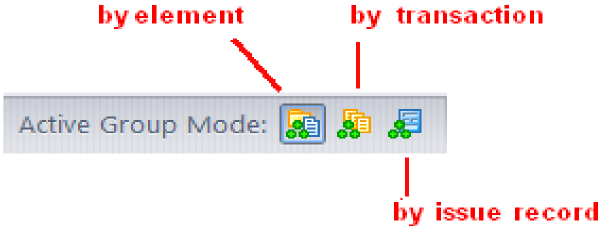


Name	In Folder	Status
client.c	\.\treadstone\	(member)
foo.txt	\.\	(member)
layout.xml	\.\flounder\	(member)
Local.testsettings	\.\flounder\	(member)

Total items: 6

A control below the stream or workspace opens or closes a subwindow that displays the details of the development activity. The activity details can appear in several ways -- *by element*, *by transaction*, or *by issue record*.

Use the development-activity mode controls at the right side of the StreamBrowser toolbar to determine how the activity details will be displayed. (The icons on the controls below the streams and workspace change accordingly.) You can change modes either before or after opening an activity subwindow.

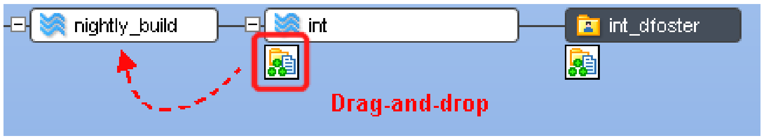


Viewing Activity By Element (Active Files)

When displaying activity by element, the subwindow displays elements with a **(member)** status. You can select one or more of the element versions and invoke commands on them, using the subwindow's toolbar or context menu. The commands available are the same as those in the File Browser. See *Chapter 3 The File Browser* for more information.

You can also perform these drag-and-drop operations:

- Dragging the subwindow control (but not elements within the subwindow) to the parent stream or another dynamic stream performs a *Promote* or *Send to Change Palette* operation on all the element versions in the subwindow.



- Dragging the subwindow control (or a selection of one or more versions within the subwindow) to a workspace performs a *Send to Change Palette* operation.

Viewing Activity By Transaction (Active Transactions)

When displaying a stream's activity by element, the subwindow displays the transactions that created the versions that are currently active in the stream. This display is the same as that in the Summary pane of an Active Transaction tab; see *The Active Transactions Tab (Show Active Transactions)* on page 172. You can select one or more transactions and perform commands on them, just as in the Active Transactions tab.

Viewing Activity By Issue Record (Active Issues)

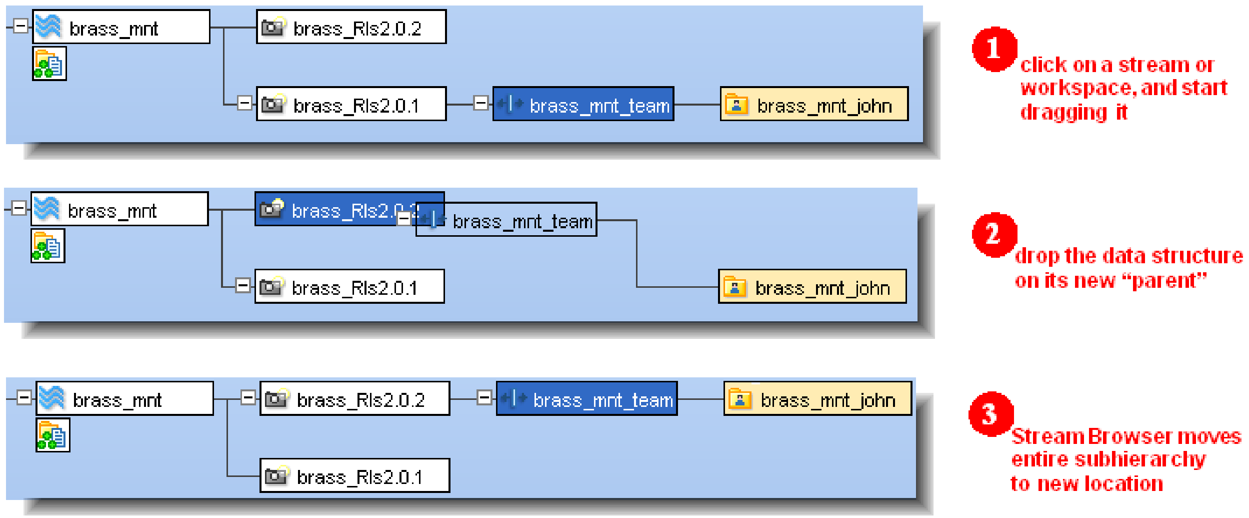
When displaying a stream's activity by issue record, the subwindow displays the issue records that are in a particular stream. This display is the same as that in the Issues pane of a Stream Issues tab (see *The Active Issues Tab (Show Active Issues)* on page 356). You can select one or more issue records and perform commands on them, just as in the Stream Issues tab.

You can also perform this drag-and-drop operation:

> Dragging the subwindow control (but not elements within the subwindow) to the parent stream or another dynamic stream performs a *Promote* or *Send to Change Palette* operation on all the versions in the change packages of the subwindow's issue records.

Reparenting Streams and Workspaces

AccuRev lets you change the backing stream (parent stream) of any dynamic stream or workspace. The StreamBrowser makes it simple: you just drag-and-drop a stream or workspace from its current location in the hierarchy to its new parent.



click on a stream or workspace, and start dragging it

drop the data structure on its new "parent"

StreamBrowser moves entire subhierarchy to new location

The entire subhierarchy moves to the new location. Changing a data structure's location in the stream hierarchy is called **reparenting**. If the StreamBrowser display contains many streams or the stream structure is very complex, an alternative method of reparenting a stream is to select *Reparent* from the context menu of the stream or workspace, then navigate to and click the new parent stream. You are asked to confirm the operation.

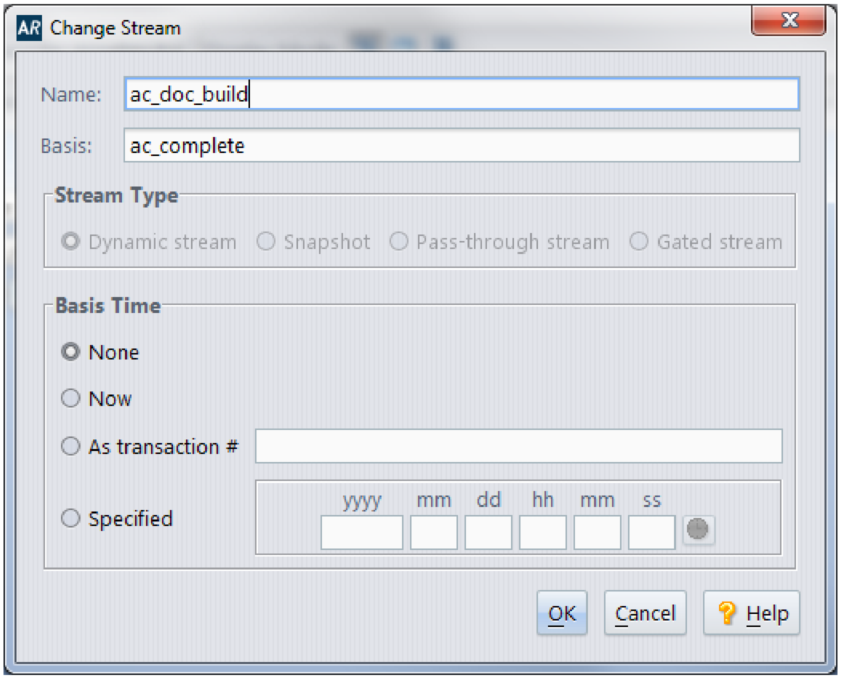
Tip: After you change the location of a workspace, AccuRev prompts you to *update* it. Updating ensures that the workspace contains the correct set of versions, many of which it will inherit from its new parent. Likewise, after changing the location of a stream, make sure you update all the workspaces in the subhierarchy below that stream.

Certain types of streams are subject to rules regarding reparenting:

- **Snapshots --** You cannot reparent a snapshot; both the contents and the parentage of snapshot are fixed permanently.
- **Exclusive file locking workspaces --** You cannot reparent (or change any of the other specifications) of a workspace created with the exclusive file locking feature.
- **Time-based streams --** If you reparent a time-based stream, AccuRev asks whether you wish to retain the basis time or remove it.
- **Staging streams --** You cannot reparent staging streams. You can reparent the gated stream it is associated with, however.

Reconfiguring a Stream

Using the drag-and-drop operation changes the stream's parent. To change other stream values (like its name or basis time, for example), use the *Change Stream* command on the stream context menu.



Change Stream

Name: ac_doc_build

Basis: ac_complete

Stream Type

● Dynamic stream ○ Snapshot ○ Pass-through stream ○ Gated stream

Basis Time

● None

○ Now

○ As transaction #

○ Specified yyyy mm dd hh mm ss

Although you can give a new name to an existing stream, you cannot proceed to create a new stream with the old name. The old name remains associated with its stream. The only way to reuse a stream name is to completely remove the stream's depot from the AccuRev repository, using the AccuRev administration utility, *maintain*.

Setting or Changing a Stream's Basis Time

By default, a stream inherits all versions from its parent, no matter when those versions were created. If you assign a basis time to a stream, it inherits only those versions created before the specified point in time. (The File Browser's *include/exclude* facility provides another way to restrict which versions get inherited from the parent stream.)

Creating New Streams, Snapshots, and Workspaces

The context menu of any data structure in the StreamBrowser includes commands for creating new structures at that point in the hierarchy. *New Stream* and *New Snapshot* display dialogs similar to that of *Change Stream*. The *New Workspace* command invokes a wizard that steps you through the process of defining a new workspace: you specify a name and a location on disk; you can also make some optional settings, such as controlling how line endings in text files are to be handled.

Working with Stream Properties

A *stream property* is a name/value pair you associate with a stream. Values can be strings or files. For example, you might want to create a property named *nightly_build* for your build stream, and specify a log file as its value -- **2_20_14_build.txt**. You can use special stream properties, called *reserved properties*, to set display characteristics of streams in the StreamBrowser. See *Using Reserved Properties* on page 157 for more information. Note that stream properties are not TimeSafe; if you edit or remove a property, you cannot roll back to the prior name/value pair.

This section describes how to create and manage stream properties using the AccuRev GUI. You can also set and manage stream properties -- including reserved stream properties -- using the AccuRev CLI. See *setproperty* on page 229 of the *CLI User's Guide* or the CLI help for more information.

Note: Do not confuse stream properties with configuration settings for a stream (such as stream name and basis time). You manage a stream's configuration settings through the Change Stream dialog box. See *Reconfiguring a Stream* on page 153 for more information.

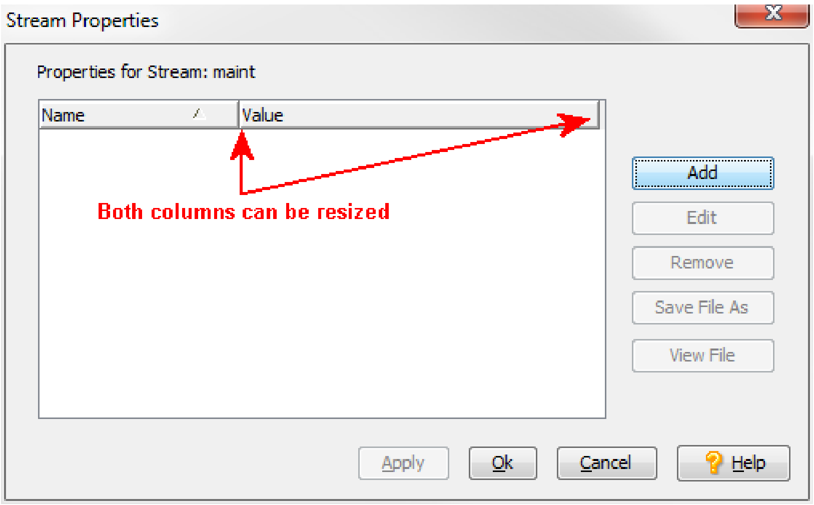
Creating a Stream Property

To create a stream property:

1. In the StreamBrowser, right-click the stream for which you want to set one or more properties and choose **Properties...** from the context menu.

   Alternatives: If the stream already has one or more properties defined for it, you can click the stream properties icon (), or choose **Actions** > **Properties** from the menu.

   The Stream Properties dialog box appears.



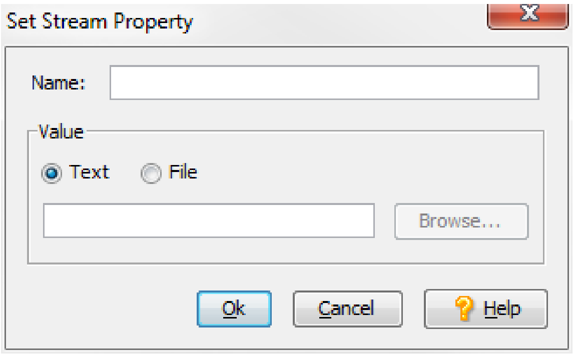
Properties for Stream: maint

Both columns can be resized

   Tip: Both the Name and Value columns can be resized.

2. Click the **Add** button.

The Set Stream Property dialog box appears.



3. In the Name field, enter the name you want to use for the property.
4. In the Value field, select the Text or File radio button based on the property value you want to enter. Then:
   - For a string value, type the value in the entry field.

     Note: You cannot use reserved words *streamStyle* or *streamCustomIcon* when setting stream properties using the AccuRev GUI. See *Using Reserved Properties* on page 157 for more information.
   - For a file value, type the full path and file name in the entry field, or click the **Browse...** button to navigate your file system and select the file. (See *Working with File Type Values* on page 156 for more information on using files for stream property values.)

   Tip: To use a stream property as a "tag" or label, leave the Value field empty.
5. Click **Ok**.

   The Set Stream Property dialog box closes, and the new stream property appears in the Stream Properties dialog box.

   Note: If you specify a file for the stream property value, AccuRev displays only the file’s name, and not its full path, in the Value field of the Properties for Stream: list.
6. If you want to add another stream property, click **Apply** to save your work and then return to *Step 2*. Otherwise, click **Ok** to save your work and close the Stream Properties dialog box.

Editing a Stream Property

When you edit a stream property, you can change its value (including the value type), but you cannot change the property’s name. To rename a stream property, first remove the stream property as described on page 156 and then create a new stream property with the name you want.

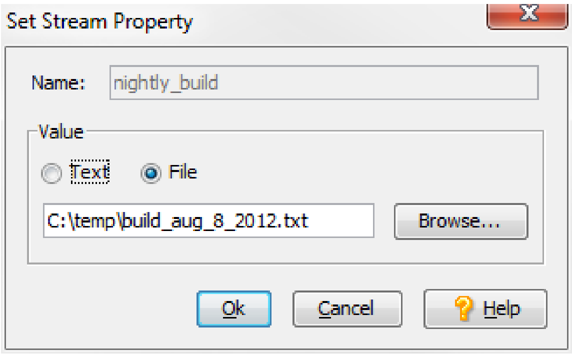
To edit a stream property:

1. In the StreamBrowser, right-click the stream whose properties you want to edit and choose **Properties...** from the context menu.

   Alternatives: Click the stream properties icon () on the stream, or choose **Actions** > **Properties** from the menu.

   The Stream Properties dialog box appears.
2. Select the stream property whose value you want to change and click the **Edit** button.

The Set Stream Property dialog box appears. For example:



Set Stream Property

Name: nightly_build

Value

○ Text ● File

C:\temp\build_aug_8_2012.txt

3. Change the value type and/or value as needed. See *Creating a Stream Property* on page 154 if you need help with this step.
4. Click **Ok**.

   The Set Stream Property dialog box closes, and the revised stream property appears in the Stream Properties dialog box.
5. If you want to edit another stream property, click **Apply** to save your work and then return to *Step 2*. Otherwise, click **Ok** to save your work and close the Stream Properties dialog box.

Removing a Stream Property

To remove a stream property:

1. In the StreamBrowser, right-click the stream whose property you want to remove and choose **Properties...** from the context menu.

   Alternatives: Click the stream properties icon () on the stream, or choose **Actions** > **Properties** from the menu.

   The Stream Properties dialog box appears.
2. Select the stream property you want to remove and click the **Remove** button.

   Tip: You can use **Shift+** click and **Ctrl+** click to select multiple stream properties.

   The properties you selected are removed from the Stream Properties dialog box.
3. If you want to remove another stream property, click **Apply** to save your work and then return to *Step 2*. Otherwise, click **Ok** to save your work and close the Stream Properties dialog box.

Working with File Type Values

If you create a stream property that uses a file as its value, you can:

- Save a copy of the file -- Select the stream property in the Stream Properties dialog box and click the **Save File As** button.

  AccuRev displays the Save As dialog box, which you can use to save a local copy of the file associated with the stream property.
- View a copy of the file -- Select the stream property in the Stream Properties dialog box and click the **View File** button.

AccuRev displays a copy of the file in the editor associated with that file type.

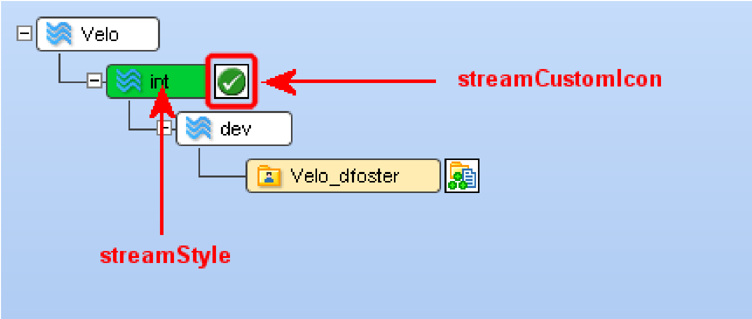
Note that, in either case, you are not working with the file that was uploaded to the AccuRev Server when you created the stream property. If you change the local copy of the file and want to use the new version for the stream property value, you must edit the stream property and specify the file's full path and file name as you did when you first created the property.

Using Reserved Properties

AccuRev supports reserved stream properties **streamStyle** and **streamCustomIcon**, which you can use to modify the display characteristics of any stream you choose:

- **streamStyle** can be used to specify a stream color using a hex value. By default, dynamic streams have a white background and workspaces have a light yellow background. If you want, you can change the background color. You might want to manually highlight streams of interest, or use a script to set a stream color based on events -- to set a build stream to red if the build fails, and green when the build passes, for example.
- **streamCustomIcon** can be used to set characteristics you might want to associate with a build stream. Such characteristics include:
  - Icons that indicate running, success, warning, and failed statuses
  - Customizable tooltips that appear when the pointer is placed over the icon
  - A customizable URL that opens the default browser at the specified address when clicked

You can define both **streamStyle** and **streamCustomIcon** reserved properties for the same stream, as shown in the *int* stream in the following illustration.



Unlike other stream properties, reserved properties can be set only through the CLI. See *setproperty* on page 229 of the *CLI User's Guide* or the CLI help for more information.

Commands Available in a StreamBrowser Tab

Open

(File and context menu only) Using a *File Browser* tab, display the entire contents of the selected stream, snapshot, or workspace.

Find In and Below This Stream

Find the streams whose name includes a specified character string. A *Find* dialog appears.

New Stream

Create a new dynamic stream or pass-through stream, based on (that is, as a child of) the selected stream or snapshot. A *Stream Configuration* dialog appears.

New Workspace

Create a new workspace, based on (that is, as a child of) the selected stream or snapshot. A *New Workspace* wizard helps you to configure the workspace.

New Snapshot

Create a new snapshot, based on (that is, as a child of) the selected stream A *New Snapshot* dialog appears.

Change Stream

(Actions and context menu only) Modify the configuration of the selected dynamic stream or pass-through stream. A *Stream Configuration* dialog appears.

Reparent

(Context menu only) Change the backing stream (parent stream) of the selected dynamic stream or workspace.

Remove

After *confirmation*, hide the selected stream, snapshot, or workspace. This removes the object from the StreamBrowser display.

Removal affects how the object can be used:

- Workspace: deactivates the workspace, so that it cannot be used in AccuRev commands.
- Stream or snapshot: no effect; you can still promote to the stream, create a substream of it, or reparent it.

Note: Removing a workspace that belongs to another user -- You cannot do this using the AccuRev GUI. The *rmws -s* command in the AccuRev CLI provides this capability.

Reactivate

(Actions and context menu only) Undo a *Remove* command for the selected stream, snapshot, or workspace.

Locks ... (Lock/Unlock Stream)

(Dynamic stream only) Lock a stream, to disable various operations that modify the stream. If a stream is already locked, a lock icon appears in the StreamBrowser graphical display, and in the Locks column of the tabular display. A Lock dialog appears: see *The Locks Command* on page 52.

WIP (Work in Progress)

(Actions and context menu only) Open a *WIP* tab, displaying the active elements in workspaces and/or streams related to the selected stream.

Show All Overlaps

(Workspace only) Using an *Overlaps* tab: for active file elements (those in the workspace's default group), display all overlaps involving versions in the workspace's entire backing chain (higher-level streams) -- not just overlaps between the workspace and its immediate parent stream.

Show Active ...

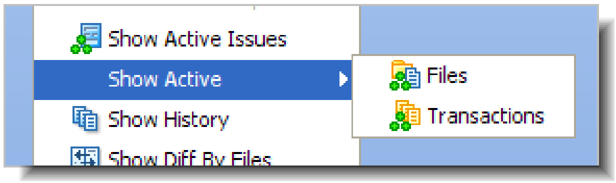
There are three commands to display the stream's development activity, using the appropriate tool for the current *development-activity mode*:

Element mode: open a *File Browser* tab on the selected stream, and execute a search for active elements.

Transaction mode: open an *Active Transactions* tab, showing the transactions that created the versions currently in the selected stream's default group.

Issue Record mode: open a *Stream Issues* tab, showing the change packages that have been (partially or completely) incorporated into the selected stream.

The way the three commands appear on a stream's context menu varies with the development-activity mode: the command for the current mode appears at the top-level of the context menu; the commands for the other two modes are in a submenu. For example, the following illustration shows how the menu choices appear if the StreamBrowser is currently displaying activity by issue record.



Show History

Using a *History Browser*, display the transactions that created versions in the selected stream.

Show Diff By Files

Using a Diff Streams window, show the element-by-element differences between the selected stream and another stream (which AccuRev prompts you to specify).

Show Diff By Issues

(Actions and context menu only) Using a *Stream Issues* window, show the differences between the selected stream and another stream (which AccuRev prompts you to specify). The differences are shown in terms of change packages instead of individual elements.

Send to Change Palette

Send the active elements for the selected stream to the *Change Palette*.

Show Patch List

(Actions and context menu only) Using a *Patch List* tab, list all the individual versions whose changes are in the selected stream, but are not in another stream (which AccuRev prompts you to specify).

Properties

(Actions and context menu only) Displays the Stream Properties dialog box, which lets you set properties for a stream or workspace. See *Working with Stream Properties* on page 154 for more information.

Using Stream Favorites

A *Stream Favorite* is a way to limit the stream hierarchy shown in the StreamBrowser to the stream path of a specific stream and all its children. If your organization has depots containing thousands of streams, Stream Favorites is a powerful tool to improve system performance and focus attention on just the streams you're interested in.

Note: Although you create and apply Stream Favorites in the StreamBrowser, their use affects the display of streams throughout the AccuRev GUI. Any list of streams available in the GUI will contain **only** the subset of streams shown in the StreamBrowser.

Stream Favorites and Stream Filters

Prior to AccuRev Version 5.5, Stream Favorites were called *stream filters*. The idea behind them -- to provide a handy way to create different views of the streams in your depot -- is the same, but there are a few important differences:

- Stream Favorites are created and managed in a more straightforward way
- Stream Favorites can be shared with other users in the depot
- Stream Favorites are accessed through the StreamBrowser

If you are upgrading from AccuRev Version 5.4.1 or earlier, AccuRev automatically converts existing stream filters to Stream Favorites.

Stream Favorites and StreamBrowser Display Filters

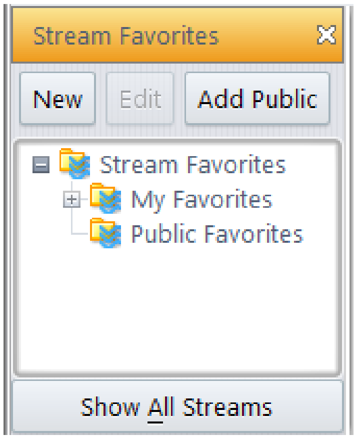
When AccuRev displays streams in the StreamBrowser, it applies Stream Favorites first. Any display controls in the StreamBrowser are then applied to the filtered set of streams. Occasionally, the interaction between Stream Favorites and display controls might produce an unexpected result. For example, if you have selected the **Only stream paths to workspaces** checkbox at the bottom of the StreamBrowser, only those paths that lead to workspaces will be displayed, even if the current Stream Favorite includes streams that do not lead to workspaces. If your Stream Filter does not produce the expected display in the StreamBrowser, remember to check the display controls to verify that these local settings are not affecting the outcome.

Creating a Stream Favorite

To create a Stream Favorite:

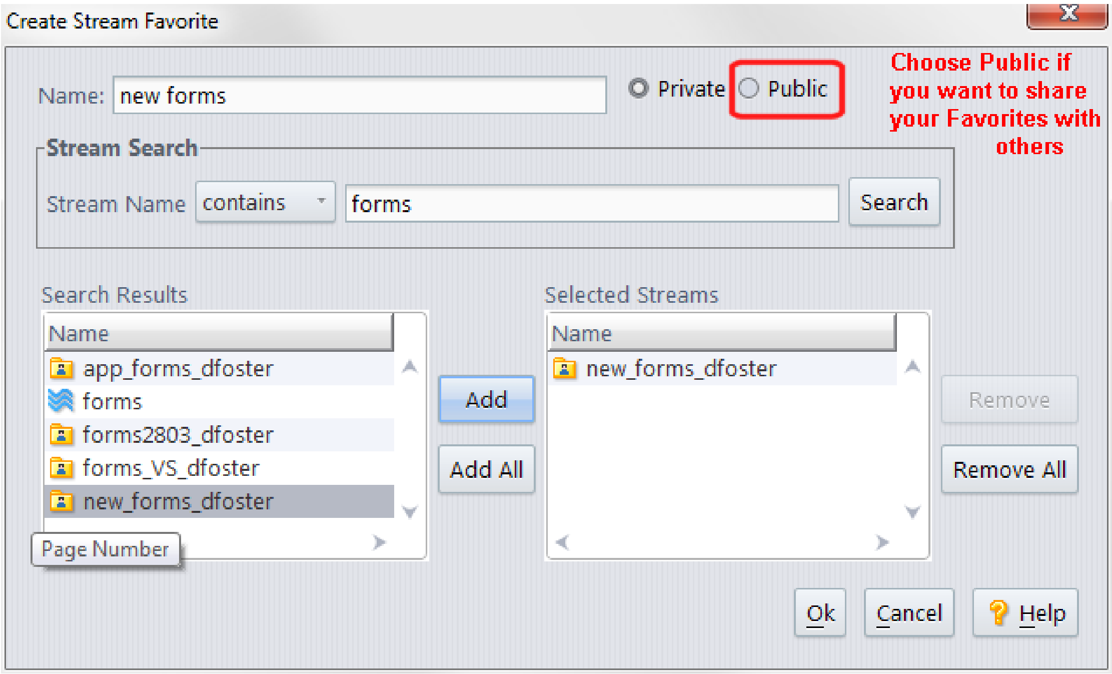
1. In the StreamBrowser, click the Show Stream Favorites button (⭐).

   The Stream Favorites panel appears.



2. Click the **New** button.

   The Create Stream Favorite dialog box appears.



Name: new forms   ● Private ○ Public

Stream Name contains forms

Search Results

Name
app_forms_dfoster
forms
forms2803_dfoster
forms_VS_dfoster
new_forms_dfoster

Selected Streams

Name
new_forms_dfoster

Choose Public if you want to share your Favorites with others

3. In the **Name** field, enter the name for the Stream Favorite.

4. By default, Stream Favorites are private -- that is, the Stream Favorite you create can be used only by you. If you want to be able to share your Stream Favorite with others, click the **Public** radio button.

5. Use the Stream Search to locate the streams in your depot that you want to mark as favorites:

   Note: By default, AccuRev searches the depot for any streams that contain the string you enter, but you can change that to matches, starts with, or ends with. The search field is case-sensitive -- searching on, say, *AccuRev* and *accurev* will yield different results.

6. When you have specified your search criteria, click the **Search** button.

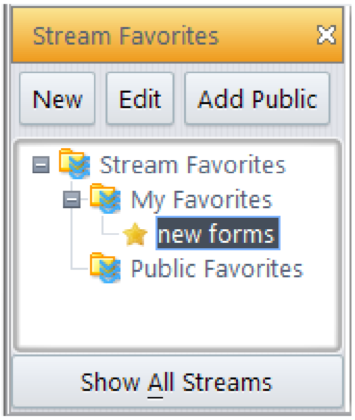
   AccuRev displays the streams that match your search criteria in the Search Results list box.

7. Use the **Add** and **Add All** buttons to choose the streams you want to mark as favorites.

Tip: You can use Shift+ and Ctrl+ click to select more than one stream at a time.

8. If you want to search for other streams to include in this Stream Favorite, return to *Step 5*.
9. If you change your mind about including a stream in this Stream Favorite, select it in the Selected Streams list box and click the **Remove** button.
10. When you have finished choosing streams, click **Ok** to create the Stream Favorite.

    The new Stream Favorite appears in the Stream Favorites panel under the My Favorites folder.



    Tip: Private Stream Favorites are displayed with a simple star icon (★). Public Stream Favorites are displayed with the star icon against a blue background (★).

11. To apply the Stream Favorite, double-click it. See *Applying a Stream Favorite* on page 162 for more information.

Working with Stream Favorites

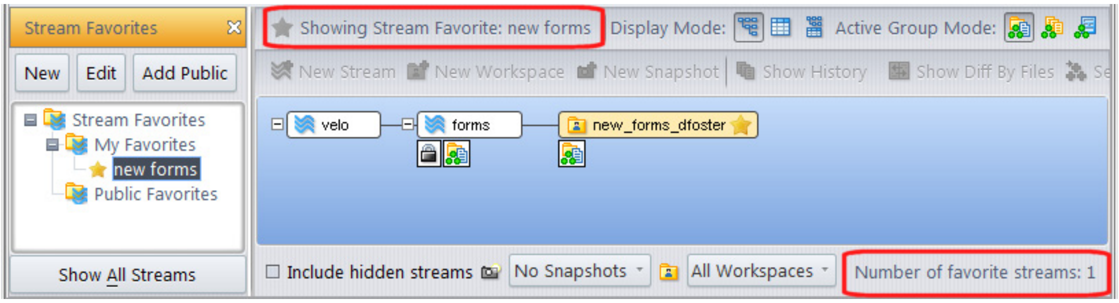
This section describes common procedures used with Stream Favorites.

Applying a Stream Favorite

To apply a Stream Favorite, double-click it.

In addition to changing the StreamBrowser display itself, AccuRev shows information about the current Stream Favorite in two places:

- The **Showing Stream Favorite** field at the top of the StreamBrowser identifies which Stream Favorite is active.
- The **Number of favorite streams** field at the bottom of the StreamBrowser indicates how many streams were specified as favorites in the current Stream Favorite.

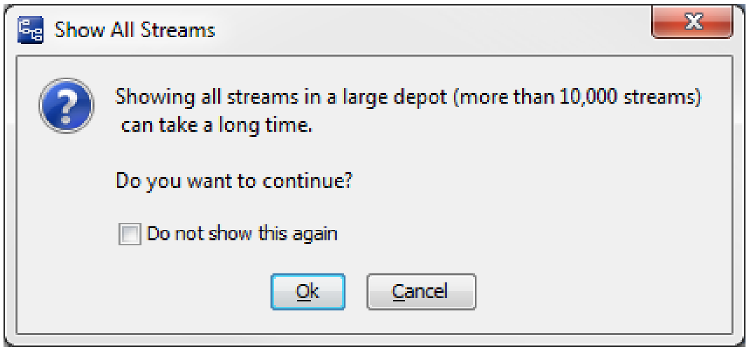

Showing Stream Favorite: new forms

Number of favorite streams: 1

Clearing a Stream Favorite from the StreamBrowser

To clear a Stream Favorite, click the **Show All Streams** button at the bottom of the Stream Favorites panel.

Depending on the size and complexity of your stream hierarchy, showing all streams can take a long time. AccuRev displays an informational message, which gives you the chance to cancel the operation.



Showing all streams in a large depot (more than 10,000 streams) can take a long time.

Do you want to continue?

□ Do not show this again

Tip: If you check the **Do not show this again** check box, you can restore this dialog box by clearing the **Skip "Show All Streams"** checkbox on the General tab of the AccuRev Preferences dialog box (**Tools** > **Preferences**).

Editing a Stream Favorite

You can make the following changes to a Stream Favorite:

- Name
- Public or private status
- The streams that make up the Stream Favorite

To edit a Stream Favorite:

1. In the Stream Favorites panel, right-click the favorite you want to change and choose **Edit** from the shortcut menu.

   The Edit Stream Favorite dialog box appears.

2. Change the Stream Favorite as needed.
3. Click **Ok**.

Deleting a Stream Favorite

When you delete a Stream Favorite that you created as a public Stream Favorite, it will no longer be available to any users who might have added it to their Stream Favorites panel.

To delete a Stream Favorite:

- In the Stream Favorites panel, right-click the Stream Favorite you want to delete and choose **Delete** from the shortcut menu.

Getting Others' Stream Favorites

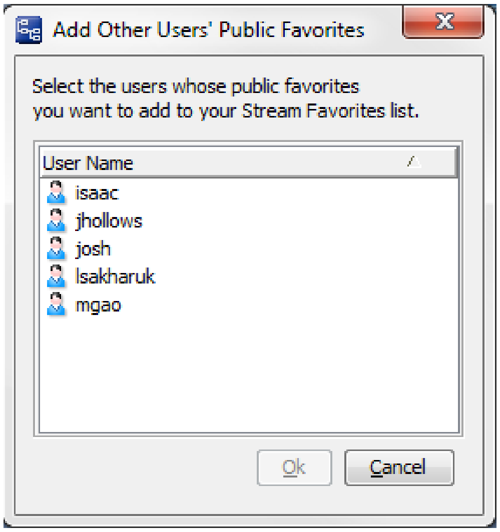
In addition to creating your own Stream Favorites, you can add Stream Favorites created by other users to your Stream Favorites panel. To be accessible to you, they must have been created as public, and not private, Stream Favorites. If the author of a public favorite you have added to your Stream Favorites panel later deletes that favorite, it will no longer appear on your Stream Favorites panel.

Note: When you add a user, you get all of that user's public favorites.

To add others' Stream Favorites to your Stream Favorites panel:

1. If the Stream Favorites panel is not already open in the StreamBrowser, click the Show Stream Favorites button (★).
2. In the Stream Favorites panel, click the **Add Public** button.

   The Add Other Users' Public Favorites dialog box appears.



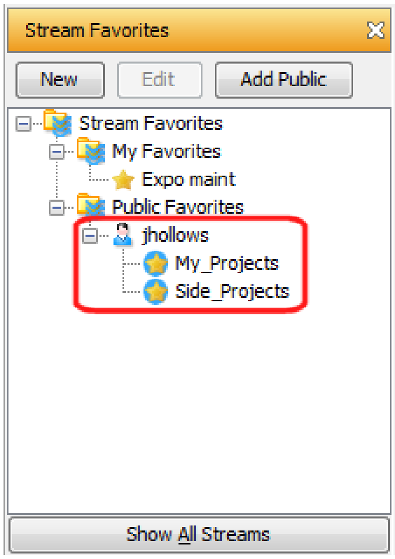
Select the users whose public favorites you want to add to your Stream Favorites list.

isaac
jhollows
josh
lsakharuk
mgao

3. Select one or more users from the list.

   Tip: Use Crtl+ and Shift+ click to select multiple users at the same time.
4. Click **Ok**.

   The users you selected appear in the Public Favorites folder.
5. To view other users' public Stream Favorites, click the + next to their name.



Removing Other Users's Stream Favorites

To remove another user's Stream Favorites from the Public Favorites folder in your Stream Favorites panel:

- In the Stream Favorites panel, right-click the user whose Stream Favorites you want to remove and choose **Remove** from the shortcut menu.

  The user, and *all* of that user's favorites, is removed from the Public Favorites list in your Stream Favorites panel.

The Overlaps Tab

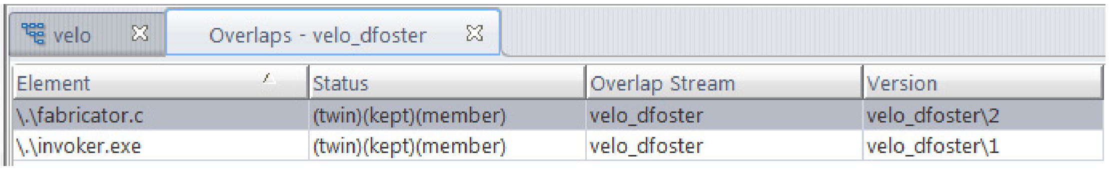
The Overlaps tab displays all versions with **(overlap)** status, both in the selected workspace and in the workspace's entire backing chain.

Opening an Overlaps Tab

Choose *Show All Overlaps* from the context menu of a workspace on a *Stream Browser* tab.

Overlaps Tab Layout

The Overlaps tab contains a table with these columns:



Element	Status	Overlap Stream	Version
\.\fabricator.c	(twin)(kept)(member)	velo_dfoster	velo_dfoster\2
\.\invoker.exe	(twin)(kept)(member)	velo_dfoster	velo_dfoster\1

Element (or separate "Name" and "In Folder" columns)

The pathname within the depot of the element with the overlap.

Status

The status of the version in that workspace or stream.

Overlap Stream

The workspace or stream that contains a version with **(overlap)** status.

Version

The version-ID of the version in that workspace or stream.

Working in an Overlaps Tab

For versions in dynamic streams (but not workspaces), you can begin the process of resolving the overlap:

1. Select one or more versions located in the same dynamic stream.
2. Right-click the selection, and choose *Send to Change Palette* from the context menu.

This opens a *Change Palette* tab, with an entry for each version you selected. The destination stream is automatically set to the parent of the stream where those versions are located. You can resolve the overlaps by performing merges on those versions.

The Remove Stream Command

(and the Remove Workspace Command)

You can invoke the *Remove* command on a stream, snapshot, or workspace. This hides the item from the *Stream Browser* and the *Workspaces* tab, but you can use the checkbox in both these tabs to view removed ("hidden") items.

This command does not remove the object from the depot — that would violate AccuRev's TimeSafe property. It disables some operations, but not all:

- You cannot work in a removed workspace.
- You can promote versions to a removed stream.

To bring back a removed object, make it visible with the *Include deactivated trees* checkbox, then invoke the *Reactivate* command on it.

Invoking the Remove Stream Command

In the *StreamBrowser*, choose *Remove* from the context menu of a stream or snapshot. Restrictions:

> This command is disabled if there are any streams, snapshots, or workspaces below the one to be removed.

Invoking the Remove Workspace Command

In the *StreamBrowser*, choose *Remove* from the context menu of a workspace. Restriction: You must be logged in as the user who owns the workspace

Default Group Subwindows in the StreamBrowser

Each workspace and dynamic stream has a **default group**, consisting of the elements that are currently active in that context. (More precisely, the default group contains a *version* of each active element.) In graphical mode, the StreamBrowser can display the default group for one workspace or stream at a time.

Opening a Default Group Subwindow

The StreamBrowser displays a ✚ Default Group icon for each workspace and dynamic stream with a non-empty default group. Click this icon to open a subwindow displaying the default group.

You can display default groups just one-at-a-time -- the subwindow closes automatically if you click the ✚ icon of another workspace or stream.

> **Note:** If you don't see any ✚ icons, but you know some elements are active...
>
> > A user preference (*Display Default Group in StreamBrowser*) controls whether the Stream Browser spends the time to determine the contents of the default group of each stream and workspace in its display.

Default Group Subwindow Layout

The default group subwindow has essentially the same layout as the File Browser's Outgoing Changes mode:

- A table, showing the members in the default group.
- A toolbar, for performing operations on the members.

Default Group vs. Outgoing Changes in a Workspace

A workspace's set of active elements (that is, its default group) is not necessarily the same as the set of elements that have changed in the workspace. In practice, the sets will often be the same. But in theory, the sets are overlapping, but not identical:

- The Default Group shows only those elements with a status of **(member)**. This status can be the result of "checking out" elements using Send to Workspace or Anchor commands, or simply the result of adding a new file to the workspace.
- The Outgoing Changes mode includes elements in the Default Group, but it can also show external objects and missing elements.

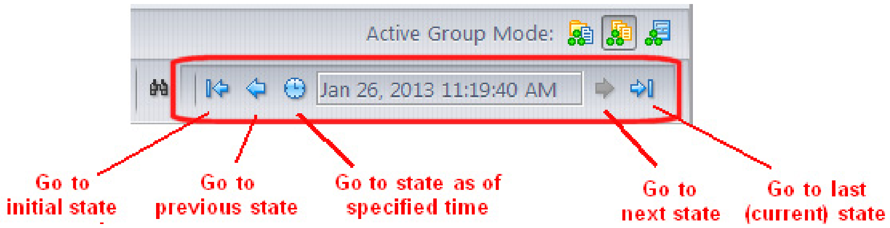
Working in a Default Group Subwindow

The toolbar in the Default Group subwindow contains many of the same buttons as the toolbar of the *File Browser*'s Details pane. In particular, you can select one or more elements, then promote them to the backing stream.

> **Note:** If you want to promote all the elements in the default group subwindow to the backing stream, you don't need to open the subwindow at all. Just drag-and-drop the active group icon ✤ that is displayed adjacent to the workspace or stream to the backing stream. (The backing stream is the only stream you can target with this drag-and-drop operation.)

StreamBrowser History

Like the individual files under version control, a depot’s stream hierarchy undergoes change, too. Initially, the depot contains a single base stream. Thereafter, additional dynamic streams are created, along with snapshots and user workspaces. Each such change in the stream hierarchy creates a new "state" of the hierarchy. You can browse through all these states, like browsing through a slide show, using the History controls in the StreamBrowser toolbar:



Find Stream

The StreamBrowser search feature enables you to find a stream or workspace whose name contains a specified character string. You can use this feature in any of the StreamBrowser's display modes: *graphical*, *tabular*, or combined.

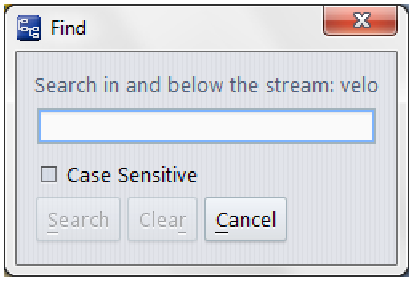
Invoking the Find Stream Dialog

You can access the StreamBrowser Find dialog in one of three ways:

- Click the Search icon ( ) in the StreamBrowser toolbar.
- Type **Ctrl-F** in a StreamBrowser view.
- Right-click over a stream and click **Find**.

Note that if you right-click, your search will be limited only to the streams under the stream where you right-clicked. Otherwise, ALL streams in the depot matching your search will be found.

Using the Find Dialog



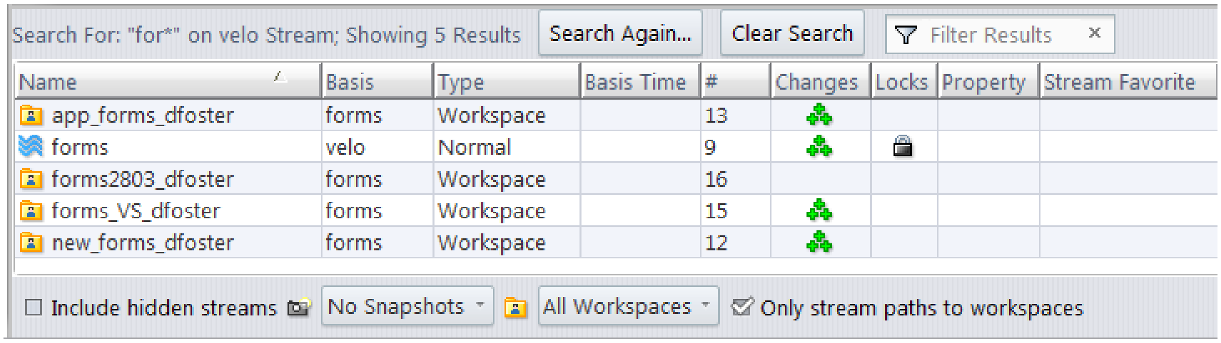
Search in and below the stream: velo

☐ Case Sensitive

Type characters in the text box that match all or part of the stream name(s) you are looking for. Make sure the *Case Sensitive* option is set as desired. You can enter any character that is legal for an AccuRev stream name. Wildcards ("*" and "?") are accepted. The find operation will match the string in any part of the name. So for example, performing a case-insenstive search for "Devel" will return streams with names like "devel_stream", "Old_Development".

Displaying and Filtering Stream Find Results

The results are displayed in a StreamBrowser tabular display.



Search For: "for*" on velo Stream; Showing 5 Results

Name	Basis	Type	Basis Time	#	Changes	Locks	Property	Stream Favorite
app_forms_dfoster	forms	Workspace		13	♣			
forms	velo	Normal		9	♣	🔒		
forms2803_dfoster	forms	Workspace		16				
forms_VS_dfoster	forms	Workspace		15	♣			
new_forms_dfoster	forms	Workspace		12	♣			

☐ Include hidden streams   No Snapshots   All Workspaces   ☑ Only stream paths to workspaces

To fine-tune the results, you can enter characters in the Filter Results field. As you type a string, the streams shown in the results display will be narrowed down to only those containing the characters you enter. The rules for Filter strings are the same as for Find String.

The Stream Diff (Files) Tab

(Show Diff By Files command)

The Stream Diff (Files) tab displays the element-by-element differences between two streams. Each stream can be a dynamic stream, a snapshot, or a workspace stream. AccuRev determines the differences by examining the elements' version-IDs in each stream. It does not compare the contents of file versions (but you can, using this tab's *Show Difference* command).

> **Note:** You can also perform a **Stream Diff by Issues** operation. See *The Stream Diff (Issues) Tab* on page 365

A stream comparison can report these differences

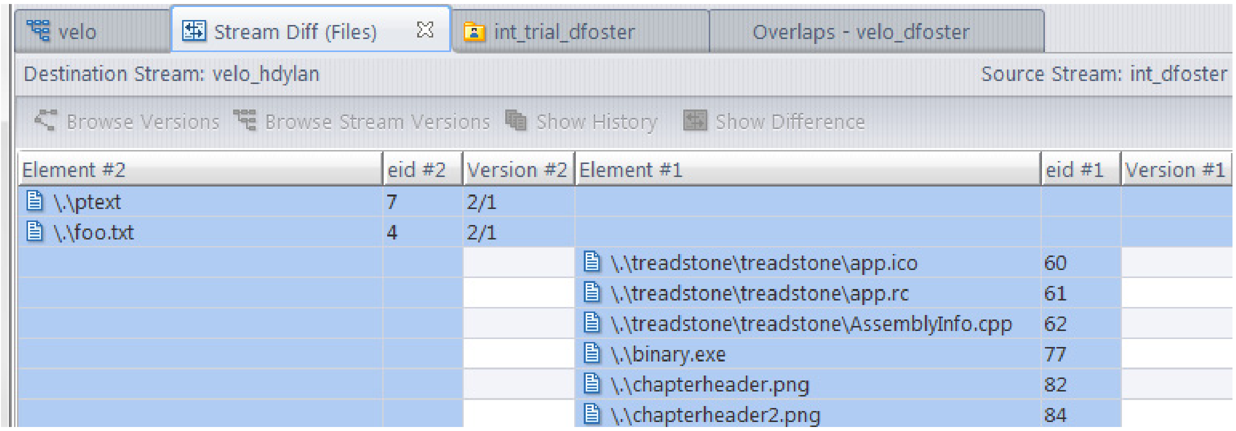
- A file element has been modified in one or both streams. (When analyzing the contents of a workspace, this command always uses the version currently in the workspace stream (which might be inherited from a higher-level stream). This version differs from the file in the workspace tree if the file has **(modified)** status.)
- An element has been renamed in one or both streams.
- A new element has been added to one of the workspaces (*Add to Depot* command) or streams (*Promote* command)
- An element has been removed from one of the streams (Defunct command).

Opening a Stream Diff (Files) Tab

1. In the StreamBrowser, select a stream, snapshot, or workspace.
2. Choose the *Show Diff by Files* command from the selection's context menu, or click the ⊞ button on the StreamBrowser's toolbar.
3. The mouse pointer changes to ⊞. Left-click another stream, snapshot, or workspace.

Stream Diff (Files) Tab Layout

The Stream Diff (Files) tab contains a table, each row of which details one element. Above the table is the name of the **Destination Stream** (the one you selected with the ⊞ mouse pointer) and **Source Stream** (the one you selected before invoking the command).



Destination Stream: velo_hdylan

Source Stream: int_dfoster

Element #2	eid #2	Version #2	Element #1	eid #1	Version #1
\.\ptext	7	2/1			
\.\foo.txt	4	2/1			
			\.\treadstone\treadstone\app.ico	60	
			\.\treadstone\treadstone\app.rc	61	
			\.\treadstone\treadstone\AssemblyInfo.cpp	62	
			\.\binary.exe	77	
			\.\chapterheader.png	82	
			\.\chapterheader2.png	84	

The columns of this table are:

Element #2 (or two columns: Name #2, In Folder #2)

The pathname of the same element in the destination stream. If empty, the element is present only in the source stream. Note that a *user preference* controls whether the element pathnames are displayed in a single column, or in two columns.

eid #2

The element-ID of the element.

Version #2

The version-ID of the destination stream's version of the element.

Element #1 (or two columns: Name #1, In Folder #1)

The pathname of the element in the source stream. If empty, the element is present only in the destination stream. Note that a *user preference* controls whether the element pathnames are displayed in a single column, or in two columns.

eid #1

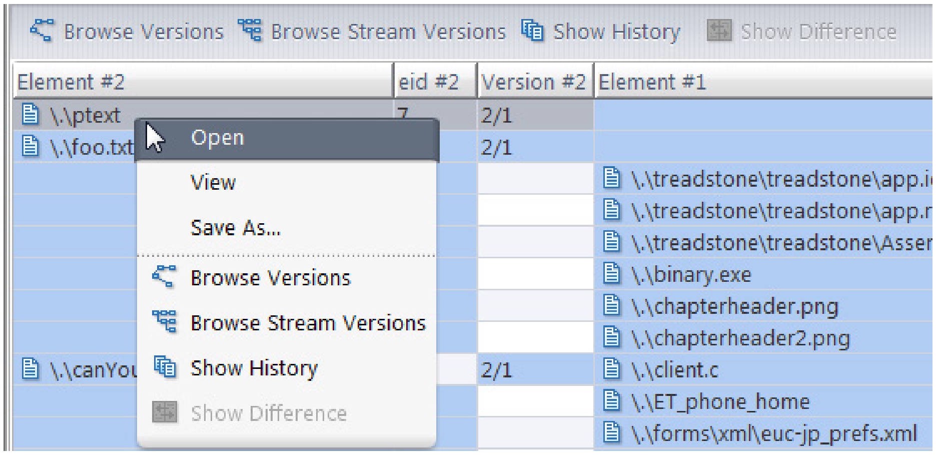
Same as eid #2. (The element-ID is immutable. It is unique within the depot, but not within the entire repository.)

Version #1

The version-ID of the source stream's version of the element.

Working in a Stream Diff (Files) Tab

In any row of the Stream Diff table, you can left-click to select an element's version in either stream/workspace:



Element #2	eid #2	Version #2	Element #1
\.\ptext	7	2/1	
\.\foo.txt		2/1	
			\.\treadstone\treadstone\app.i
			\.\treadstone\treadstone\app.r
			\.\treadstone\treadstone\Asser
			\.\binary.exe
			\.\chapterheader.png
			\.\chapterheader2.png
\.\canYou		2/1	\.\client.c
			\.\ET_phone_home
			\.\forms\xml\euc-jp_prefs.xml

Then, you can invoke a command, using the Stream Diff tab toolbar or the selection's context menu. Some commands operate on the particular version you selected; other commands operate on the overall element.

Commands That Operate on the Selected Version

Open (equivalent to double-click)

Windows: Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

UNIX: Open a text editor on the file.

View

(text files only) Open a text editor on a temporary copy of the selected version.

Save As

Copy the selected version to another filename.

Commands That Operate on the Overall Element

Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

Browse Stream Versions

Open a *Stream Version Browser* tab for the selected element. This tab displays the depot's stream hierarchy, much like the *Stream Browser*. But on this tab, each stream represents the *version* of the specified element that appears in that stream.

Show History

Open a *History Browser* tab, containing the transactions involving the selected element.

Show Difference

(file elements only) Compares the contents of the two streams' versions of the selected element. Note that when analyzing the contents of a workspace, this command always uses the version currently in the workspace stream (which might be inherited from a higher-level stream). This version differs from the file in the workspace tree if the file has **(modified)** status.

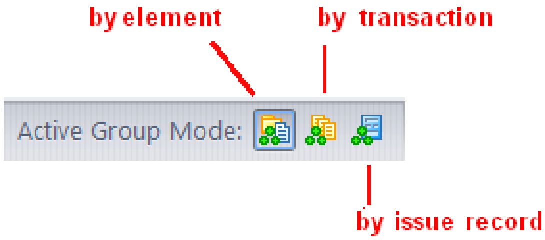
The Active Issues Tab

The **Active Issues** tab (also referred to as "**Stream Issues**") displays the issue records that are in a particular stream, as well as the entries in a selected issue record's change package.

Because the majority of operations in the tab concern change packages, they are covered in that chapter. See *The Active Issues Tab (Show Active Issues)* on page 356

The Active Transactions Tab (Show Active Transactions)

When the *Stream Browser* is displaying element activity by transaction (not by element or by issue record), the *Show Active Transactions* command opens an Active Transactions tab, containing the transactions that created the versions that are currently active in a particular stream or workspace.



It may be the case that some, but not all, of the transaction's versions are currently active; the other versions may have already been promoted to the parent stream.

Notes:

- **Multiple transactions for the same element --** If several Promote transactions created successive versions of the same element in a stream, and that element has not been promoted to its parent (or purged with **Revert to Basis**), then all of those Promote transactions will appear in the Active Transactions listing. That is, a transaction is listed if the changes that it sent *to* the stream have not yet been promoted (or purged) out of the stream.
- **"Checkout" and "Revert to Basis" transactions not included --** Transactions whose value in the Action column is co ("checkout") or purge are not included in the Active Transactions tab. Such transactions are created by the ***Send to Workspace***, ***Anchor***, and ***Revert to Basis*** commands.
- **Comparison with *Show History* command --** The set of transactions displayed by Show Active Transactions is a subset of the transactions displayed by Show History. You can think of Show Active Transactions as displaying a stream's "current history", rather than its "ancient history".

Opening an Active Transactions Tab

In the *StreamBrowser*, make sure development activity is being displayed by transaction (button at right side of StreamBrowser toolbar). Choose *Show Active Transactions* from the context menu of a stream or workspace.

> **Note:** You can't invoke this command on a depot's base stream. The "active" concept does not apply to the versions in this stream, because there is no higher-level stream to promote versions to, and to inherit versions from.]

Active Transactions Tab Layout

The layout of the Active Transactions tab is virtually identical to that of the *History Browser*. Exception:

> The Versions pane includes an extra column, "Status". The **(member)** flag indicates that the element is currently active in the workspace or stream. The lack of this flag indicates that the element is no longer active (because it has been processed with the *Promote* command or the *Revert to Basis* command). This column also includes the **(overlap)** flag, if appropriate.

Working in a Active Transactions Tab

The sets of commands available in the Summary pane (upper) and Versions pane (lower) of the Active Transactions tab are virtually identical to those available in the *History Browser*. The differences are:

Promote

This command is enabled only in the Active Transactions tab, not in other History Browser contexts. In the Summary pane, it promotes the transaction's versions to the parent stream.

Note: Some of the transaction's versions may no longer be active, having been promoted or purged (**Revert to Basis** command). Invoking Promote operates on the versions that are still active.

In the Versions pane, it promotes the selected version (or set of versions) to the parent stream.

Revert

The *Revert Transaction* command is enabled in other History Browser contexts, but not in the Active Transactions tab.

Show Current Change Package

By default, the lower pane of the Active Transactions tab displays change package details for the current time. Use the **show current change package** checkbox to display the state of the change package at the time of the selected transaction. For more information, see *A Note about the Active Transactions Tab* on page 364.

The Stream Version Browser

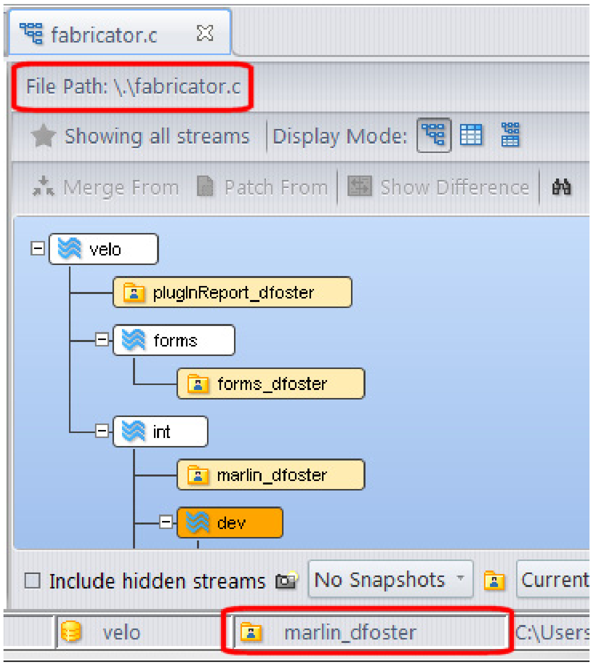
The Stream Version Browser tool is a variant of the *Stream Browser*. It displays the stream hierarchy for an entire depot; each stream, snapshot, and workspace in the display represents a version of a specified element.

Opening a Stream Version Browser Tab

The Stream Version Browser displays information about one particular element. In a *File Browser*, select a file or directory element in the Details pane. Then click the *Browse Stream Versions* toolbar button.

Stream Version Browser Tab Layout

The Stream Version Browser display is almost exactly the same as the *StreamBrowser* display. But the pathname of the specified element is displayed in the upper left corner of the tab. Also, the workspace that you were in (when you invoked the Stream Version Browser from in the File Browser) is displayed at the bottom of the GUI window.


File Path: \.\fabricator.c

Each box in the Stream Version Browser display represents the version of the specified element that is currently in that stream, snapshot, or workspace. A workspace box indicates the version in the workspace stream, which is not necessarily the same as the file in the workspace tree. They differ if the file has **(modified)** status.

Working in a Stream Version Browser Tab

The Stream Version Browser has many of the same commands as the *StreamBrowser* for controlling the display. These commands, available in the Stream Version Browser toolbar, include:

Search for Stream

Find the stream(s) whose name includes a specified character string. A *Search for Stream* dialog appears.

Graphical Display

Display the depot's stream hierarchy as a tree

Tabular Display

Display the depot's stream hierarchy as a table

Graphical/Tabular Display

Display the depot's stream hierarchy as a tree in one pane, and as a table in another pane.

Similarly, the Stream Version Browser includes many of the same commands as the *Version Browser*, for operating on a selected version of an element. These include:

Open

Run the appropriate command on the version, according to its file type.

View

Open a text editor on a temporary copy of the currently selected version (text files only).

Save As

Copy the currently selected version to another filename.

Merge From

Merge the selected version into the version in the workspace from which the Stream Version Browser was invoked.

Patch From

Patch the selected version into the version in the workspace from which the Stream Version Browser was invoked.

Send To Workspace

Activate the selected element in the workspace from which the Stream Version Browser was invoked.

Send To Issue

Record the selected version in the change package section (Changes tab) of one or more issue records. The default query is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected version(s) will be sent.

Show Difference

Compare the selected version with another version of the element. AccuRev changes the mouse pointer, to prompt you to select the other version.

Properties

Displays a message box containing information about the selected version.

The Patch List Tab
(Show Patch List command)

The Patch List tab lists all the individual versions whose changes need to be sent from one stream (or snapshot, or workspace) to another, so that the second stream will include all the changes in the first stream. (See *Patch and Patch From* on page 227 for more information about patches.)

Notes:

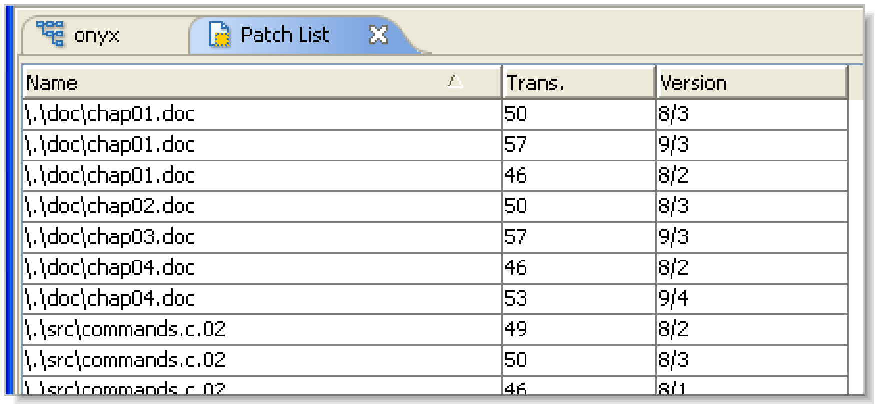
- *Comparison with Stream Diff*—You can think of the Patch List tab as providing an "exploded view" of the Stream Diff (Files) tab. Whereas Stream Diff provides a simple "before" and "after" perspective on how each element changed, Patch List shows all the individual steps (that is, the versions) in progressing from the "before" state to the "after" state.
- *Patch List and the Patch command*—A Patch List table with 24 entries does not mean that you would need 24 invocations of the Patch command to propagate all the changes from one stream to the other. Each invocation of Patch incorporates all the changes in the specified patch ,which can include multiple versions. (Prior to AccuRev 3.5.5, each invocation of the Patch command did incorporate the changes in one version, so the correspondence to the Patch List table was exact.)

Opening a Patch List Tab

1. In the *Stream Browser*, select a stream, snapshot, or workspace.
2. Choose the *Show Patch List* command from the selection's context menu, or click the button on the StreamBrowser's toolbar.
3. The mouse pointer changes to . Left-click another stream, snapshot, or workspace.

Patch List Tab Layout

The Patch List tab contains a table, each row of which details one version.



Name	Trans.	Version
\.\doc\chap01.doc	50	8/3
\.\doc\chap01.doc	57	9/3
\.\doc\chap01.doc	46	8/2
\.\doc\chap02.doc	50	8/3
\.\doc\chap03.doc	57	9/3
\.\doc\chap04.doc	46	8/2
\.\doc\chap04.doc	53	9/4
\.\src\commands.c.02	49	8/2
\.\src\commands.c.02	50	8/3
\.\src\commands.c.02	46	8/1

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.
Note: This is the element's pathname in the first stream at the current time. This may not match its pathname at the time of the transaction listed in the Transaction column.

Transaction

The transaction in which this version was created.

Version

The version-ID of the version

Working in a Patch List Tab

The Patch List tab display is informational only. See also:*Working with Tables* on page 14.

5. The History Browser

The History Browser displays some or all of the transactions associated with a particular element, stream, or depot. In addition to viewing information about a transaction and the versions involved in it, you can perform operations on the individual versions (*View*, *Diff*, etc.) and on the transaction's set of versions (*Promote*, *Revert*).

Opening a History Browser Tab

There are several ways to launch the History Browser:

- *To view the transactions involving an individual element*

  You can have the History Browser display the transactions in which one particular element was involved. (Other elements may have been involved in these transactions, too.) Elements are listed in various places in the GUI:

  - In the Details pane of a File Browser tab
  - In the Default Group Contents subwindow of a StreamBrowser
  - In an issue record's change package (Changes subtab)

  In any of these contexts, you can select an element and click the *Show History* toolbar button. Alternatively, right-click an element and select *History* > *Show History* from the context menu.

- *To view the transactions involving a particular stream*

  You can have the History Browser display transactions that affected a particular stream. For a workspace stream, this principally includes the *keep* transactions that create real versions in the stream. It can also include *co*, *move*, *defunct*, and *undefunct* transactions. For a dynamic stream, this includes *promote* transactions *to* the stream, but not promotions *from* the stream.

  In the StreamBrowser, right-click a stream and select *Show History* from the context menu.

- *To view the active transactions for a particular stream*

  Every stream has a default group, consisting of the elements that are active in that stream. In the Stream Browser, a stream's context menu includes the command *Show Active Transactions*; it opens a History Browser tab and loads the set of transactions (*Keep*, *Promote*, etc.) in which the stream's active versions were created.

- *To view the transactions involving a particular depot*

  You can view the transactions for an entire depot -- all the elements, in all the streams. Use the *Admin* > *Depots* command to list all of the repository's depots. Then, select a depot and choose *Show History* from its context menu, or from the *Actions* group in the main menu.

- *To view a single transaction or a specified group of transactions*

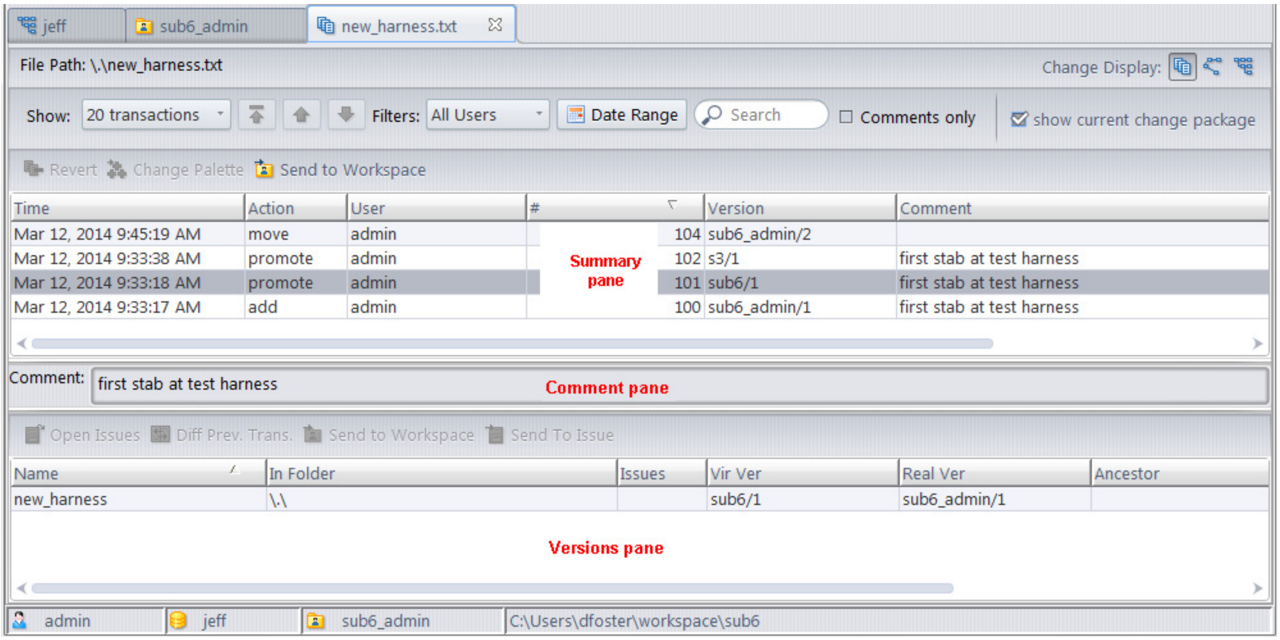
  In the Version Browser, selecting *Show History* from any version's context menu displays the transaction in which that particular version was created.

If you have used AccuWork to promote files, an issue record's *affectedFiles* field contains a list of transaction numbers. Click the *Show History* button next to this field to display just those listed transactions.

History Browser Tab Layout

The History Browser appears in a separate tab of the AccuRev GUI's multiple-tab display. The tab is divided into three panes:

- The *Summary pane* displays a group of transactions, one per line. This pane displays overall information: transaction number, timestamp, transaction type, etc.
- The *Comment pane* shows the comment string, if any, that was specified for the currently-selected transaction.
- The *Versions pane* shows all the versions that were involved in the currently-selected transaction. It also indicates which change packages, if any, those versions belong to (Issues column).



File Path: \.\new_harness.txt

Time	Action	User	#	Version	Comment
Mar 12, 2014 9:45:19 AM	move	admin	104	sub6_admin/2	
Mar 12, 2014 9:33:38 AM	promote	admin	102	s3/1	first stab at test harness
Mar 12, 2014 9:33:18 AM	promote	admin	101	sub6/1	first stab at test harness
Mar 12, 2014 9:33:17 AM	add	admin	100	sub6_admin/1	first stab at test harness

Comment: first stab at test harness

Name	In Folder	Issues	Vir Ver	Real Ver	Ancestor
new_harness	\.\		sub6/1	sub6_admin/1	

C:\Users\dfoster\workspace\sub6

The Summary Pane

The summary pane displays a table containing a set of transactions, one per row. Each row of the summary pane's table displays the following information about an individual transaction:

Time

A timestamp, indicating when the transaction took place.

Action

The kind of transaction: *keep*, *promote*, etc.

User

The *principal-name* of the user who initiated the transaction.

# (transaction number)

The unique number (within this depot) of the transaction.

Version

(only for transactions involving an individual element) The real version or virtual version of the element that was created in this transaction. This column doesn't appear in displays of a stream's history or an entire depot's history; the version(s) created by the transaction appear in the Versions pane.

Comment

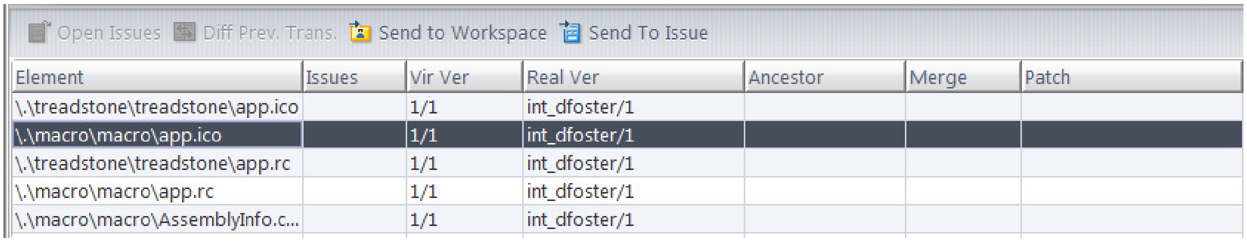
The first line of the user-supplied comment for this transaction. If the comment extends to additional lines, an ellipsis ("dot dot dot") appears here. For the full text of the comment, look in the Comment pane.

The Comment Pane

The Comment pane displays the full text of the user-supplied comment for the transaction currently selected in the Summary pane.

The Versions Pane

The Versions pane displays a table of all the versions, if any, created by the currently selected (highlighted) transaction in the Summary pane.



Element	Issues	Vir Ver	Real Ver	Ancestor	Merge	Patch
\.\treadstone\treadstone\app.ico		1/1	int_dfoster/1			
\.\macro\macro\app.ico		1/1	int_dfoster/1			
\.\treadstone\treadstone\app.rc		1/1	int_dfoster/1			
\.\macro\macro\app.rc		1/1	int_dfoster/1			
\.\macro\macro\AssemblyInfo.c...		1/1	int_dfoster/1			

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

Virtual Version

Real Version

The two version-IDs of this version. The real version-ID indicates the workspace stream where the version was originally created. The virtual version-ID is different if this transaction records:

- a promotion of the version to a dynamic stream *(Promote* command), or
- a check-out of the version to a workspace *(Send to Workspace* or *Anchor* command)

See *Real Versions and Virtual Versions* on page 136.

Ancestor

(transactions in a workspace stream only) The predecessor version, which this version was derived from.

Merge

(**keep** transaction recorded by the *Merge* command only) The "from" version to the merge operation that created this version.

Patch

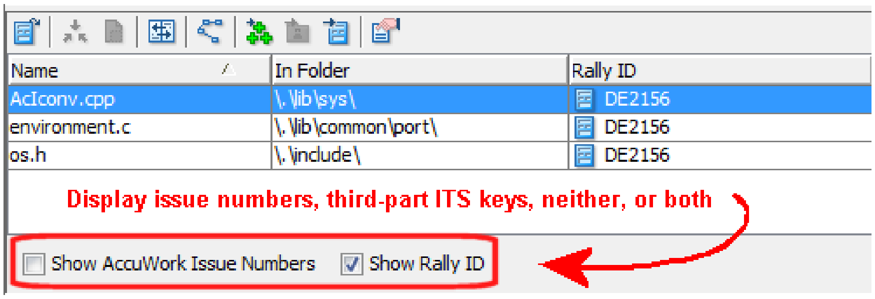
(**keep** transaction recorded by the *Patch* command only) The "from" version to the patch operation that created this version.

Issues

The AccuWork issue records (change packages) to which this version has been added. The value in this field is influenced by the **show current change package** checkbox. See *Show Current Change Package Checkbox* on page 183 for more information.

Features for Third-Party Issue Tracking System (ITS)

If you have enabled AccuRev to work with a third-party ITS, the Versions Pane displays controls that let you display AccuWork issue numbers, third-party ITS issue keys, neither, or both.



Name	In Folder	Rally ID
AcIconv.cpp	\.\lib\sys\	DE2156
environment.c	\.\lib\common\port\	DE2156
os.h	\.\include\	DE2156

Display issue numbers, third-part ITS keys, neither, or both

□ Show AccuWork Issue Numbers ☑ Show Rally ID

The first time you display the History Browser, AccuRev displays an informational message alerting you to this feature.

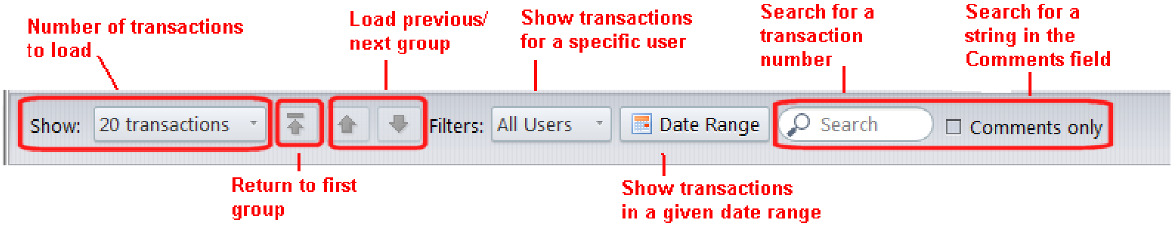
Working in the Summary Pane

There's potentially a large number of transactions to display in the Summary pane -- for example, when displaying the history of an entire depot. Retrieving all the transactions at once from the depot's database can be time-consuming. The History Browser handles this situation by initially retrieving a small number of transactions, then letting you control the retrieval of additional transactions with the toolbar located just above this table.

Controls at the top of the Summary pane determine which transactions are displayed. You can invoke commands on one or more transactions that you select. See *Working with Tables* on page 14.

Summary Pane Controls

The controls pictured below determine how many transactions (and which ones) are retrieved from the depot's database and loaded into the summary pane. Click on a control in the image below for more information about it:



Number of transactions to load

Load previous/ next group

Show transactions for a specific user

Search for a transaction number

Search for a string in the Comments field

Show: 20 transactions

Filters: All Users

Date Range

Search

□ Comments only

Return to first group

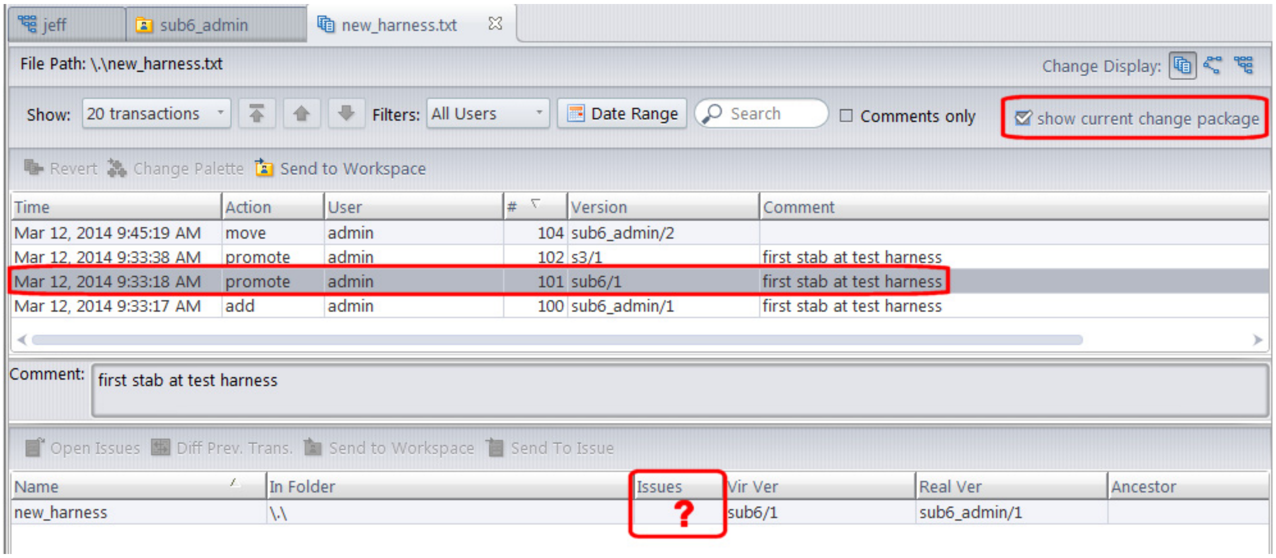
Show transactions in a given date range

Once a certain set of transactions has been loaded into the summary pane, you can browse through those transactions. Click any transaction to select it; the Comment and Versions panes are updated with the transaction's details. You can use the usual navigation keys to change the selected transaction: up-arrow and down-arrow, *PgUp* and *PgDn*, *Ctrl-PgUp* and *Ctrl-PgDn*.

Show Current Change Package Checkbox

The preceding controls affect the contents of the Summary pane --what transactions are displayed, how many, in what order, and so on. The **show current change package** check box, which appears on the far right side of the display controls, can affect the information that is displayed in the Versions pane.

When the **show current change package** checkbox is selected (the default setting), the Versions pane reflects the change package contents at the *current* time. In the following illustration, you can see that the version created with transaction #101 does not appear to be associated with a change package:



File Path: \.\new_harness.txt

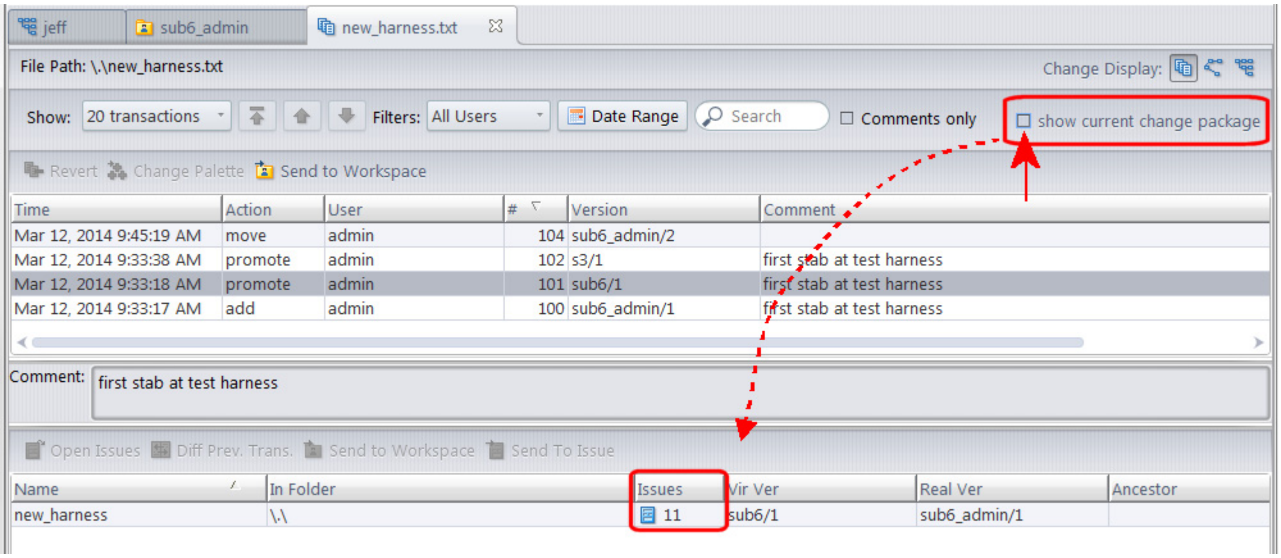
Show: 20 transactions Filters: All Users □ Comments only ☑ show current change package

Time	Action	User	#	Version	Comment
Mar 12, 2014 9:45:19 AM	move	admin	104	sub6_admin/2	
Mar 12, 2014 9:33:38 AM	promote	admin	102	s3/1	first stab at test harness
Mar 12, 2014 9:33:18 AM	promote	admin	101	sub6/1	first stab at test harness
Mar 12, 2014 9:33:17 AM	add	admin	100	sub6_admin/1	first stab at test harness

Comment: first stab at test harness

Name	In Folder	Issues	Vir Ver	Real Ver	Ancestor
new_harness	\.\	?	sub6/1	sub6_admin/1	

If you are inspecting a version and do not see an associated change package where you expect, it might indicate that the version was *removed* from the change package after it was promoted --that is, it is not currently associated with the change package. Clearing the **show current change package** check box will show you the associated change package association as it was at the time the version was created:



File Path: \.\new_harness.txt

Show: 20 transactions Filters: All Users

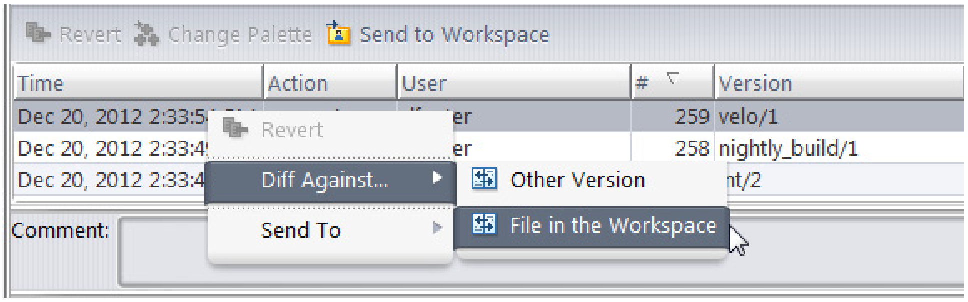
Time	Action	User	#	Version	Comment
Mar 12, 2014 9:45:19 AM	move	admin	104	sub6_admin/2	
Mar 12, 2014 9:33:38 AM	promote	admin	102	s3/1	first stab at test harness
Mar 12, 2014 9:33:18 AM	promote	admin	101	sub6/1	first stab at test harness
Mar 12, 2014 9:33:17 AM	add	admin	100	sub6_admin/1	first stab at test harness

Comment: first stab at test harness

Name	In Folder	Issues	Vir Ver	Real Ver	Ancestor
new_harness	\.\	11	sub6/1	sub6_admin/1	

Summary Pane Commands

You can select one or more transactions in the Summary pane, and invoke commands on them. Use the selection's context menu or the toolbar buttons above the Summary pane.



Time	Action	User	#	Version
Dec 20, 2012 2:33:5		er	259	velo/1
Dec 20, 2012 2:33:4		er	258	nightly_build/1
Dec 20, 2012 2:33:4				nt/2

Comment:

Revert to Basis

(from dynamic stream history only) For the selected *Promote* transaction, performs the equivalent of an "undo" in a particular child workspace of the dynamic stream. AccuRev prompts you to specify which workspace. Older versions of the transaction's elements will be activated in that workspace. For details, see the *revert* reference page in the *AccuRev CLI User's Guide*.

Diff Against Other Version

(from text-file element history only) Compare the version in the selected transaction with the version in another transaction. AccuRev prompts you to select another transaction from the summary pane.

Diff Against File in the Workspace

(from text-file element history only, when in the context of a particular workspace) Compare the version in the selected transaction with the file in the workspace from which the History Browser was invoked. This workspace is listed at the bottom on the AccuRev GUI window. This enables you to invoke a comparison with a file that you have changed in your workspace, but have not yet preserved in the repository with a *Keep* command.

This command is disabled if you invoked the History Browser from a non-workspace context.

Send to Change Palette

(dynamic stream only) Load the selected versions into the Change Palette, so that they can be merged and promoted to other streams.

Send to Change Palette (Changes)

(dynamic stream only) Load the selected versions into the Change Palette, so that they can be patched to other streams.

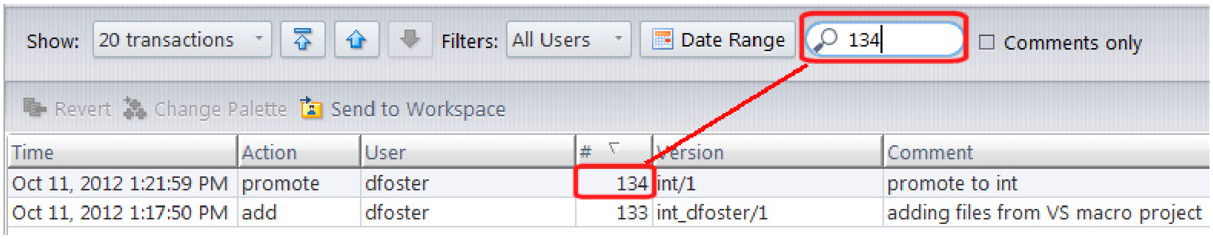
Send to Workspace

Activate each of the transaction's versions in a particular workspace. AccuRev prompts you to specify one of your workspaces.

History Browser / Summary Pane Controls

Go to a Particular Transaction

Entering a transaction number in the Search field unloads the current set of transactions, then loads the set beginning with the specified number.



Time	Action	User	#	Version	Comment
Oct 11, 2012 1:21:59 PM	promote	dfoster	134	int/1	promote to int
Oct 11, 2012 1:17:50 PM	add	dfoster	133	int_dfoster/1	adding files from VS macro project

If that particular transaction is not relevant -- e.g. it did not involve the element whose history you're viewing -- the set begins with the next older relevant transaction.

History Browser / Summary Pane Controls

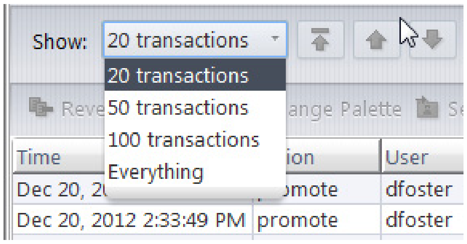
Display Transactions with a Particular Comment

By default, the Summary pane displays transactions no matter what comments they contain (if any). To filter the display based on transactions' comment fields, type a search string in the Search field, then select the Comments only checkbox to enable the filtering.

History Browser / Summary Pane Controls

Manage Groups of Transactions

The Summary pane table initially contains the 10 most recent relevant transactions.



Time	ion	User
Dec 20, 2	omote	dfoster
Dec 20, 2012 2:33:49 PM	promote	dfoster

To change this count, use the Transaction Count list box. (Each time you change the count, the browser returns to displaying the most recent transactions.) Selecting *Everything* loads all the relevant transactions into the summary pane.

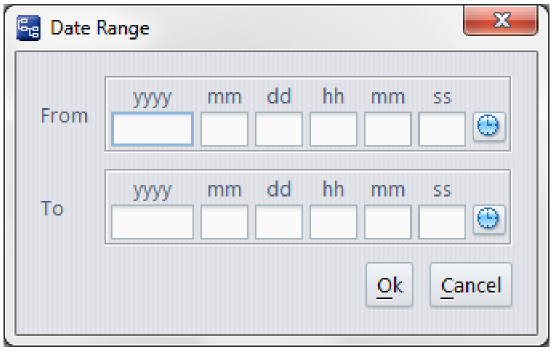
Several controls on the toolbar enable you to retrieve additional transactions, which have not yet been loaded into the summary pane:

- The ⇩ button unloads the current set of transactions, then loads the next older set of transactions.
- The ⇧ button unloads the current set of transactions, then loads the next newer set of transactions. This button is enabled only when you're viewing the history of an entire depot.
- The ⤒ button unloads the current set of transactions, then loads the first (newest) set of transactions.

History Browser / Summary Pane Controls

Display Transactions Created in a Particular Interval

By default, the Summary pane displays transactions no matter when they were created. Use the 🕒 *Set Date Interval* toolbar button to restrict the display to transactions that occurred in a specified interval.

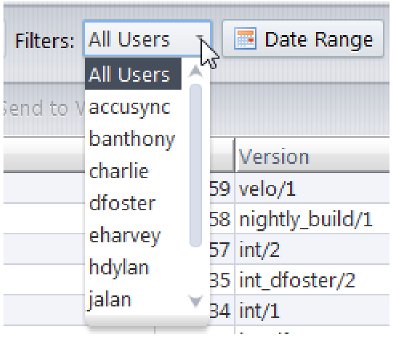


This changes the setting in the Transaction Count list box to *Custom* . To cancel the date interval restriction, change this list box setting back to *10 Transactions* (or something else).

History Browser / Summary Pane Controls

Display a Particular User's Transactions

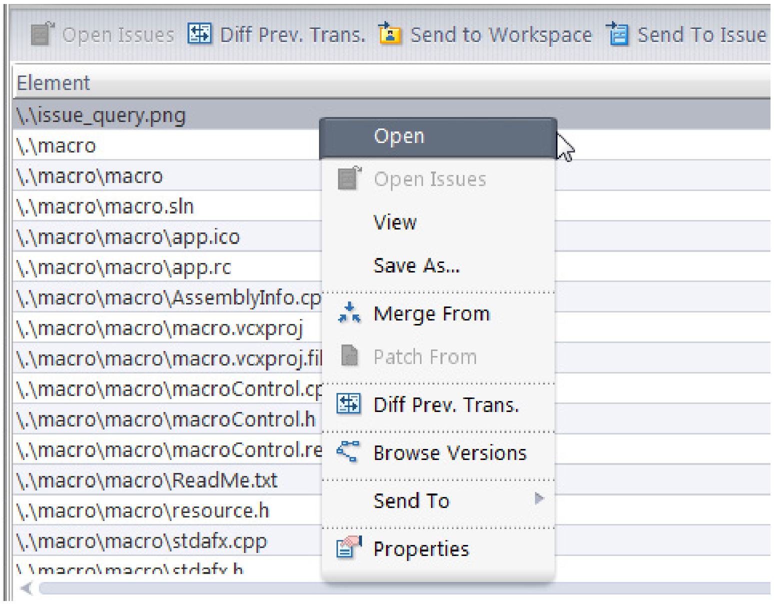
By default, the Summary pane displays transactions created by all users.



To restrict the display to one user's transactions, use the User Filter list box. Select **For All Users** to turn off by-user filtering.

Working in the Versions Pane

You can perform several operations on a selected version in the Versions pane, using its context menu or the versions pane toolbar.



Element

\.\issue_query.png
\.\macro
\.\macro\macro
\.\macro\macro.sln
\.\macro\macro\app.ico
\.\macro\macro\app.rc
\.\macro\macro\AssemblyInfo.cp
\.\macro\macro\macro.vcxproj
\.\macro\macro\macro.vcxproj.fil
\.\macro\macro\macroControl.cp
\.\macro\macro\macroControl.h
\.\macro\macro\macroControl.re
\.\macro\macro\ReadMe.txt
\.\macro\macro\resource.h
\.\macro\macro\stdafx.cpp
\.\macro\macro\stdafx.h

Open (equivalent to double-click)

Windows: Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

UNIX: Open a text editor on the file.

Open Issues

Displays the change package(s) -- that is, the issue record(s) -- to which the version belongs.
Note: A version is "in" a change package if it -- or any of its descendants -- is the head version of an entry in that change package.

For a version that belongs to a single change package, an edit form opens on that one issue record. For a version that belongs to multiple change packages, an Issues tab opens, listing all the issue records.

View

Open a text editor on a temporary copy of the currently selected file (text files only).

Save As

Copy the currently selected file to another filename.

Merge From

(only when displaying the history of an individual file) Merge the selected version into the version in the workspace from which the History Browser was invoked.

Patch From

(only when displaying the history of an individual file) Patch the selected version into the version in the workspace from which the History Browser was invoked.

Diff Against Previous Transaction

Compare the selected version with the version that was in its workspace or stream just before the version was created.

Promote

(only available in the *Active Transactions* tab, not in other History Browser contexts)

Browse Versions

Open a Version Browser on the element whose version you selected.

Send to Change Palette

(dynamic stream only): Load the selected versions into the Change Palette, so that they can be promoted to other streams.

Send to Workspace

(only when displaying the history of an individual file) Activate the selected element in the workspace from which the History Browser was invoked.

Send to Issue

Create a new entry (or modify an existing entry) in the change package section (Changes tab) of one or more issue records. The version you specify becomes the head version of the change package entry; AccuRev determines the corresponding basis version automatically.

The default query is executed, and the results are displayed in a dialog (see *Send to Issue : Associating Changes with an Issue* on page 354). You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

Properties

Displays information about the selected element. The data items displayed vary with the type of element.

6. The Version Browser

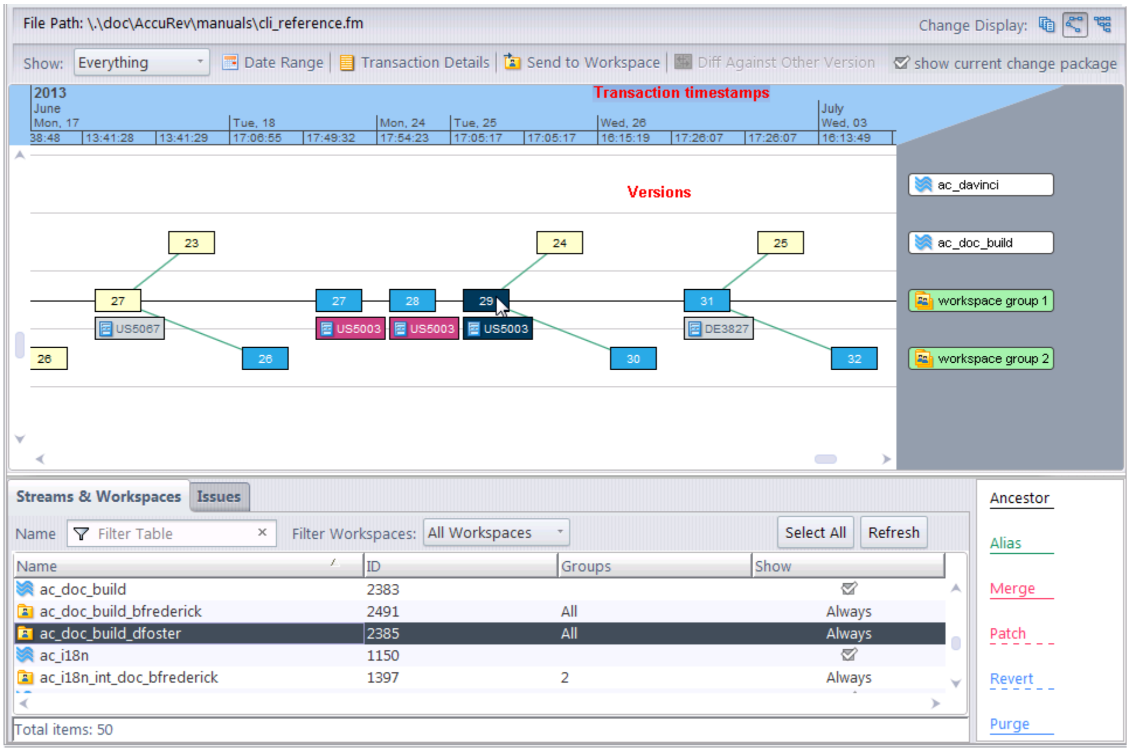
AccuRev maintains complete ancestry information for each element. The Version Browser displays some or all of an element's versions, using color-coded lines to indicate the way in which each version was created. You can perform various version-related operations, such as comparing any two versions (text files only) and copying any version to your workspace.

Opening a Version Browser

The names of elements appear in many contexts in the AccuRev GUI. These include the Details pane of the File Browser, the versions pane of the History Browser, and the Changes tab (change package) of a AccuWork issue record. In any of these contexts, you can select one element and invoke **Browse Versions**. In some cases, this option is a sub-choice. For instance, one of the most common ways to invoke the Version Browser is to right-click an element name and then click **History -> Browse Versions**.

Version Browser Layout

The Version Browser display consists of two panes. The upper pane provides a graphical summary of element versions, where they exist in the stream and workspace structure, and the relationships between them. The lower pane provides a list of the involved streams, workspaces, and issues. (**Note:** Some screen shots may not reflect the latest Version Browser color scheme.)
Selections made in one pane are reflected in the other. Default behavior of the Version Browser is set in the **Tool -> Preferences** dialog box (see *AccuRev Preferences (Tools > Preferences Command)* on page 39).


File Path: \.\doc\AccuRev\manuals\cli_reference.fm

Show: Everything

Total items: 50

Name	ID	Groups	Show
ac_doc_build	2383		☑
ac_doc_build_bfrederick	2491	All	Always
ac_doc_build_dfoster	2385	All	Always
ac_i18n	1150		☑
ac_i18n_int_doc_bfrederick	1397	2	Always

Upper (Graphical) Pane

The borders of the graphical pane provide information or viewing functionality:

- The top border provides a time line for each transaction. Looking to see who created a version on a particular day? Adjust the time span as described below and scroll to the date of interest.
- The toolbar above the time line provides a variety of controls: use **Show:** to select from a predefined number of transactions to display in the Version Browser. This will adjust the time line depending on how long it took for that many transactions to occur -- an active element that gets changed often might have multiple transactions in a single day, while a relatively static element might have only a few transactions over the course of multiple years. Alternatively, use the **Date Range** control to display everything between specific dates. The remaining controls provide information about selected versions. Mouse over the controls to display tool tip information about their purpose.
- The right border lists the streams and workspaces where versions reside. White labels indicate dynamic streams; light yellow indicate individual workspaces. Gold labels represent *workspace groups*, which make the version browser easier to view and navigate. To view the workspaces that belong to workspace group, either mouseover the group, or view individual workspaces in the tabular pane (see below) and examine any entries under the Groups column.
- The left and bottom borders contain scroll bars for navigating within the graphical pane. (You can also click and drag within the graphical pane to adjust the view.)

The main body of the graphical view displays the versions of the current element, the stream(s) where they exist, the AccuWork issue(s) related to the version, and the relationship between a version and its immediate predecessor.

- Each version is represented by a yellow box. The number in the box is the version number of the element in that particular stream at that particular time (as represented in the timeline along the upper border).
- The yellow box with a blue outline is the version in the workspace or stream from which you launched the Version Browser.
- Workspace versions associated with an issue have a box underneath them showing the issue number. If the version is associated with more than one issue, or if the issue number is too long to display, then the issue number will end with an ellipsis (...). The number that is displayed is either an AccuWork number or a Third Party ITS number, depending on how your installation is configured. To open the issue, double-click it, or right-click for the context menu.
- Issue numbers with a period and a number appended indicate issue *variants*, an automatically-generated variation of the main issue that AccuRev uses to track merges, patches, and reverts. You generally do not need to concern yourself with variants, but if you would like to understand how they work, see *Variants* on page 369. If an issue number display includes ellipses (...), mouse over the box to display the list of issues.
- When you mouse over a version box, the tool tip for the version box displays information about the creation of the version: the transaction number and timestamp, the action (**keep**, **promote**, etc.), the stream name and version number of the current version and of the “real” version, as well as any comments that were entered at the time of the transaction. Note that in the case of alias versions (see below) the “current” version number in the tool tip is the same as the version in the yellow box, and the second version number in the tool tip reflects the real version, which differs from the version number displayed in the yellow box.

- The color-coded ancestry lines connecting the versions indicate how the later version was derived from the earlier version.

Lower (Tabular) Pane

The bottom half of the version browser displays in a table only stream and issue details that are relevant in the upper graphical pane. You can focus on either **Streams & Workspaces** or **Issues** by selecting the appropriate tab.

Streams & Workspaces Tab

- To highlight the versions associated with a particular stream or workspace in the upper pane, select that stream or workspace entry in the table in the lower pane. If a workspace is collapsed into one or more workspace groups in the upper pane, the versions in those workspace groups will be highlighted.
- To limit the display of streams and workspaces to those with names that include specific text, enter that text string in the **Name** filter.
- To limit the display of workspaces, select an option from the **Filter Workspaces** drop-down menu.
- To control the display of individual streams, use the checkboxes under the **Show** column, then click the **Refresh** button. **Note:** You cannot turn off the display of workspaces or workspace groups from the checkboxes. Use the **Filter Workspaces** control instead. You also cannot turn off the display of dynamic streams that contain real versions, such as those where a direct revert operation was performed. Streams and workspaces that cannot be hidden are always checked and cannot be unchecked.
- To identify which workspace group(s) a workspace belongs to (if any), look for a number in the "Groups" column.
- When stream labels are selected on the right of the top, graphical pane, they will highlight the corresponding rows in the Streams and Workspaces tab.
- If a single line is selected, then the context menu works exactly like it does on a stream label or a single workspace label. (The context menu would have no meaning on a workspace group.)

Issues Tab

- Select an issue's entry in the table to highlight the versions associated with that issue in the upper pane.
- Double-click or right-click an issue to open its Edit Form.

Workspace Groups

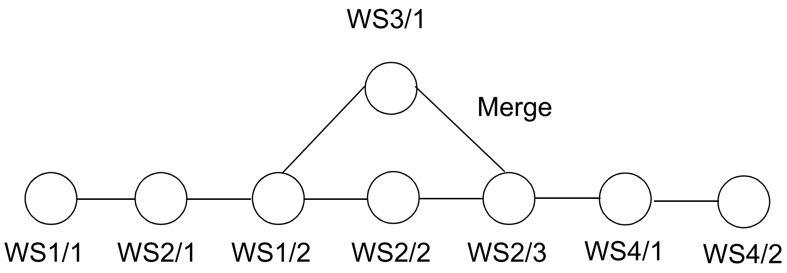
The Version Browser makes use of *workspace groups* to simplify the display of element versions that have been modified and promoted sequentially from multiple workspaces. Note that workspace groups do not exist or have any meaning anywhere other than the Version Browser.

Assume the following common sequence of events:

1. User1 creates and promotes a new file called file1.c in workspace WS1.
2. User2 updates her workspace (WS2), modifies file1.c, and promotes it.
3. User1 updates WS1, makes more changes to file1.c, and promotes it again.
4. User2 updates WS2 and makes yet more changes to file1.c.

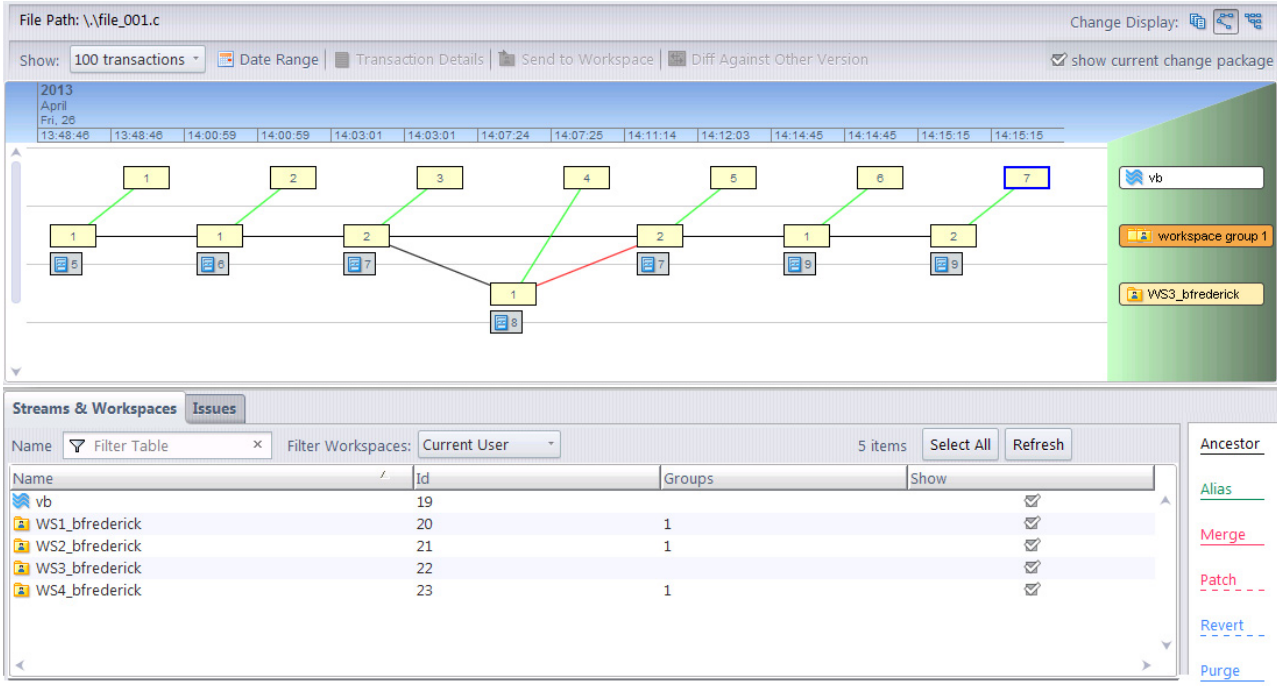
5. This time, User3 also makes changes in WS3 to the version of file1.c last promoted by User1. (User2 and User3 are making changes to them same file in parallel.)
6. User3 promotes his changes from WS3, causing an overlap conflict for User2.
7. User2 merges in User3's changes so she can promote her file.
8. User4 then updates his workspace (WS4) with User2's merged version, and makes his own changes and promotes. After a review, he makes some further changes and promotes again.

In a traditional version graph, the above scenario could be expressed as shown in the following figure:



In earlier versions of the AccuRev Version Browser, each workspace would have generated a new row in the graphical view, along with ancestry lines connecting the versions. In large installations with many workspaces and many changes, the Version Browser display could quickly become unmanageable.

As of AccuRev 6.0, the Version Browser provides a workspace group which represents all of the workspaces that fall along a common line in the version graph:


File Path: \.\file_001.c

Show: 100 transactions

☑ show current change package

Filter Workspaces: Current User

5 items

Name	Id	Groups	Show
vb	19		☑
WS1_bfrederick	20	1	☑
WS2_bfrederick	21	1	☑
WS3_bfrederick	22		☑
WS4_bfrederick	23	1	☑

This makes the version browser much easier to navigate and understand. To see which workspaces belong to which workspace groups, view the **Streams & Workspaces** tab in the lower pane and check the "Groups" column.

Ancestry Lines

The Version Browser uses these color-coded lines to indicate ancestry relationships

- *direct ancestor* (**black**) -- A user started with the earlier version, modified (or overwrote) it, and then performed a **Keep**, creating the later version. (There are a few other operations, including **Rename**, that might have created the later version. See below for details.)
- *alias* (**green**) -- A user promoted the earlier version to create the later version.
- *merge* (**red**) -- A user executed a *Merge* operation to create the later version. There are always two earlier versions: the black line indicates the direct ancestor (see above) of the later version; the red line indicates the version that was merged with the direct ancestor.
- *patch* (**dashed green**) -- Similar to *merge* above, but the later version was created by **Patch**.
- *revert* (**dashed blue**) -- The later version was created by a 'reverse patch', performed by **Revert**. This effectively 'undoes' the changes in a specified transaction or change package.
- *purge* (**blue)** -- The version in the user's workspace was restored to an earlier version with a **Purge** (**Revert to Basis**) operation.

Ancestry Relationships

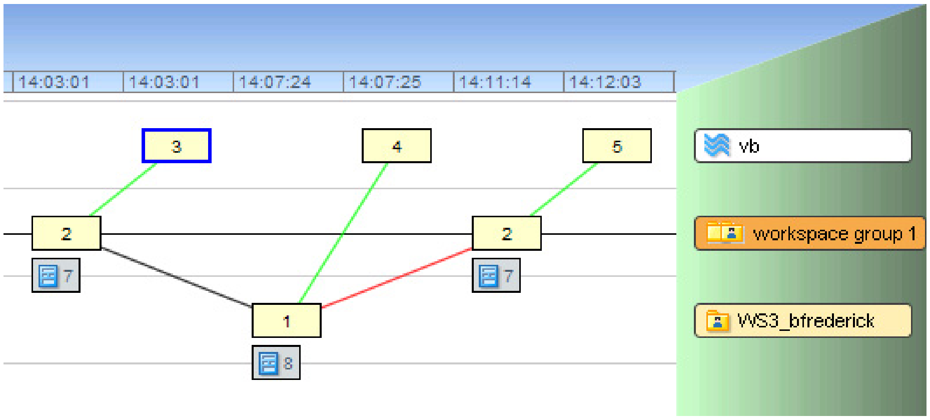
The following sections discuss the kinds of ancestry in more detail.

Direct Ancestor -- Modification of an Existing Version

Probably the most common AccuRev operation is starting with an existing version, making changes, and then executing a **Keep** to save the changes in a new version. The existing version might have been created by you -- for example, with a previous **Keep** operation. Or it might have been created by someone else: that user promoted the version, then you brought it into your workspace with an update.

The version in your workspace that is created by **Keep** is termed a real version, because it represents a change to the element. Other operations can create real versions in your workspace, too: **Rename**, **Defunct**, **Undefunct**.

The Version Browser uses a **solid black line** to connect the existing version (termed the direct ancestor or predecessor) with the new version. (Exception: a version created by a revert operation is connected to its direct ancestor by a **dashed blue line**.)

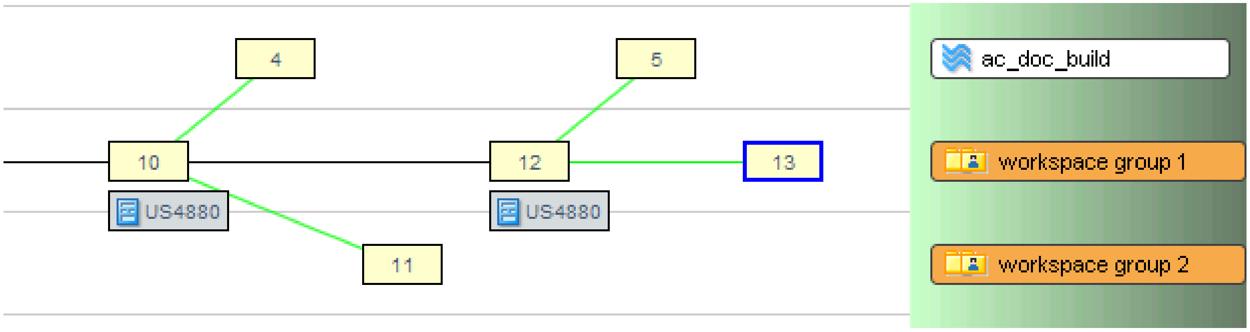


In the figure above:

- The left-hand box labeled “2” was created in a workspace within Workspace Group 1. If you mouseover the box, you would see that this is Version 2 in workspace WS1 (“WS1_*username*/2”).
- The right-hand box labeled “2” was created in a different workspace within in the same workspace group. (If you mouseover the box, you would see that this is Version 2 in workspace WS2 (“WS2_*username*/2”). It is just a coincidence that they are both Version 2 in their respective workspaces.) The black line connecting them indicates that version WS2/2 was created in part by updating WS2 with the changes from WS1 and then editing and promoting the file.
- Version 1 in WS3 was also created by updating from the changes from WS1 and then editing and promoting the file. However, these changes conflicted with the changes made by the user in WS2, and had to be merged. That is why there is red line rather than a black line between these two version.

Alias -- Virtual Version Ancestry

Workspaces contain real versions, which represent changes to elements. However, almost all versions in dynamic streams are virtual versions, created with **Promote**. (The one exception is real versions that are created in a dynamic stream using **Revert**.) Each virtual version is an alias for -- that is, a pointer to -- some real version in a user's workspace. The Version Browser uses a green line to connect a virtual version in a dynamic stream to the corresponding real version in a workspace.

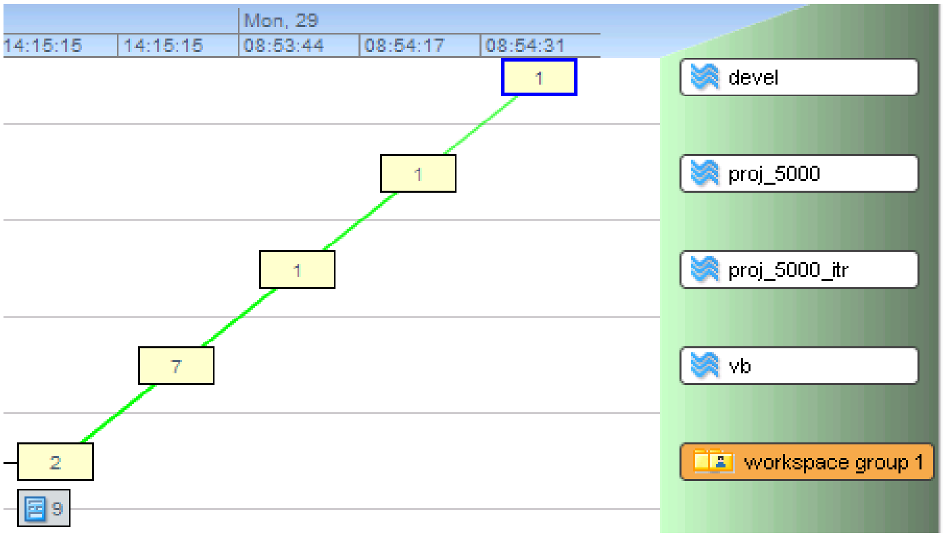


In the figure above:

- version 4 in stream *ac_doc_build* is an alias for (was promoted from) version 10 in a workspace within *Workspace Group 1*.
- version 5 in stream *ac_doc_build* is an alias for version 12 in a workspace within *Workspace Group 1*.
- version 11 in *Workspace Group 2* is an alias for version 10 in a workspace within *Workspace Group 1*.

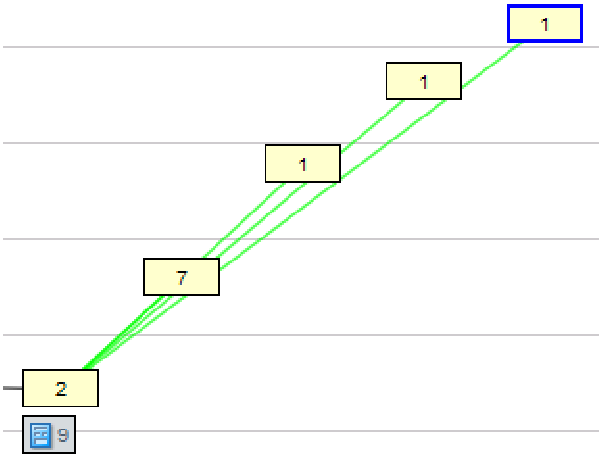
There's one exception to this scheme. The **Anchor** or **Send to Workspace** operation creates a virtual version in a workspace. It's a virtual version because it doesn't represent a change to the element, but merely the restoration of an existing version to the workspace.

Successive Promotions. In a depot with a deep stream hierarchy, it's common to successively promote a particular version to the parent stream, then to the grandparent stream, then to the great-grandparent stream, etc.



All the versions created by this series of promotes are aliases for the same real version. The Version Browser shows how all the virtual versions relate back to the original real version.

The default display shows all versions in a straight line. However, each virtual version is actually connected directly back to the real version in the workspace within Workspace Group 1, which can be seen by clicking and dragging each version box.



Merge -- Merging of Two Versions

A standard merge operation combines the contents of two versions of a file:

- The most recently kept version in your workspace stream. This version contains the changes that you have made to the file in your workspace.

  Note: It's possible -- but not a best practice -- to perform a merge on a file with **(modified)** status. Since you haven't preserved the recent changes with **Keep**, AccuRev will have no way to restore the file to its state just before the merge. This can be painful if you change your mind or make an error during the merge process.]

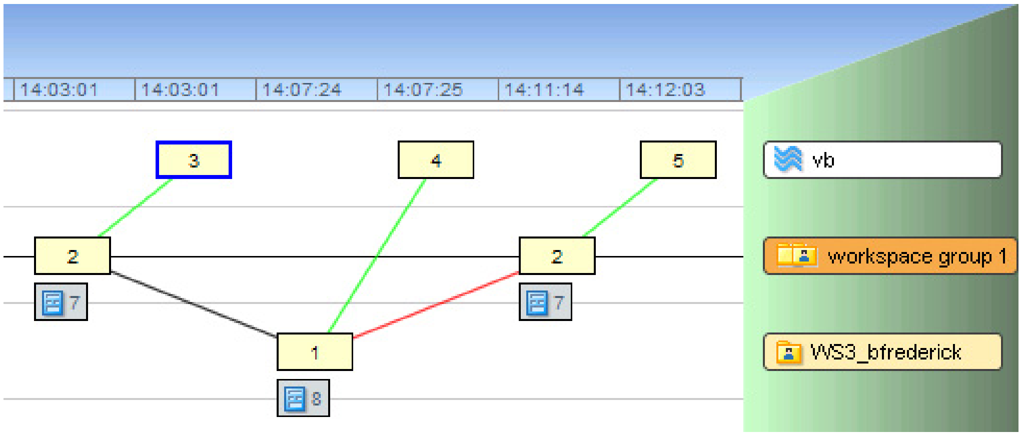
- The most recent version in the backing stream.

The result file of the merge operation is kept as a new version in the workspace stream. This new, merged version has two ancestors: the two versions listed above.

This is all simple enough. There's a twist, though, which shows up in the Version Browser display: AccuRev always records real versions, not virtual versions, as the two ancestors of a new, merged version. Thus, the ancestors in the standard merge scenario described above are:

- The most recently kept version in your workspace stream.
- The version in some other workspace stream that was promoted to the backing stream, causing the overlap that necessitated the merge.

Example: This screen shot shows a merge from the backing stream *vb* to a workspace within *Workspace Group 1*.



The new, merged version is *2* which is in a workspace within Workspace Group 1. (You can see which workspace by mousing over the version box. In this example, it is WS2.)

Its ancestors are:

- Real version WS1/2 (again, you can see workspace WS1 by mousing over the ancestor version box labeled “2”).
- Real version WS3/1, which was promoted to become virtual version vb/4 in the backing stream.

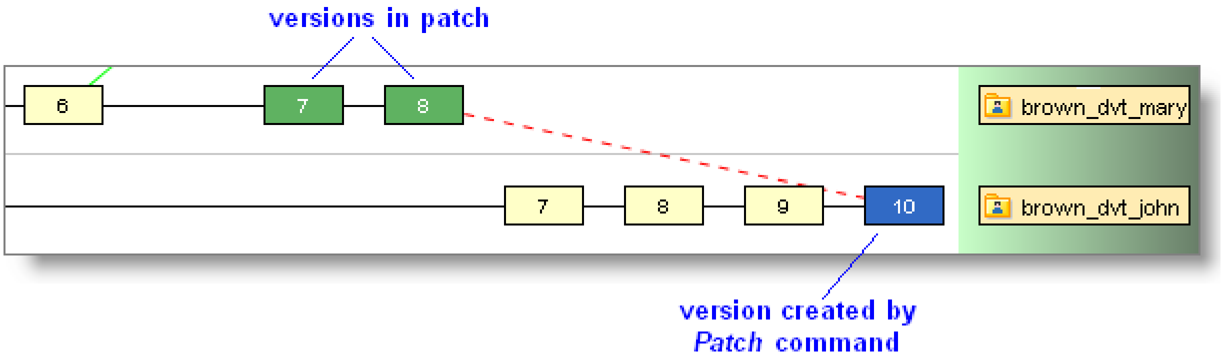
A solid red line shows the merging of data from one workspace, WS3, to a different workspace within Workspace Group 1.

Patch -- Selective Inclusion of Another Version's Changes

A patch operation is similar to a merge operation. In both, text from another version (the "from" version, at the left end of the dashed red line) is incorporated into your workspace's version. Here's the difference:

- A merge operation considers the entire contents of the "from" version.
- A patch operation considers only *some* of the changes in the "from" version. Typically, it's the 'recent changes' made by one user, recorded in one version or a set of consecutive versions. See Patch and Patch From on page 227 for full a discussion of patches.

When you select a version created by **Patch**, the Version Browser highlights in green the versions contained in that patch.



Note: Patching and the closest common ancestor

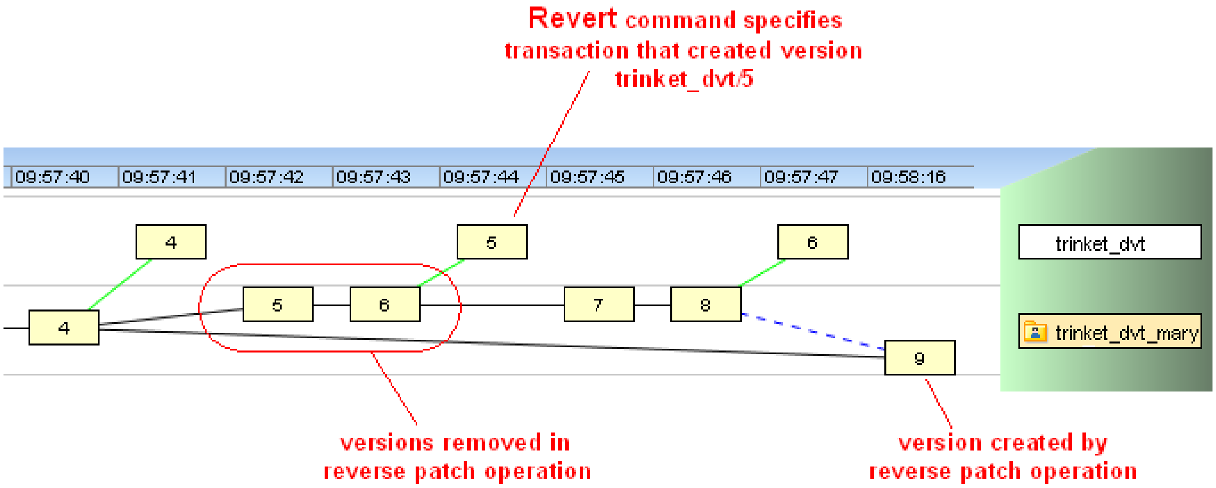
AccuRev tracks patch ancestry separately from merge ancestry. In determining the closest common ancestor of two versions for a merge operation, AccuRev takes into account previous merge operations (solid red), but not previous patch operations (dashed red) or revert operations (dashed blue).

Revert -- Selective Removal of Changes from a Version

A revert operation is the opposite of a patch operation. (And we describe *Revert* as performing a 'reverse patch' operation.) Whereas a patch *adds* a selected set of changes, a revert *removes* a selected set of changes.

A version created by *Revert* has two ancestry lines:

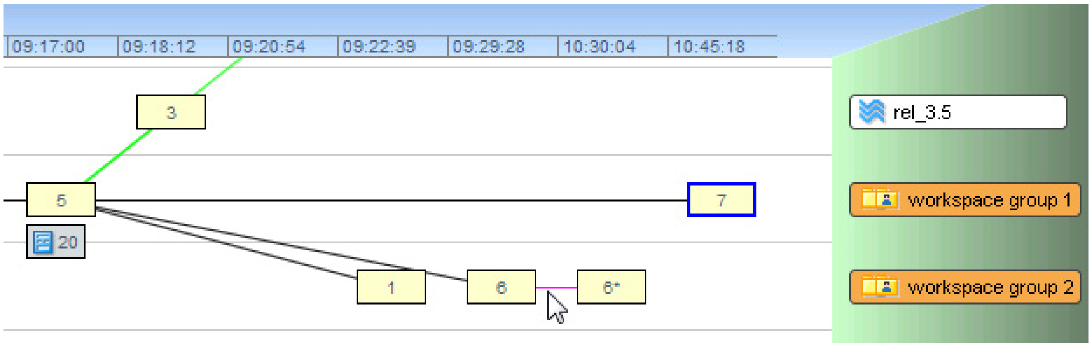
- A dashed blue line indicates the version from which *Revert* removed some changes.
- A solid black line indicates the basis version of the removed patch or change package entry.



See *Reverse Patch: Removing Content Changes* on page 238 for more information on how these ancestry lines are created.

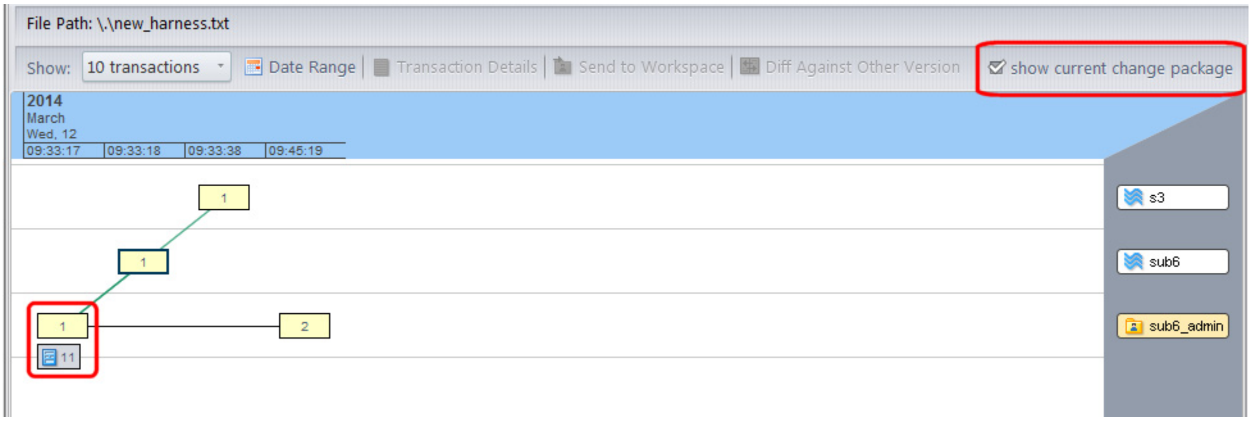
Purge -- Revert to Basis

The CLI **purge** command is analogous to the **Revert to Basis** function in the GUI. Assume that you have started to work on a file and that you have made one or more keeps in your workspace. Then something happens: perhaps the changes are no longer required due to a new project direction, or you've come up with a better approach and want to start over from scratch. So you perform a **Revert to Basis** (**purge**) to abandon all your keeps and restore the file to the way it was when you started working on it. This means that your keep versions are something of a dead-end. This is represented in the Version Browser by a magenta line and an asterisk in the version box.



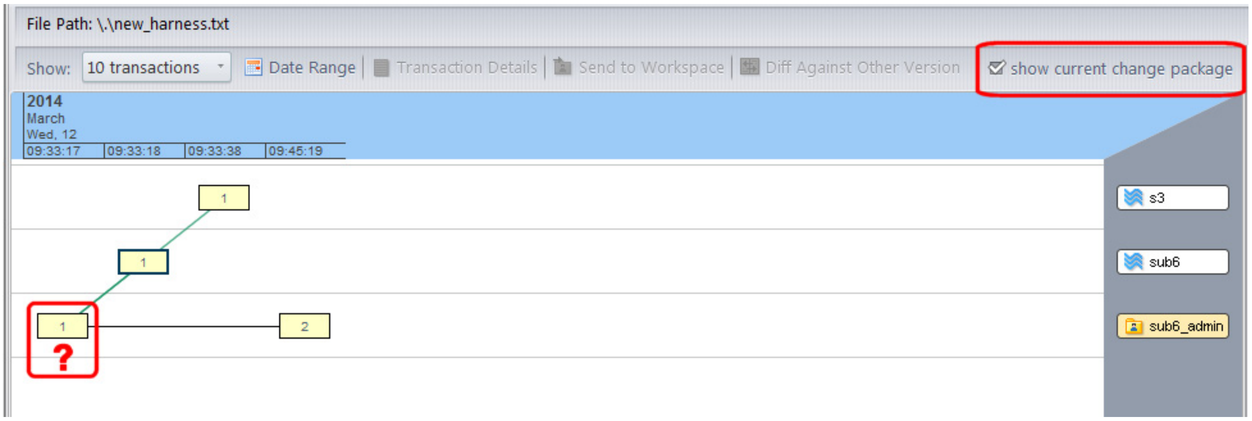
Displaying a Version's Change Package

By default, the Version Browser displays the change package that is *currently* associated with a given element version. In the following illustration, suppose that version 1 of the file **new_harness.txt** was promoted against issue 11. In this example, the Version Browser shows that --at the current time-- **new_harness.txt** is part of the change package for issue 11:

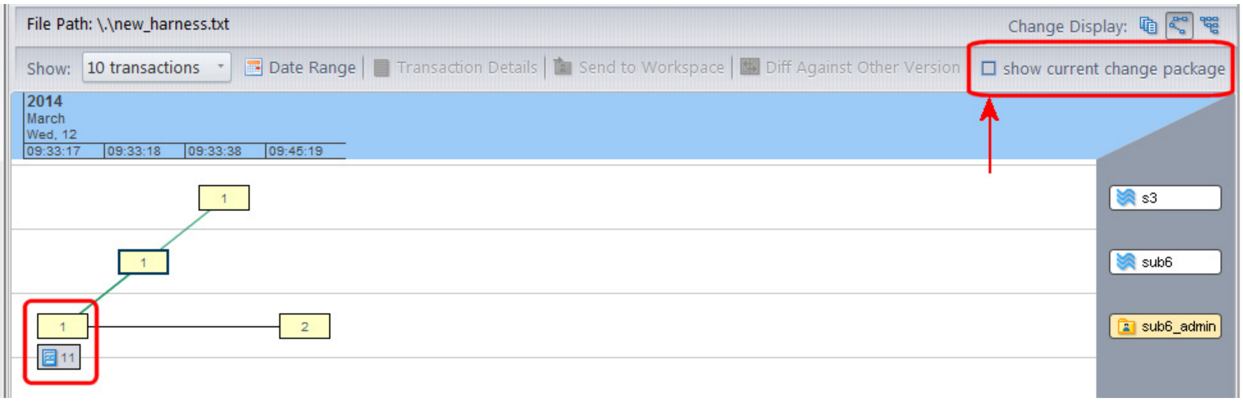

File Path: \.\new_harness.txt

Show: 10 transactions

Imagine **new_harness.txt** is subsequently removed from the change package. If you view the Version Browser at that time, you will see that it no longer displays a change package for version 1 of **new_harness.txt**:



This is because --at this time-- **new_harness.txt** is not part of the change package for issue 11. It was promoted against issue 11 previously, and you can view this history by clearing the **show current change package** checkbox, as shown here:



The **show current change package** checkbox, which acts as a toggle, can be used to help understand why a previously promoted version does not appear to be associated with a change package.

> **Note:** If you associate a version with a change package using Send to Issue and later remove that element from the change package, the Version Browser will not provide any insight into the change package history regardless of how the show **current change package** checkbox is set.

Closest Common Ancestor

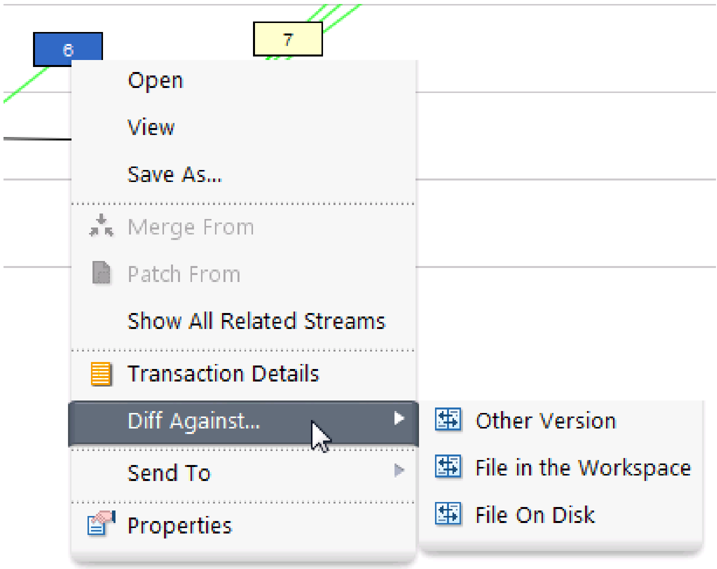
It's instructive to follow all the black and solid-red lines in an element's Version Browser display. This traces the entire ancestry of real versions of an element. In particular, you can use the real-version ancestry to determine the **closest common ancestor** of any two versions. This is the most recent version upon which the two versions are both based, by some combination of direct ancestor and merge connections. (When considering a virtual version in a closest-common-ancestor analysis, first follow the green line back to the corresponding real version.)

You can also use the CLI command *accurev anc -c* to find the closest common ancestor of two versions.

In a **Merge** operation, AccuRev determines the closest common ancestor of the two versions to be merged, and uses this version to perform a *3-way merge*. See *The Merge, Patch, and Reverse Patch Algorithms* on page 232 for more information on the merge algorithm.

Operations on Versions

You can perform several operations on a selected version, using its context menu or the Version Browser toolbar. This menu is available from a version, a stream label, or from a single-selected line in the Streams & Workspaces table.



Open (equivalent to double-click)

Windows: Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

UNIX: Open a text editor on the file.

View

Open a text editor on a temporary copy of the currently selected file (text files only).

Save As

Copy the currently selected file to another filename.

Merge From

Merge the selected version into the version in the workspace from which the Version Browser was invoked.

Patch From

Patch the selected version into the version in the workspace from which the Version Browser was invoked.

Show All Related Streams

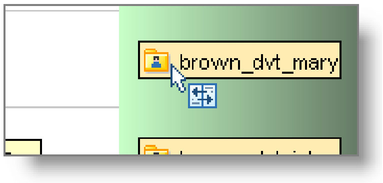
Display any streams in the hierarchy above the current stream, as far up as this element has been promoted. If all relevant streams are already displayed, you will not see any change when clicking this menu item.

Transaction Details

Displays information about the transaction that created this version, including the time stamp, user, operation, any related comments, and any other elements that belonged to the change package.

Diff Against Other Version

Compare the selected version with another version of the element.



AccuRev changes the mouse pointer, to prompt you to select the other version. You can click on any version in the Version Browser display. You can also click on a stream or workspace label to indicate the version currently in that stream or workspace.

Note: If the **Diff Against...** options on this context menu are grayed (disabled), you are probably attempting to diff a non-image binary file, which is not supported.

Diff Against File in the Workspace

Compare the selected version with the file in the workspace from which the Version Browser was invoked. This workspace is listed at the bottom on the AccuRev GUI window. This enables you to invoke a comparison with a file that you have changed in your workspace, but have not yet preserved in the repository with **Keep**.

Diff Against File On Disk

Compare the selected version with the an arbitrary file on your machine. A *File Chooser* dialog appears, in which you specify the file.

Send to Workspace

Activate the selected element in the workspace from which the Version Browser was invoked. The element must currently have **(backed)** status in the workspace -- that is, it must *not* be active.

Send to Issue

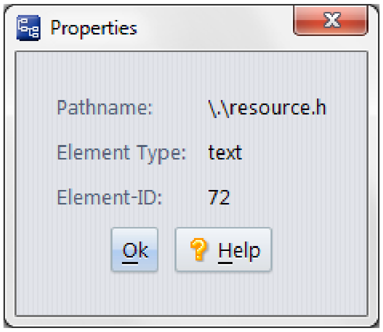
Record the selected version(s) in the change package section (Changes tab) of one or more issue records. The default query is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected version(s) will be sent.

Send to Issue (specifying basis)

Record the selected version in the change package section (Changes tab) of one or more issue records. You are prompted select a version in the Version Browser display; this version will become the basis version in the change package entry.Then, the default query is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected versions will be sent.

Properties

Displays information about the selected element in a pop-up window. The data items displayed vary with the type of element.



Pathname: \.\resource.h

Element Type: text

Element-ID: 72

7. Diff, Merge, Patch, and the Change Palette

AccuRev provides a variety of tools for performing common developer tasks. For comparing files, you can either use AccuRev's built-in diff tool or specify a third-party utility. AccuRev identifies files that must be merged, and can help you merge non-conflicting changes automatically, or provide you with the diff tool or the Change Palette to handle different types of merges manually. The Change Palette can also be used for patching files and cross-promoting them between streams.

The Diff Tool

The AccuRev GUI provides many ways to invoke a "Diff" command to compare two versions of a text-file element. By default, the comparison is performed by AccuRev's own **Diff** tool, which displays the two versions next to each other, using color coding to show the differences and provided convenient navigation controls. You can also configure the GUI to use another file-comparison tool (see *AccuRev Preferences (Tools > Preferences Command)* on page 39).

Invoking the Diff Tool (or Another File-Comparison Tool)

No matter which tool you've configured to perform file comparisons, you invoke the tool in the same way, as described below. AccuRev's Diff tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

There are multiple ways to specify the two versions of a file element to be compared:

In a File Browser, each file's context menu includes a **Diff Against...** item. There are several choices, each of which compares your version (i.e. the file in your workspace) with a version stored in the depot:

- **Most Recent Version:** Compares your file with the version currently in your workspace stream. Use this choice if you've modified the file since the last time you performed a *Keep* on it (or if you've never performed a *Keep* on it since the last update).
- **Backed Version:** Compares your file with the version currently in the backing stream. For example, you might use this choice to see *all* the changes you've made to this file since you updated your workspace and starting working on the file. (And assuming no one else has promoted a new version to the backing stream in the meantime.) This might include the changes stored in several intermediate versions that you've created with *keep*.
- **Basis Version:** Compares your file with the version that you started working with, before making your "recent" changes. For a discussion of the meaning of "recent", see *Patch and Patch From* on page 227.
- **Overlapping Changes:** Provides two options to help you identify and resolve conflicts between two versions of an element that result in an **(overlap)** status:

  - **Mine -** Compares the workspace or stream version against the common ancestor in the backing stream. This option identifies the changes in the workspace or stream that did not come from the backing stream; it is the equivalent of diffing against the last version that you merged with.
  - **Theirs -** Displays only the changes added to the backing stream since the last time you merged with the backing stream. This option is useful for identifying the changes to the backing stream that caused the overlap condition.

  The Overlapping Changes options are available only for elements with an **(overlap)** status.

- **File in Other Stream:** Compares the selected file with a file in another stream or workspace. By default, AccuRev selects the file with the same name in the target stream or workspace you select, but you can choose any file you wish.
- **File on Disk:** Compares your file with an arbitrary file on your machine. A standard operating system "open file" dialog box open, in which you specify the file.

In a Stream Browser display, view the elements in a stream's default group by clicking the control below the stream. An element's context menu includes the *Diff Against...* items described above.

In a Version Browser or Stream Version Browser display for a text-file element, you can compare any two versions: right-click any version, select ***Diff Against Other Version*** from the context menu, then left-click any other version.

In a History Browser display for a text-file element, you can compare the versions created in any two transactions. Right-click a transaction in the Summary (upper) pane, select **Diff Against Other Version** from the context menu, then left-click another transaction in the Summary pane. Similarly, the **Diff Against File in the Workspace** command enables you to compare the version created in a particular transaction with the file currently in your workspace. (That file might contain changes that you haven't yet preserved with **Keep**.)

> **Note:** If a text file has multiple segments because it has been promoted against different AccuWork issues and you want to compare only the versions of a file that were promoted to the *same* issue, you can accomplish this by using the **Diff Across All Segments** option. This option is available on the **Changes** tab of an open AccuWork issue. For more information about this option, see *Comparing Versions Across Segments*.

Diffing One File Against Another

Unlike most other **Diff Against** operations, in which you choose which version of a file you want to diff, there are two **Diff Against** operations that allow you to diff one file with any other file you choose, **Diff Against File in Other Stream** and **Diff Against File on Disk**.

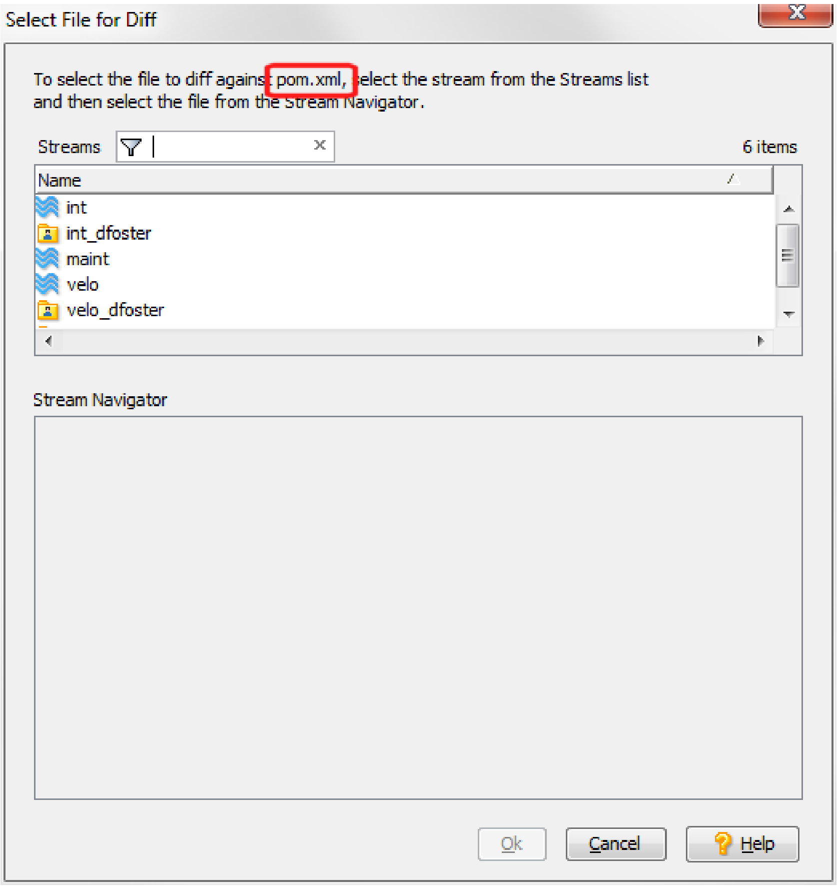
Diff Against File in Other Stream

Use this procedure when you want to diff two files that are under AccuRev control.

To diff a file against a file in another stream:

1. In the File Browser, right-click the file you want to diff and choose **Diff Against** > **File in Other Stream** (or type **Ctrl+Shift+O**).

   AccuRev displays the Select File for Diff dialog box:

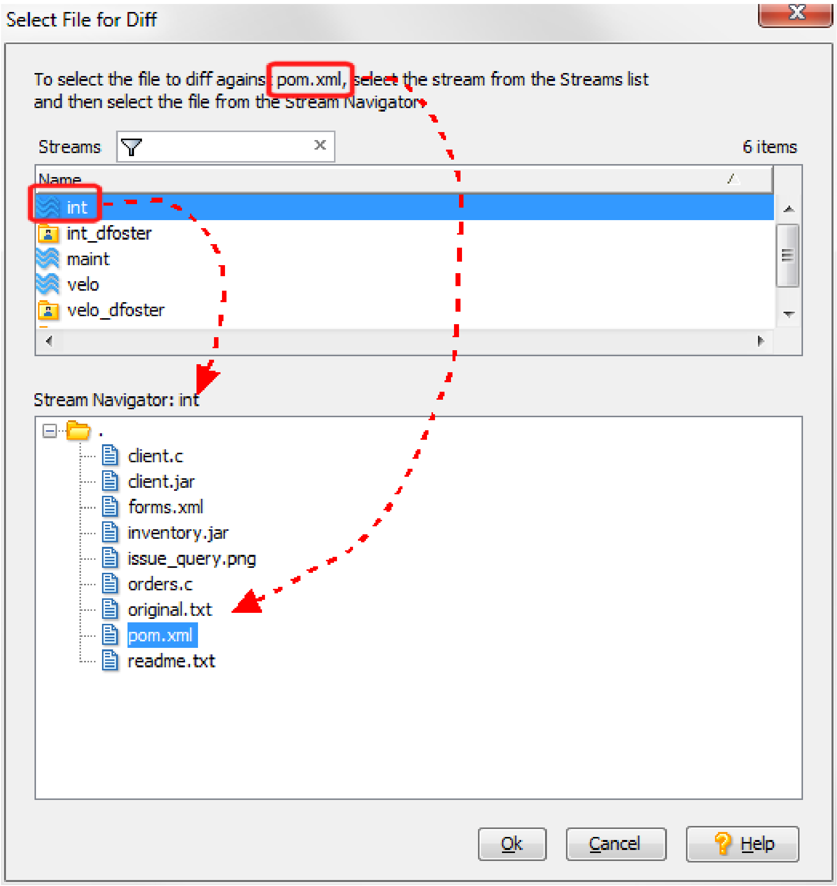


To select the file to diff against pom.xml, select the stream from the Streams list and then select the file from the Stream Navigator.

Note that the file you selected is displayed in the instructional text.

2. In the Streams field, select the stream or workspace associated with the file you want to diff against the file you selected in *Step 1*.

The Stream Navigator is populated with a tree for the stream or workspace you select.



Select File for Diff

To select the file to diff against pom.xml, select the stream from the Streams list and then select the file from the Stream Navigator.

Streams 6 items

Name
int
int_dfoster
maint
velo
velo_dfoster

Stream Navigator: int

.
client.c
client.jar
forms.xml
inventory.jar
issue_query.png
orders.c
original.txt
pom.xml
readme.txt

If the file you chose for the **Diff** operation exists in the selected target stream, AccuRev automatically highlights the file in the Stream Navigator pane. You can choose a different file for the **Diff** operation, if appropriate.

Note: To locate the file in the target stream, AccuRev navigates as much of the element's path as possible. If AccuRev cannot find a file with the same name in the stream or workspace you select, it displays a warning message.

3. Click **Ok** to diff the two files.

   AccuRev displays the Diff view. See *Diff Tab Layout* on page 207 to learn about how diff files are displayed.

Diff Against File on Disk

Use this procedure when you want to diff a file under AccuRev control with any file on a local or networked disk.

1. In the File Browser, right-click the file you want to diff and choose **Diff Against** > **File On Disk** (or type **Ctrl**+**Shift**+**F**).

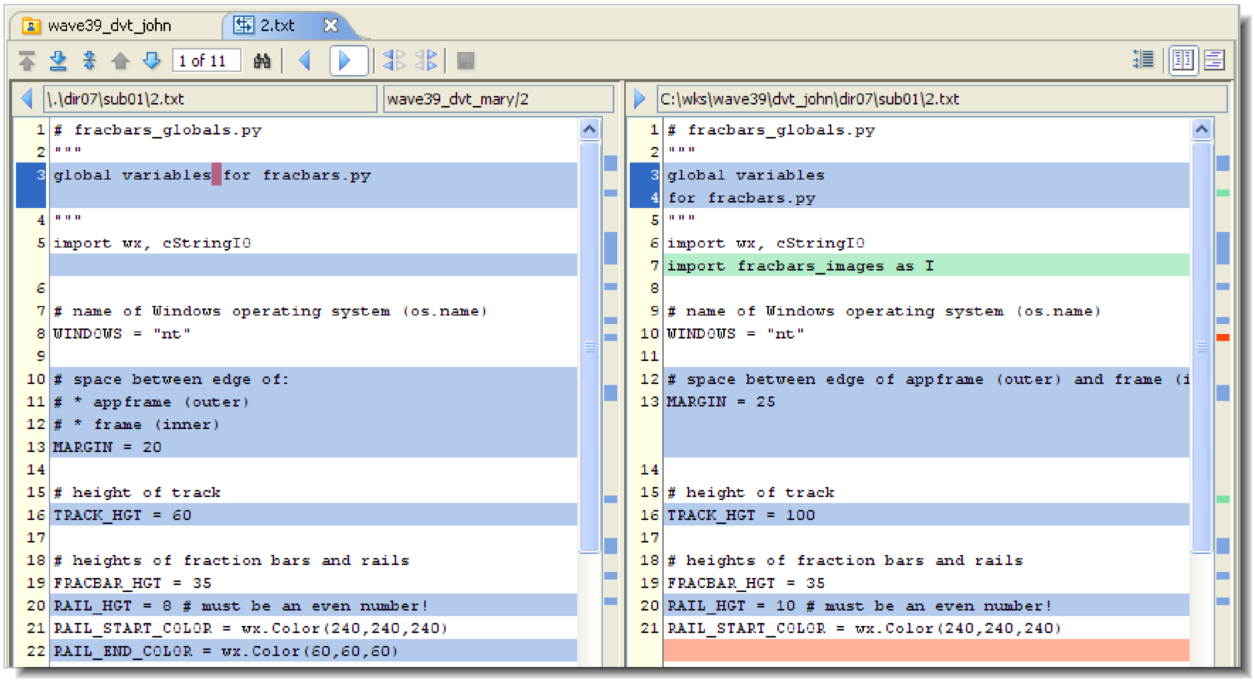
   AccuRev displays the Open dialog box.

2. Select the file against which you want to diff the file selected in *Step 1* and click the **Diff** button.

AccuRev displays the Diff view. See *Diff Tab Layout* on page 207 to learn about how diff files are displayed.

Diff Tab Layout

The Diff tool displays the two versions side-by-side in separate panes, so that corresponding text lines in the versions line up visually. (You can use buttons **Horizontal Layout** and **Vertical Layout** at the right side of the toolbar to arrange the panes above-and-below.) We'll call them the "before version" (displayed on the left, or above) and the "after version". The element pathname and the version-IDs are displayed above the versions' contents.



\.\dir07\sub01\2.txt

wave39_dvt_mary/2

```
# fracbars_globals.py
"""
global variables for fracbars.py
"""
import wx, cStringIO

# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of:
# * appframe (outer)
# * frame (inner)
MARGIN = 20

# height of track
TRACK_HGT = 60

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 8 # must be an even number!
RAIL_START_COLOR = wx.Color(240,240,240)
RAIL_END_COLOR = wx.Color(60,60,60)
```

C:\wks\wave39\dvt_john\dir07\sub01\2.txt

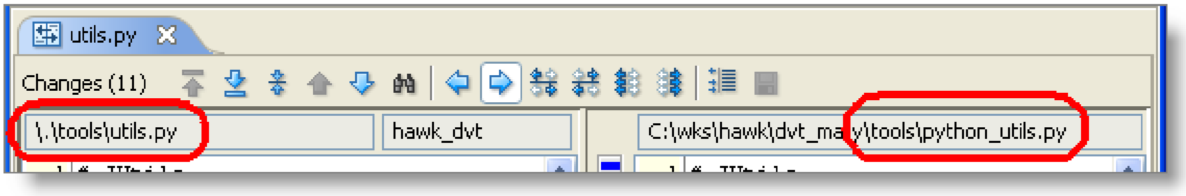
```
# fracbars_globals.py
"""
global variables
for fracbars.py
"""
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of appframe (outer) and frame (i
MARGIN = 25

# height of track
TRACK_HGT = 100

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 10 # must be an even number!
RAIL_START_COLOR = wx.Color(240,240,240)
```

Initially, the two panes are the same width, but you can drag the vertical separator to change the relative widths. To make both panes wider, just increase the size of the overall AccuRev GUI window.

Notes:

- *Namespace differences*

  The two versions being compared can have different pathnames: the element may have been renamed or moved to another directory within the depot. (This may have occurred in one of the versions, or in both of them.) The **Diff** tool does not highlight or announce such namespace differences.



\.\tools\utils.py

hawk_dvt

C:\wks\hawk\dvt_ma y\tools\python_utils.py

- *Left vs. right*

  Depending on how you launch the Diff tool, an older version might be displayed on the right, and a newer version on the left. We recommend keeping the older version on the left, so that "before" and "after" in the descriptions below correspond to reality.
- *Comparing binary files*

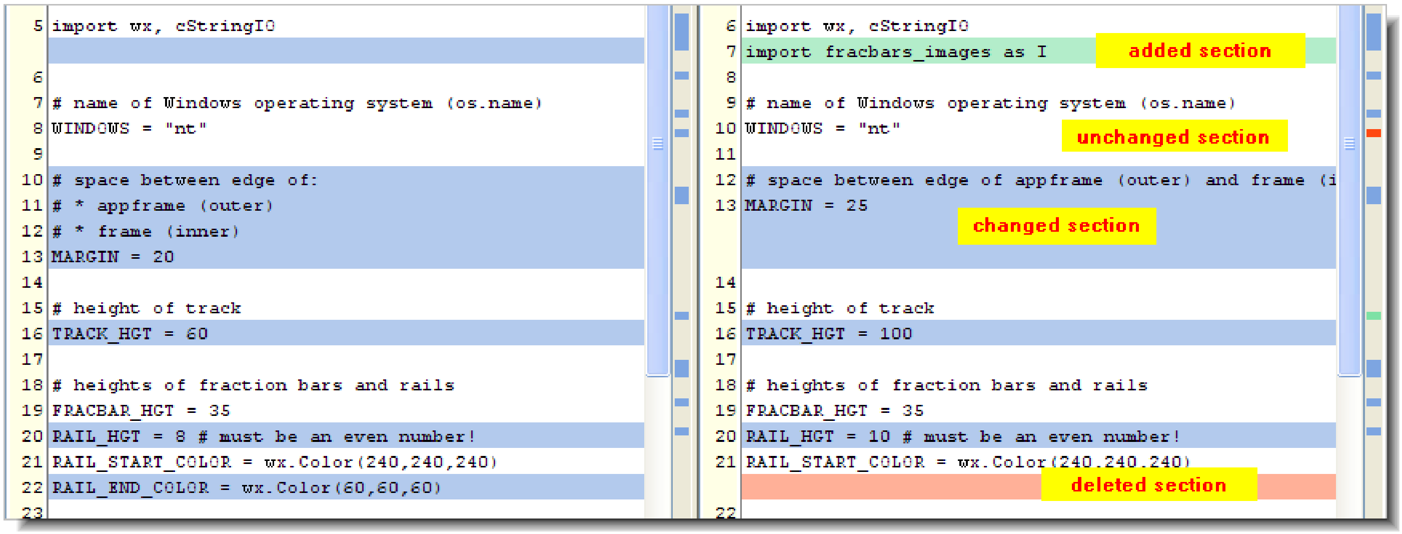
  If the versions to be compared are in a binary image format that AccuRev can render, the Diff tool simply displays the versions, so that you can determine their differences by inspection. AccuRev can render the following image formats:
  - JPEG (**.jpg** or **.JPG** filename suffix)
  - PNG (**.png** or **.PNG**)
  - GIF (**.gif** or **.GIF**)
- *Applying changes from the "before" version to your workspace's version*

  If you're comparing the file in your workspace with another version, you can use the Diff tool to edit your file as you view the differences.

  See *Editing a File Using the Diff Tool* on page 211.

Difference Section Color-Coding

Through color-coding, the **Diff** tool partitions the text into the following kinds of "difference sections":

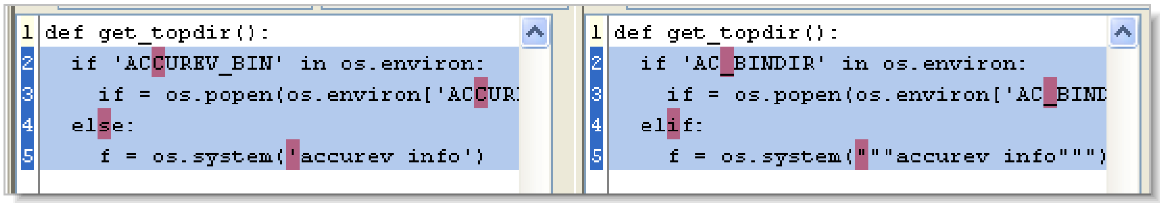


```
import wx, cStringIO



# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of:
# * appframe (outer)
# * frame (inner)
MARGIN = 20

# height of track
TRACK_HGT = 60

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 8 # must be an even number!
RAIL_START_COLOR = wx.Color(240,240,240)
RAIL_END_COLOR = wx.Color(60,60,60)

```

```
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of appframe (outer) and frame (i
MARGIN = 25



# height of track
TRACK_HGT = 100

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 10 # must be an even number!
RAIL_START_COLOR = wx.Color(240.240.240)


```

- *Unchanged Section* (white background): Text section in which the versions are identical.
- *Added Section* (green background): Text section that occurs only in the "after" version. Empty space with a blue background appears at that point in the "before" version.
- *Deleted Section* (red background): Text section that occurs only in the "before" version. The deleted section appears with a blue background in the "before" version.
- *Changed Section* (blue background): Sometimes, the Diff tool decides that a text section in the "before" version has been revised, producing the corresponding section in the "after" version. The before and after sections are not necessarily the same length, in which case some empty space is displayed in one of the versions. Both the before and after sections appear with a blue background.

Both sections contain red character highlights, indicating the first location in a line where the two versions differ:



```
def get_topdir():
  if 'ACCUREV_BIN' in os.environ:
    if = os.popen(os.environ['ACCUR
  else:
    f = os.system('accurev info')
```

```
def get_topdir():
  if 'AC_BINDIR' in os.environ:
    if = os.popen(os.environ['AC_BIND
  elif:
    f = os.system("""accurev info""")
```

If a text block has been moved from one location in the file to another, the Diff tool indicates this as two separate changes: a deleted section at the original location, and an added section at the new location.

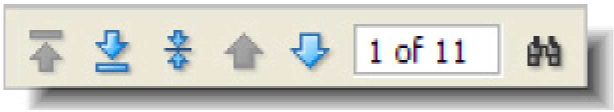
Your user preferences include controls for the Diff tool's handling of whitespace-only differences between the versions. (See *AccuRev Preferences (Tools > Preferences Command)* on page 39.)

Navigating in a Diff Tab

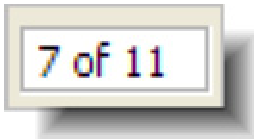
Most text files are too long to fit on the display screen, and some individual text lines are too wide. Accordingly, the panes in which the versions appear have both vertical and horizontal scroll bars. Scrolling of the panes is synchronized automatically. Line numbers appear at the left edge of each pane.

For a text file that contains hundreds or thousands of lines, there may be only a few difference sections (added, deleted, or revised), separated by large unchanged sections. The **Diff** tool provides both navigation buttons on its toolbar and a difference map for direct access to a particular difference section.

Diff Toolbar Navigation Buttons



Whenever you use one of these buttons to jump to a particular difference section, the **Diff** tool remembers it as the current difference and highlights the lines numbers in both panes.



A "difference N of M" window in the toolbar indicates which difference section is current.

First Diff

(Accelerator key: **Crtrl+Shift+↑**) Go to the first difference section.

Last Diff

(Accelerator key: **Crtrl+Shift+↓**) Go to the last difference section.

Center Diff

Using the scroll bars or the difference map, you can make the current difference scroll offscreen. Click this button to bring it back onscreen.

Previous Diff

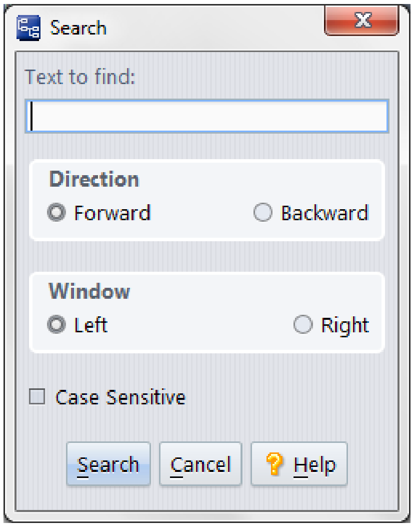
(Accelerator key: **Ctrl+↑**) Go to the previous difference section.

Next Diff

(Accelerator key: **Ctrl+↓**) Go to the next difference section.

Search

(Keyboard shortcut: **Ctrl+F**) Search for a text string, in either of the versions being compared. A dialog appears, in which you define the search.



Text to find:

Direction: ● Forward ○ Backward

Window: ● Left ○ Right

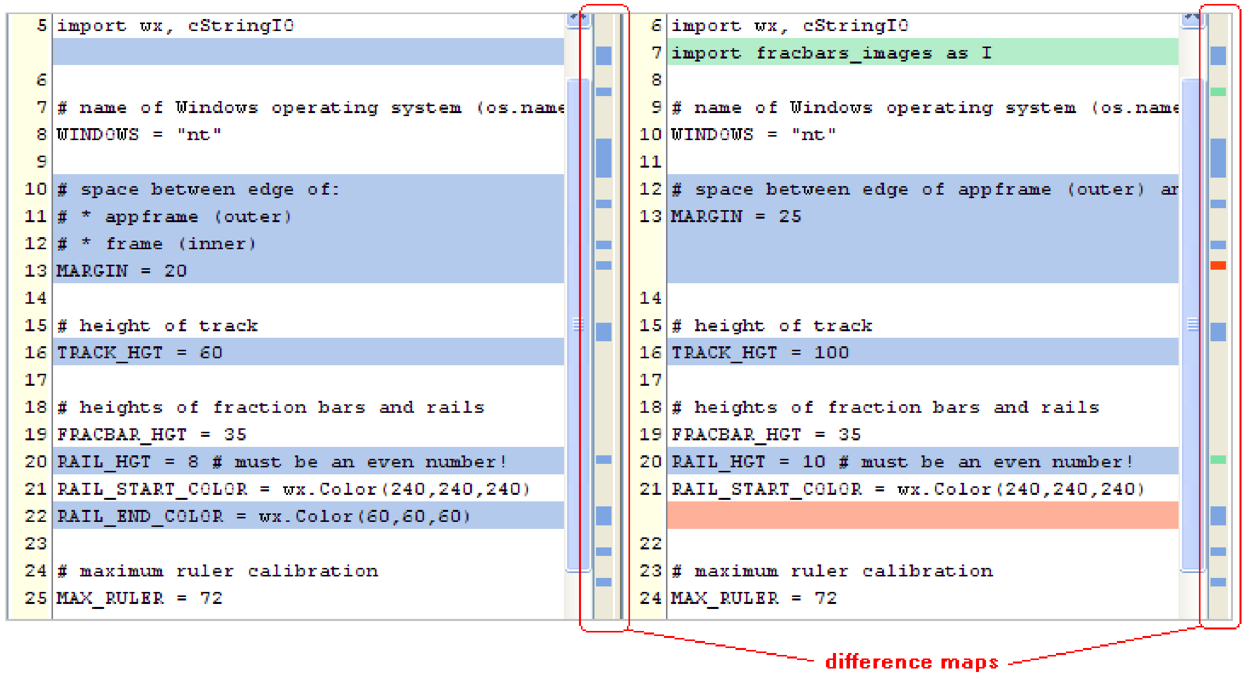
□ Case Sensitive

This command is also available on the GUI window's main menu: **File > Search**.

After you've defined a search, you can perform it again, using the command **File > Search Again** on the GUI window's main menu, or its keyboard shortcut: function key **F3**.

Difference Maps

To the right of each pane, there's a difference map that shows the relative locations and sizes of all the difference sections. The maps use the same color-coding (see *Color-Coding of the Contributor Versions* on page 217) as the difference sections themselves. Click on any colored area within a map to scroll both panes directly to the corresponding difference section. (Actually, you can click anywhere in the difference map; the panes will scroll to that location, even if the files are identical there.)



```
import wx, cStringIO


# name of Windows operating system (os.name
WINDOWS = "nt"

# space between edge of:
# * appframe (outer)
# * frame (inner)
MARGIN = 20

# height of track
TRACK_HGT = 60

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 8 # must be an even number!
RAIL_START_COLOR = wx.Color(240,240,240)
RAIL_END_COLOR = wx.Color(60,60,60)

# maximum ruler calibration
MAX_RULER = 72
```

```
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name
WINDOWS = "nt"

# space between edge of appframe (outer) ar
MARGIN = 25

# height of track
TRACK_HGT = 100

# heights of fraction bars and rails
FRACBAR_HGT = 35
RAIL_HGT = 10 # must be an even number!
RAIL_START_COLOR = wx.Color(240,240,240)


# maximum ruler calibration
MAX_RULER = 72
```

Using a difference map merely scrolls the panes; it does not affect which difference is the current difference. After you jump to a particular location, click the difference section in either pane to make it the current difference.

If you use a difference map, but then want to scroll back to where you were before, click the **Center Diff** toolbar button.

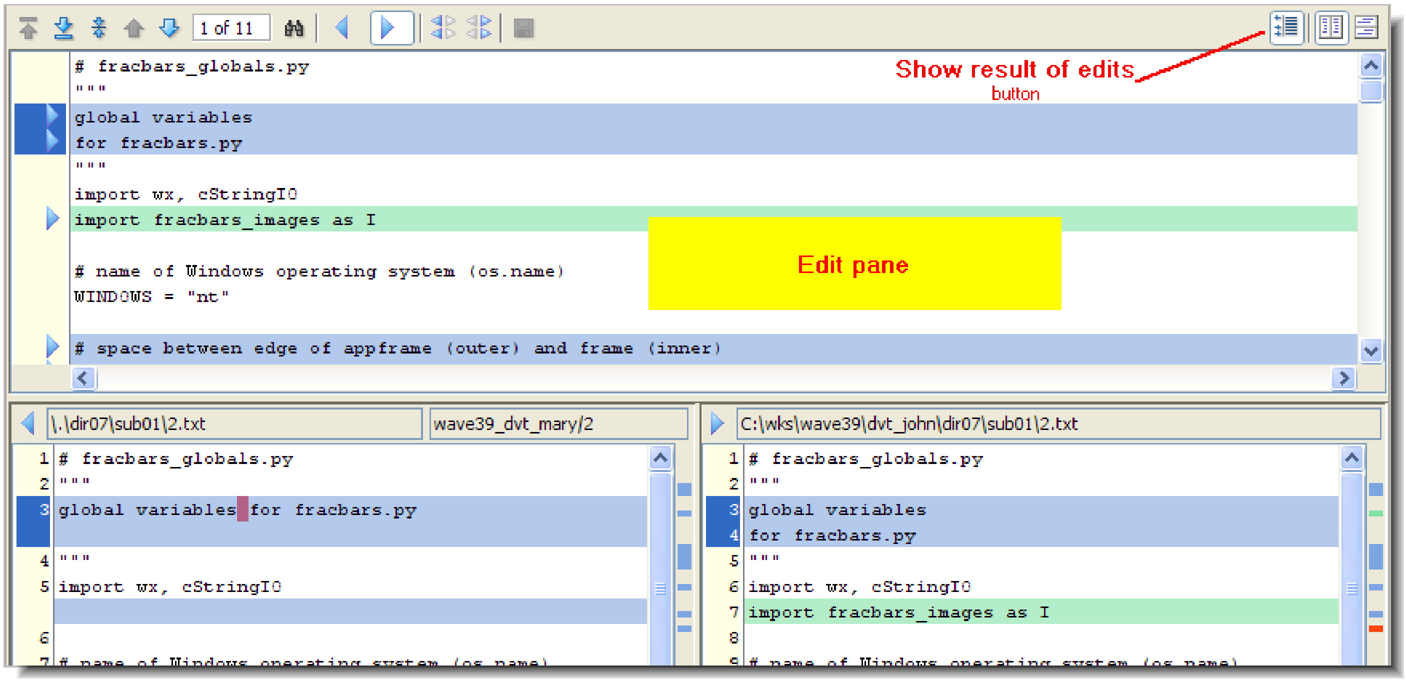
Editing a File Using the Diff Tool

In addition to comparing two versions of a file, the **Diff** tool can help you to edit the contents of the version in your workspace. This "edit-by-diff" capability is available only when you're comparing your workspace version with another version; it's not available when you invoke the **Diff** tool from the Version Browser or History Browser.

Here's a procedure for using this capability (see the toolbar button summary below):

1. Invoke the **Diff** tool in a File Browser, to compare your workspace version with some other ("before") version of the same file.

2. Click the **Show result of edits** button in the **Diff** toolbar. This opens a third pane (the Edit pane), which initially contains the same text as your workspace version.



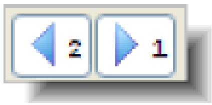
```
# fracbars_globals.py
"""
global variables
for fracbars.py
"""
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of appframe (outer) and frame (inner)
```

```
# fracbars_globals.py
"""
global variables for fracbars.py

"""
import wx, cStringIO


# name of Windows operating system (os.name)
```

```
# fracbars_globals.py
"""
global variables
for fracbars.py
"""
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name)
```

3. As you browse through the difference sections, you can:

   - Edit text manually in the Edit pane.
   - Click the **Revert my change** toolbar button to swap in the "before" version's text at that point (a change, addition, or deletion). The button becomes highlighted to indicate your selection.
   - Include both the "before" and "after" versions' text at that point: click the **Revert My Change** toolbar button, then hold down the **Ctrl** key and click the **Restore my change** toolbar button.



   - Alternatively, click the two buttons in the opposite order. Both buttons become highlighted, and number annotations indicate the order you selected:

     You can then modify the combined text manually.

     - Change your mind about a difference section where you've used and/or , by selecting that section and clicking one of those buttons. Any manual edits you've made in that difference section are lost.
     - Click the **Revert all of my changes** button to make the Edit pane the same as the "before" version.
     - Click the **Restore all of my changes** button to make the Edit pane the same as the "after" version.

4. Finish your work in either of these ways:

   - Click the 💾 **Save edits and close** toolbar button. This simply replaces the file in your workspace. (It doesn't perform an AccuRev **Keep** command.)
   - Close the Diff tab to cancel the edit-by-diff session. No changes are saved to the file in your workspace.

*Note: **Edit-by-Diff** vs. **Merge***

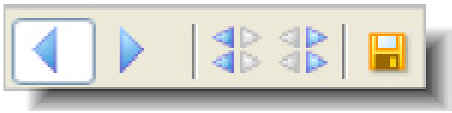
Using the edit-by-diff capability is similar to using the **Merge Tool** (see *The AccuRev Merge Tool* on page 216). But there's an important difference:

- An edit-by-diff operation involves just two versions.
- A merge operation takes into account a third version: the closest common ancestor of the two versions being merged.

Diff Previous Transaction Error

This error box appears when the selected version is the *first* version created in a stream or workspace. Thus, the search for a prior version in the stream or workspace fails.

Diff Toolbar Edit-by-Diff Buttons



◀ Revert my change

Use the "before" version's section at this point. If the ▶ button is currently highlighted, you can hold down the **Ctrl** key and press ◀ to append the "before" section to the workspace version's section.

▶ Restore my change

Use the workspace version's section at this point. If the ◀ button is currently highlighted, you can hold down the **Ctrl** key and press ▶ to append the workspace version's section to the "before" section.

Revert all my changes

Make the Edit pane the same as the "before" version.

Restore all my changes

Make the Edit pane the same as the "after" version.

💾 Save edits and close

Store the current contents of the Edit pane in the workspace's file, without performing a **Keep** operation.

Main Menu Commands Available in a Diff Tab

Several of the commands in the *Edit* submenu of GUI window's main menu are enabled when you're working in a *Diff* tab:

Cut

(Keyboard shortcut: **Ctrl-X**) Cut the selected text to the window system's clipboard.

Copy

(Keyboard shortcut: **Ctrl-C**) Copy the selected text to the window system's clipboard.

Paste

(Keyboard shortcut: **Ctrl-V**) Paste the contents of the window system's clipboard.

Select All

(Keyboard shortcut: **Ctrl-A**) Select the entire contents of the current pane.

Search

(Keyboard shortcut: **Ctrl-F**) See the *Search* description above.

Search Again

(Keyboard shortcut: **F3**) See th*e Search d*escription above.

Merge

The **Merge** operation combines two versions of an element, producing a new version. (That is, a successful merge operation always ends with a **Keep**. This is a software configuration management best practice.) AccuRev can process both content changes and namespace changes in performing a merge. By default, the merging of text-file contents is performed by AccuRev's own Merge tool, which uses a 3-way merge algorithm (see *Versions in a 3-Way Merge* on page 233).

What about elements that aren't text files?

Video Tip	Click *this link* to view an introductory video on merging binary files.

The **Merge** command can also handle any other kind of version-controlled element: binary file, directory, or link.

Configuring the GUI to use another text-file-merge tool

A user preference enables you to configure a third-party tool for merging the contents of text files.

Why Do I Need to Merge?

Perhaps the most common usage pattern in AccuRev is:

- Using **Keep** to create one or more versions of a text file in your workspace.
- Using **Promote** to propagate the most recently kept version to the backing stream.

Occasionally, someone else "gets there first". That is, both you and a colleague are working on the same file concurrently, each of you using a copy of the file in your own workspace. And the colleague promotes his changes to the common backing stream before you do.

Before you can promote your own changes to the backing stream, you must first *merge* (that is, combine) the changes from your colleague's version with your changes. Merging ensures that no one's work is inadvertently lost or overwritten in a concurrent development environment. Files that require a merge prior to promotion have a status of **(overlap)** -- that is, the work of one or more of your colleagues has overlapped your own work.

The File Browser makes it easy to tell which files require a merge prior to promotion: files with an **(overlap)** status, as well as other files that require special attention before they can be promoted, are displayed in the File Browser's Conflicts mode (⚠ Conflicts). In addition, files with **(overlap)** status are highlighted in yellow in all File Browser modes.

Higher-Level Merges

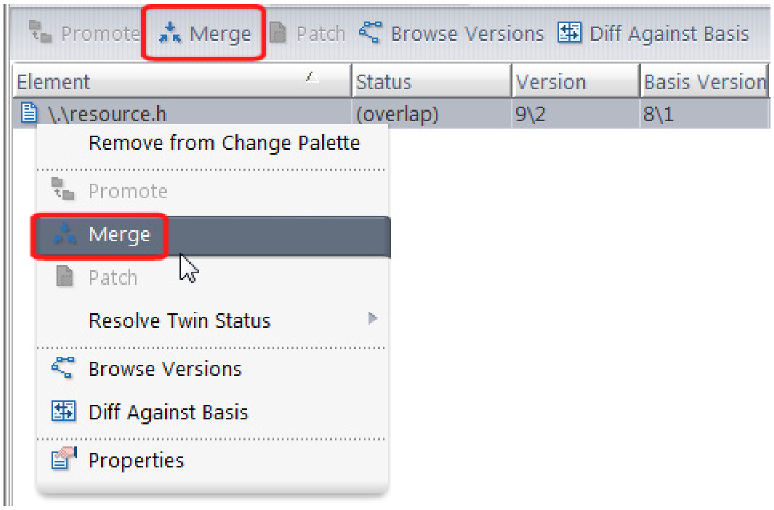
Although the above scenario is by far the most common one, AccuRev's flexibility allows for other merge scenarios, too:

- You can modify the file in your workspace by "pulling in" the changes from *any* version of the file, not just the backing stream's version.
- You can merge the versions of the file located in any two dynamic streams or snapshots. This involves using the *Change Palette*.

In all cases, however, you use the Merge tool in the same way, to combine the contents of two versions of the same file element.

Invoking Merge

No matter which tool you've configured to merge versions, you invoke the **Merge** operation in the same way: select one or more files with **(overlap)** status, then click the **Merge** toolbar button. Alternatively, choose ***Merge*** or **Merge From** on the selection's context menu. For example:



Element	Status	Version	Basis Version
\.\resource.h	(overlap)	9\2	8\1

You can merge versions in the following contexts:

- In the File Browser (when invoked from a workspace), to merge from the version in the parent stream.

- In the Version Browser (when invoked from a workspace), to merge from an arbitrary version.
- In the History Browser (when invoked from a workspace), to merge from the version in a particular transaction.
- In the Change Palette, to merge versions in higher-level streams. (You must choose a workspace to use for the merge process.)

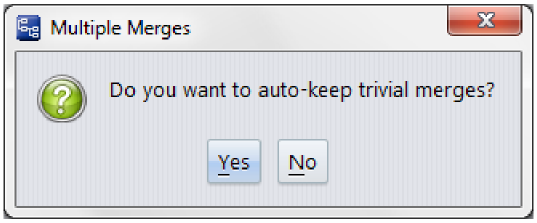
AccuRev processes the files you specify one-by-one. For each file, it does one or both of the following

- Prompts you to resolve namespace-level conflicts between the file in your workspace and the other specified version. This involves making choices in one or more dialogs.
- Invokes AccuRev's Merge tool or the other tool you've configured, in order to merge the contents of the two versions. AccuRev's Merge tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

Notes:

- *Invoking the Merge command on multiple files*

  If you select multiple files, you can choose to have the Merge tool process as many of them automatically as possible.

  Multiple Merges

  Do you want to auto-keep trivial merges?

  An automatic merge is possible if there are no conflicting changes between the two versions being merged. For files in which there are conflicting changes, the Merge tool always opens an interactive Merge tab.

- *File vs. Version*

  In all cases, one of the contributors to a merge operation is the file in your workspace. If you've just saved the file with the *Keep* command, the file in your workspace is identical to the most recent version in your workspace stream. But if you invoke *Merge* on a file that you've edited but not yet saved with *Keep*, AccuRev uses the file, not the latest workspace-stream version. Keeping before Merging is an AccuRev best practice.

The AccuRev Merge Tool

Video Tip	Click *this link* to view an introductory video on using the Merge tool to resolve conflicts.

This topic describes AccuRev's own 3-way merge tool. You can also configure the GUI to use another text-file-merge tool.

Merge Tab Layout

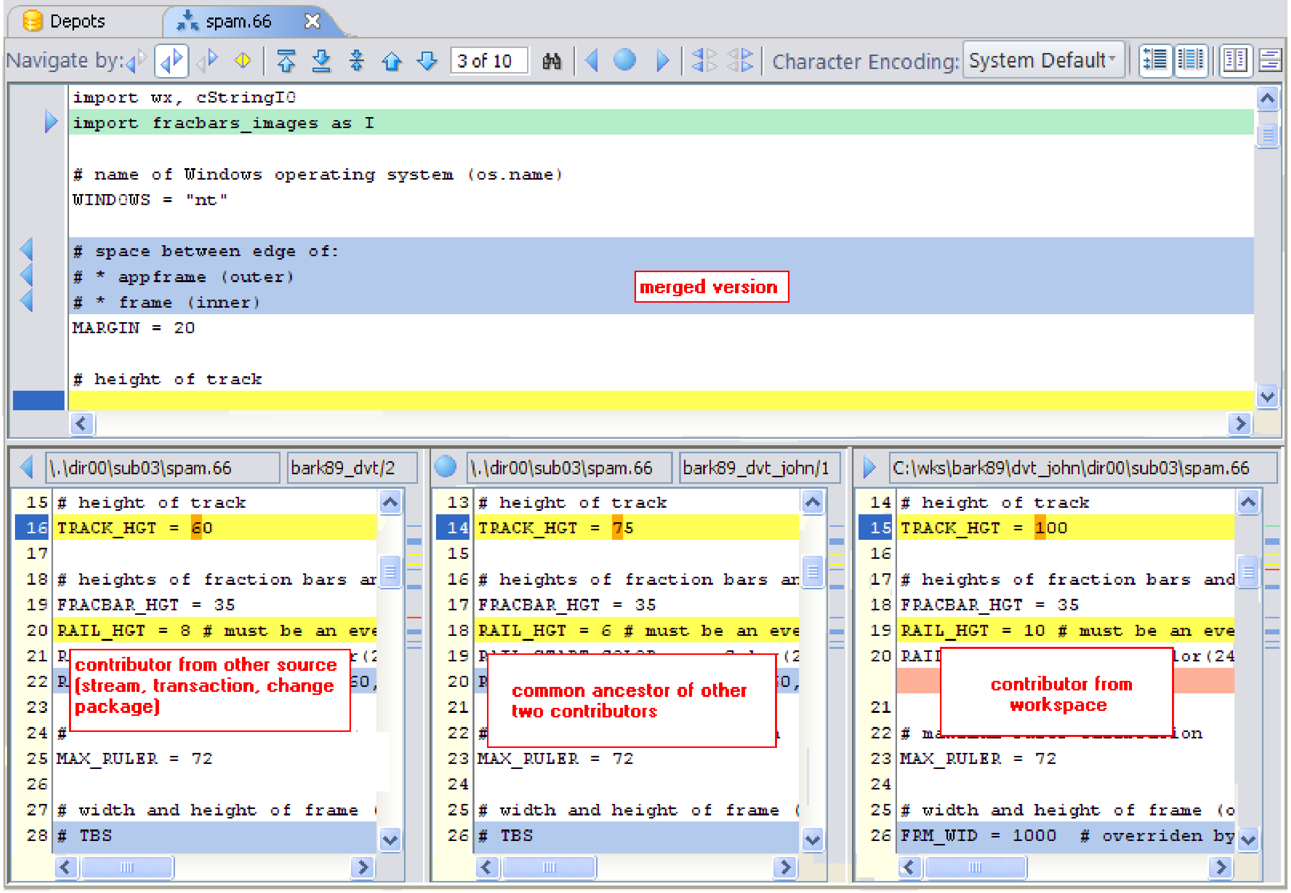
By default, the Merge tool displays the two contributors' contents in side-by-side panes with the contents of the contributors' closest common ancestor in a third pane between them.

Note: Using a toolbar button, you can arrange the contributors' panes vertically, or toggle the visibility of the pane containing the closest common ancestor.

The file in your workspace is always on the right. Above them, the Merge tool displays the merged version. The contributor panes and closest common ancestor pane are synchronized: scrolling any one of them causes the others to scroll, too.

Special cases:

- HTML files
- Binary files



```
import wx, cStringIO
import fracbars_images as I

# name of Windows operating system (os.name)
WINDOWS = "nt"

# space between edge of:
# * appframe (outer)
# * frame (inner)
MARGIN = 20

# height of track
```

merged version

```
# height of track
TRACK_HGT = 60

# heights of fraction bars ar
FRACBAR_HGT = 35
RAIL_HGT = 8 # must be an eve
R                              r(2
R                              60,

#
MAX_RULER = 72

# width and height of frame (
# TBS
```

contributor from other source (stream, transaction, change package)

```
# height of track
TRACK_HGT = 75

# heights of fraction bars ar
FRACBAR_HGT = 35
RAIL_HGT = 6 # must be an eve
R                              (2
R                              0,

#                              n
MAX_RULER = 72

# width and height of frame (
# TBS
```

common ancestor of other two contributors

```
# height of track
TRACK_HGT = 100

# heights of fraction bars and
FRACBAR_HGT = 35
RAIL_HGT = 10 # must be an eve
RAIL                           lor(24


# ma                           ion
MAX_RULER = 72

# width and height of frame (o
FRM_WID = 1000   # overriden by
```

contributor from workspace

Color-Coding of the Contributor Versions

Like the Diff tool, the Merge tool partitions the contributor versions' contents, displaying unchanged sections with a white background and change sections with colored backgrounds. The color-coding scheme is a bit different, because the Merge tool has the more complex job of distinguishing conflicting changes from non-conflicting ones. In addition, the color-coding scheme depends on which navigation mode you're working in.

- For changes that are traversed in the current navigation mode, the Diff tool's color-coding scheme is used.
- Changes that navigating in the current mode skips over are colored *gray*.
- For a conflicting change, the blocks in all contributors are colored *yellow*, no matter what the navigation mode is.

Example 1: In *My changes* navigation mode, non-conflicting additions that come from your version are colored *green*; non-conflicting additions that come from the other contributor are colored *gray*. In *Their changes* navigation mode, the color-coding of non-conflicting additions would be reversed.

Example 2: In *Conflicts* navigation mode, all conflicting changes are colored *yellow*; all other change sections are colored *gray*.

Color-Coding of the Merged Version

The merged version is displayed with color-coding, too. Each colored block is a location where a change has been incorporated from one of the contributors into the merged version. When it starts, the Merge tool performs as much merging as it can automatically, applying all of the non-conflicting changes to the merged version. The text is always colored *gray*. Annotations in the left margin indicate the origin of the change:

- left-hand (or upper) contributor: ◀ annotation
- right-hand (or lower) contributor: ▶ annotation

The Merge tool cannot resolve conflicts automatically, so the location of each conflict is initially indicated in the merged version by a blank *yellow* block. As you make selections to resolve the conflicts, the blank yellow blocks get filled in with the selected text and are annotated:

- blocks that you select from the left-hand (or upper) contributor: ◀ annotation
- blocks that you select from the right-hand (or lower) contributor: ▶ annotation
- blocks that you select from the closest common ancestor: ● annotation

In all cases, the text of a resolved conflict is colored *blue*.

Resolving Conflicts

You can think of the Merge tool as a special-purpose text editor: it combines the contents of two files, producing a third file as the result. Like any text editor, the Merge tool provides you with a set of editing and navigation functions; you have a lot of flexibility in choosing which functions to use, and in which order.

But the Merge tool does enforce this requirement: you must resolve all the conflicting changes between the contributor versions before you can complete a merge operation. Thus, Merge tool usage follows this basic pattern:

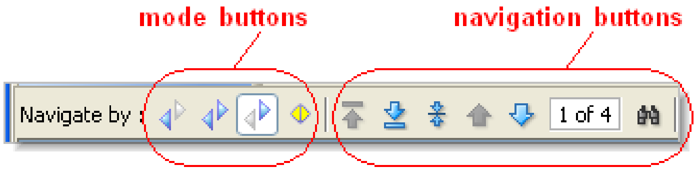
- Navigate to the various conflict sections.
- Resolve each conflict by selecting a section from the left-hand (or upper) contributor, the right-hand (or lower) contributor, or the closest common ancestor. You can also resolve the conflict by selecting the sections from more than one of these sources -- even from all three -- in any order.
- (optional) Add manual edits to the merged version.
- Save the results.

These operations (except for manual editing) are controlled by Merge toolbar buttons, as described in the following sections.

Navigating the Display and the Change Sections

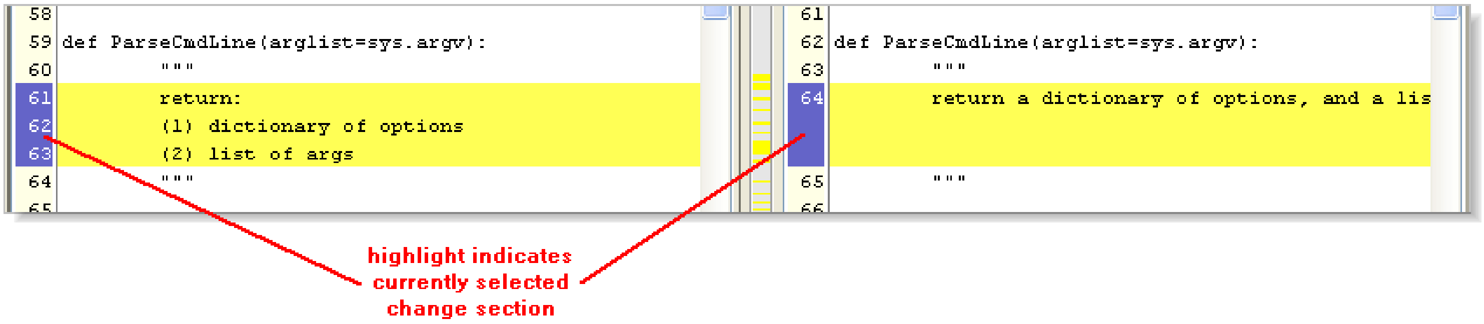
Some of the Merge tool's navigation facilities are the same as the Diff tool's:

- The panes have scroll bars, which are automatically synchronized with each other.
- The navigation buttons in the middle of the Merge toolbar work the same way as in the Diff tool, with a slight variation:



  - If the *Their changes* mode button is selected, the navigation buttons traverse just the change sections in which the left-hand (or upper) contributor differs from the closest common ancestor.
  - If the *All the changes* mode button is selected, the navigation buttons traverse all the change sections.
  - If the *My changes* mode button is selected, the navigation buttons traverse just the change sections in which the right-hand (or lower) contributor differs from the closest common ancestor.
  - If the *Conflicts* mode button is selected, the navigation buttons traverse just the sections with conflicting changes.

At any given time, one of the change sections is currently selected. The line numbers in both contributor panes are highlighted to indicate the selection.



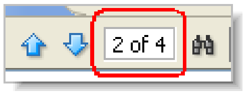
```
def ParseCmdLine(arglist=sys.argv):
        """
        return:
        (1) dictionary of options
        (2) list of args
        """
```

```
def ParseCmdLine(arglist=sys.argv):
        """
        return a dictionary of options, and a lis

        """
```

Whenever you use one of these buttons (except for *Search*) to jump to a particular change section, the Merge tool remembers it as the current change. Selecting a particular change section by left-clicking it in either contributor pane makes it the current change.

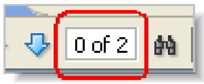
The Change Section Counter

As you navigate among the change sections, a counter in the Merge toolbar tracks your location, taking into account which navigation mode is selected.



For example, "2 of 4" might mean "you are currently in *My changes* mode, and you are currently at the 2nd of 4 such changes".

The counter typically changes as you switch navigation modes.

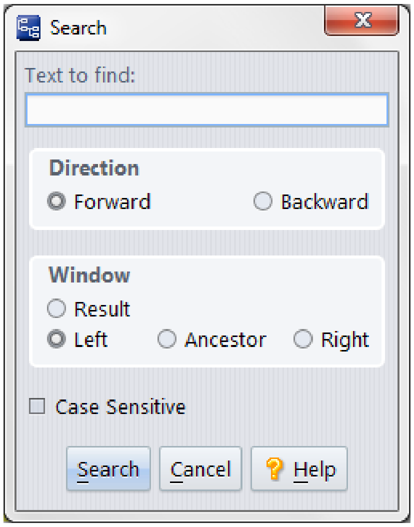


In some cases, the first number may be "0". This occurs when:

- You've just switched modes, and the current change section is not selected by the new mode.
- In Conflicts mode, you've just resolved a conflict, by selecting a section from one contributor. The current change section is no longer a conflict, so it's no longer selected by your current navigation mode.

Searching for Text

In addition to navigating among the change/conflict sections, you can search for any text string in any pane, using the *Search* toolbar button (keyboard shortcut: **Ctrl-S**).



Search

Text to find:

Direction

● Forward ○ Backward

Window

○ Result

● Left ○ Ancestor ○ Right

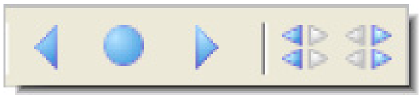
□ Case Sensitive

You can also invoke this command from the context menu that appears when you right-click anywhere in either contributor pane.

Selecting a Section from a Contributor (or Multiple Contributors)

Sometimes, the Merge tool can construct a merged version without any help from you. This occurs if all the change sections are non-conflicting. The Merge tool just applies all the changes to the merged version and announces that it's done. This section describes the more interesting case: the contributors have one or more conflicts, which *you* must resolve.

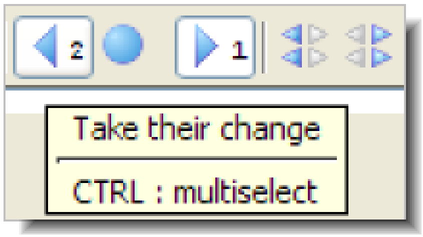
The Merge tool automatically jumps to the first conflict. That is, it scrolls to the first yellow-highlighted change section in the contributors, and the corresponding blank yellow block in the merged version. It's up to you to decide which of the changes is to be incorporated into the merged version. You indicate your decision using the conflict-resolution toolbar buttons.



- Click the ● *Take common ancestor change* toolbar button to incorporate the original block from the closest common ancestor.

- Click the ▶ *Take my change* toolbar button to incorporate the changed block from the right-hand (or lower) contributor (file in your workspace).
- You can take two of the blocks or all three of them, in any order:

1. Click the ◀ o●r ▶ or conflict-resolution button to select one block.



2. Hold down the **Ctrl** key, and click another conflict-resolution button. The newly selected block is sent to the merged version, below the previously selected block. The conflict-resolution buttons are annotated with numbers, indicating the order you selected. (And the tooltip reminds you of this "multiselect" capability.)
3. Optionally, repeat the preceding step to include the third block, as well.

You can undo a multiselect by clicking any of the conflict-resolution buttons *without* holding down the **Ctrl** key. Typically, you'll need to do some manual editing to "smooth out" the combined changes of a multiselect.

Note: You cannot undo a multiselect once you've performed a manual edit.

The following two conflict-resolution buttons make large-scale changes -- use them carefully! They both effectively override the Merge tool's automatic inclusion of non-conflicting changes into the merged version.

- The *Revert all of my changes* button makes the merged version the same as the left-hand (or upper) contributor version.
- The *Use only my changes* button makes the merged version the same as the right-hand (or lower) contributor version.

When you click one or more conflict-resolution buttons to resolve a conflict, the selected block(s) appear in the merged version and the highlight changes from yellow **(unresolved)** to blue **(resolved)**. This also decrements the change section counter in the Merge toolbar.

Use the ⇩ *Next* navigation button to scroll down to the next conflict. (It may take more than one click if your are not in *Conflicts* navigation mode.) Again, use one of the "take change" buttons to select the text from the left contributor or the right contributor (or both of them).

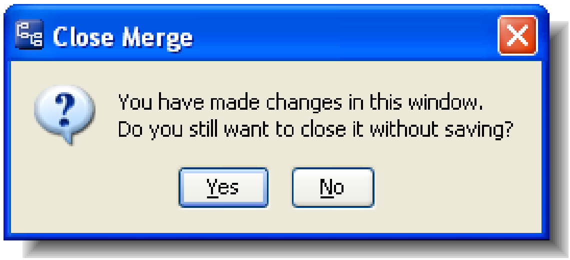
You don't have to resolve the conflicts in order. You can jump around as much as you want among the change sections, or scroll through unchanged sections to look up information the affects your merge decisions.

Changing Your Mind

You can change your merge decisions as much as you want. For example, you can navigate to a particular change section where you had selected ◀ *Take their change,* and then click the ▶ *Take my change* button. This swaps out their change and swaps in your change.

Warning: swaps of this kind also discard any manual edits that you have made in that section after initially selecting it.

If you want to discard *all* your merge work, close the Merge tab to abandon the merge session entirely.



You have made changes in this window.
Do you still want to close it without saving?

A confirmation dialog box appears, making it less likely that you'll discard your merge work accidentally.

Manual Editing

At any time during a Merge tool session, you can manually edit the contents of the merged version. Just click anywhere in the pane containing the merged version, and type. The **Delete** and **Backspace** keys work as expected. Using context (right-click) menus or the GUI main menu, you can *Cut, Copy,* and *Paste* sections of text that you've highlighted with the mouse. You can also use the common keyboard shortcuts: **Ctrl-C** or **Ctrl-Ins** to copy, **Ctrl-V** or **Shift-Ins** to paste.

Warning: If you make changes manually within a conflict section that you've resolved with one of the conflict-resolution buttons, the manual changes will be overwritten if you return to that section and click any conflict-resolution button.

Joining Change Sections

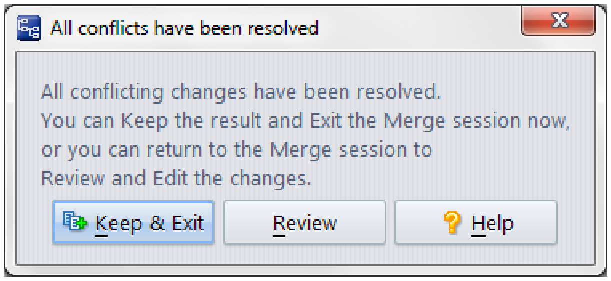
Sometimes, there is a block of lines in a contributor version that you consider to be a single change section, but that the Merge tool decides are two separate sections. You can combine the two change sections:

1. Right-click in the contributor section, and choose *Join Sections* from the context menu.
2. The mouse pointer changes to . Click the adjacent section to be joined with the first section.

The Merge tool joins the sections, then readjusts its analysis of the contributors' change sections. Typically, this changes the number of conflict sections.

Saving the Merged Version

When you resolve the last remaining conflict by clicking one of the conflict-resolution buttons, the Conflicts counter goes to "0 of 0" and the Merge tool displays this message window:



All conflicting changes have been resolved.
You can Keep the result and Exit the Merge session now,
or you can return to the Merge session to
Review and Edit the changes.

- Clicking *Keep & Exit* ends the merge session immediately, closing the Merge tab. AccuRev overwrites the file in your workspace with the merged version. Then, it a *Keeps* a new version in your workspace stream, to preserve the merge results. The overlap status of the file is removed, so that you can now *Promote* the new version to the backing stream.

- Clicking *Review* continues the merge session. This enables you to review your work, change some of your merge decisions, and perform manual edits. At any time, you can click the *Keep* button in the Merge toolbar to complete the merge process.

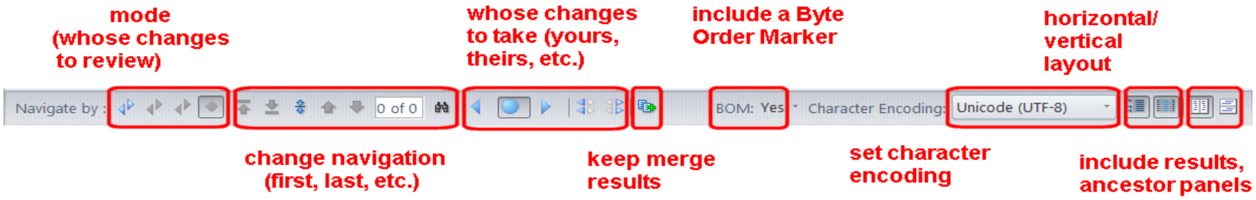
You can also cancel the entire merge during a review pass by closing the Merge tab without performing a *Keep*.

Notes About Merge Behavior

As of AccuRev Release 4.9, the *merge* command is now smarter about simple (rebase) merges, resulting in simplified display of merge lines in the Version Browser, and simplified change packages. For example, assume two developers are working in parallel on the same element, with a common ancestor, and there are only regular versions (that is, no patches, merges, or reversions) between the current versions and the common ancestor. When these are merged, the resulting version of the merge is rebased to point to the merged version.

In these cases you will not see a red merge line in the Version Browser. Instead, the comment section of the merged file will contain an entry similar to: "`merged, rebased to <version>`". In other words, if a merge results in a change that you would have received if you had simply done an **update**, it is not displayed as a merge. See the screenshots under Notes: on page 234.

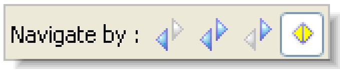
Commands Available in a Merge Tab



BOM: Yes

Character Encoding: Unicode (UTF-8)

Navigation Mode Radio Buttons



Their changes

When this radio button is selected, the navigation buttons take you to the change sections in which the left-hand (or upper) contributor differs from the closest common ancestor.

All the changes

When this radio button is selected, the navigation buttons take you to *all* the change sections.

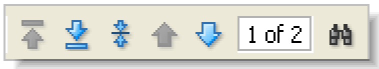
My changes

When this radio button is selected, the navigation buttons take you to the change sections in which the right-hand (or lower) contributor differs from the closest common ancestor.

Conflicts

When this radio button is selected, the navigation buttons take you to *all* the sections that contain conflicting changes.

Navigation Buttons



First change/conflict

Go to, and select, the first change section (or unresolved conflict section)

Last change/conflict

Go to, and select, the last change section (or unresolved conflict section)

Center current change/conflict

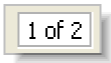
Scroll the contributor panes (if necessary), to make the currently selected change section visible.

Previous change/conflict

Go to, and select, the preceding change section (or unresolved conflict section)

Next change/conflict

Go to, and select, the next change section (or unresolved conflict section)

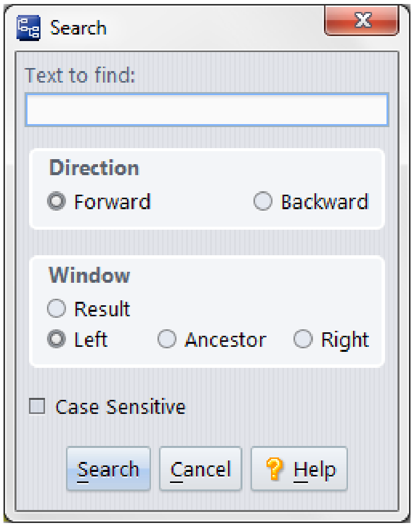


Change section counter

Display only, not a button. See *The Change Section Counter* on page 219.

Search

Bring up a dialog to define a search in one of the panes for a specified text string.



Search

Text to find:

Direction

● Forward ○ Backward

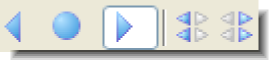
Window

○ Result

● Left ○ Ancestor ○ Right

☐ Case Sensitive

Conflict Resolution Buttons



Invoking a "Take ... change" operation replaces a previously chosen change section at that location. It also wipes out any manual edits you may have performed in that change section.

Take their change

Copy the currently selected change section in the left (or upper) contributor pane to the results pane.

Take common ancestor

Copy the original block from the closest common ancestor to the results pane.

Take my change

Copy the currently selected change section in the right (or lower) contributor pane to the results pane.

Revert all my changes

Make the merged version the same as the left-hand (or upper) contributor version.

Use only my changes

Make the merged version the same as the right-hand (or lower) contributor version.

The "Keep merge results" Button

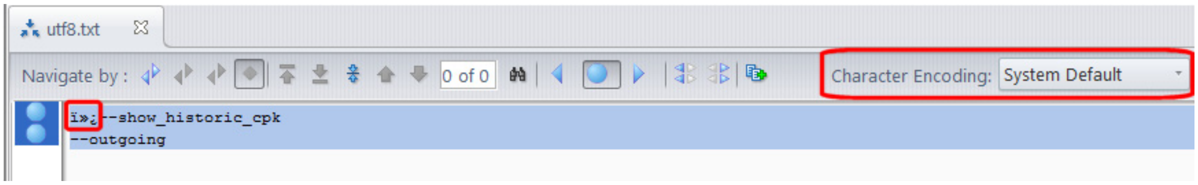
Keep results and close

Preserve the current contents of the merge results pane as a new version of the element, using the **Keep** command. (The **Keep** dialog box does not appear.) The Merge tab for this file then closes automatically.

Character Encoding Fields

The Merge view has two fields related to character encoding, **Character Encoding** and **BOM**:

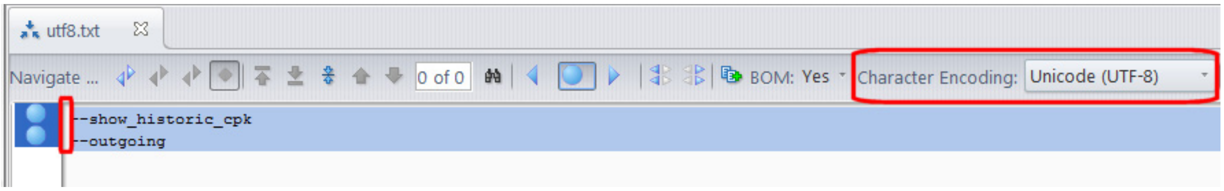
- **Character Encoding**: A drop-down list that lets you specify the character encoding of the file being merged. This field has two effects:
  - It renders the file using the encoding specified. For example, a file whose encoding is set to UTF-8 is rendered like this when the Character Encoding is set to System Default:



Character Encoding: System Default

```
i»¿--show_historic_cpk
--outgoing
```

And it is rendered like this when the Character Encoding is set to Unicode (UTF-8):



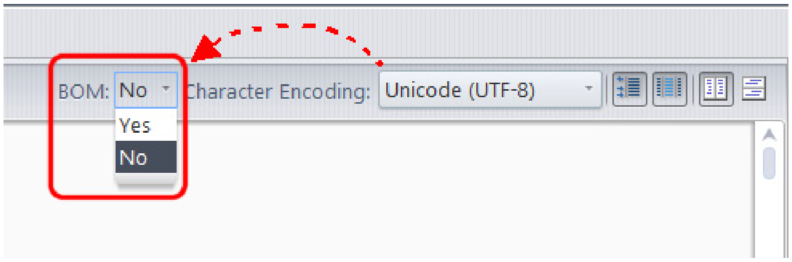
BOM: Yes

Character Encoding: Unicode (UTF-8)

```
--show_historic_cpk
--outgoing
```

  - It sets the encoding of the merged output. So, for example, if the Character Encoding is set to Unicode (UTF-8), the file is saved with UTF-8 encoding when the merge is complete.

- **BOM**: A BOM (Byte Order Mark) is part of the Unicode standard used to indicate the endianness of a text file, and which Unicode encoding is in use (UTF-8, UTF-16, or UTF-32. for example). The **BOM** drop-down is displayed only when you are merging Unicode files, as shown here:



BOM: No

Character Encoding: Unicode (UTF-8)

This can happen because the native encoding is Unicode, or if you use the **Character Encoding** field to explicitly set a Unicode encoding.

When the **BOM** field is set to Yes, the output file of the merge will be written with a leading BOM indicator from the Unicode standard. When the **BOM** field is set to No, the output file will be written without the leading BOM. If either contributor to the merge has a BOM in it, the **BOM** field defaults to Yes; otherwise it defaults to No.

In most cases, you should not write the Byte Order Mark to the output, unless you have a specific need for it (because the file is used externally by a tool that requires the BOM).

Pane Selection Buttons

Show results of edits

Toggle the visibility of the merged version pane. This enables you to see more data in the two contributor panes.

Show HTML result

(Appears for file with *.htm* or *.html* suffix only) Toggle switch: display the context of the merged version as plain text, or render the contents as HTML.

Show Ancestor Panel

Toggle switch: display/hide the closest common ancestor pane.

Horizontal Layout

(default) Places the "other" contributor pane to the left of the "workspace" contributor pane.

Vertical Layout

Places the "other" contributor pane above the "workspace" contributor pane.

The Search Command

The *Search* command in the AccuRev Diff tool or Merge tool searches for a specified character string in one of the tool’s panes -- workspace contributor (lower right), stream contributor (lower left), ancestor (center), or results (top).

Using the Search Command

To use the Search command in the Diff or Merge tool:

1. Click the 🔍 button in the Merge toolbar.

   Alternative: Type Ctrl+F.

   The Search dialog box appears.
2. Type a character string in the **Text to find** entry field.
3. Make sure the **Direction**, **Window**, and **Case Sensitive** options are set as you want them.
4. Click the **Search** button.

Patch and Patch From

Patch From (or **Patch**) perform a **patch** operation on a text-file element, incorporating *some* of the changes in a specified "patch from" version into the version in your workspace. The combined contents are saved, with **Keep**, as a new version in your workspace.

The patch operation is operationally similar to a **merge** operation, which incorporates *all* of the changes in a specified version. In fact, both operations use the same tool, either AccuRev's Merge tool or a third-party text-file-merge tool.

(An advanced extension of the patch concept called change packages, which integrates AccuRev's SCM functionality with AccuWork's issue management system is discussed in *Chapter 10 Change Packages*.)

Structure of a Patch

Patch creates a new version in a workspace. This version is termed the **head version** of the patch. AccuRev automatically determines a corresponding **basis version** by scanning backward through the element's ancestry. The basis version is the most recent version that was either:

- originally created in another workspace
- created in the same workspace, then was promoted to the backing stream

  Note: As of AccuRev 6.0, in certain circumstances users can specify the basis version, thanks to the introduction of *multi-segments*.

Patch vs. Merge

When you merge version *v* of a file into your workspace version, you are saying:

> "combine my file with version *v*, taking into account all the changes in version *v*, back to the common ancestor"

When you patch version *v* of a file into your workspace version, you are saying:

> "combine my file with version *v*, taking into account only the recent changes to version *v*"

To find the "recent changes to version *v*", AccuRev scans backward through the file's ancestry, starting at version *v* and stopping when it encounters ...

> a version that was originally created in another workspace
>
> ... or ...

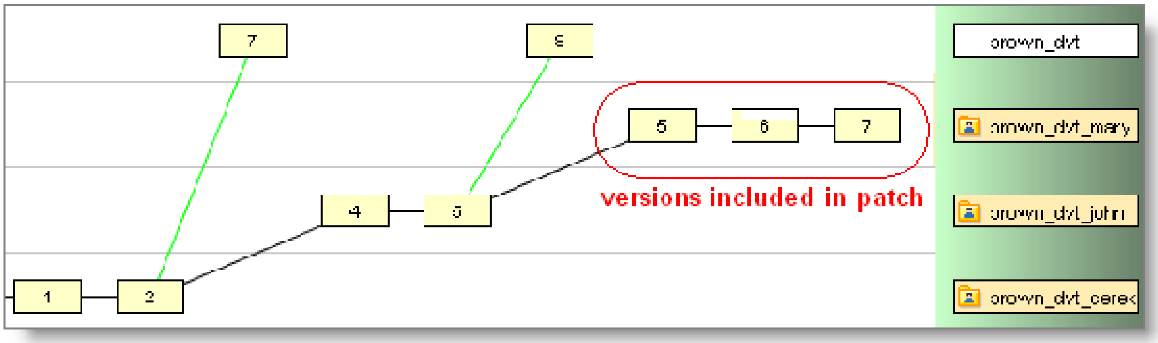
a version that was promoted to another stream

This backward search defines a patch to the element. Version *v* is termed the *head version* of the patch. The older version is termed the *basis version* of the patch. The patch consists of all the versions of the element between the basis version and head version. The basis version itself is not included in the patch -- it precedes the set of "recent changes" in version *v*. The head version is included in the patch -- it contains the latest of the recent changes.

In identifying versions "between" the head and basis versions, AccuRev follows both direct ancestor and merge ancestry lines. In particular, if version *v* was created by a merge, its "recent changes" include both contributors to the merge.

Example 1

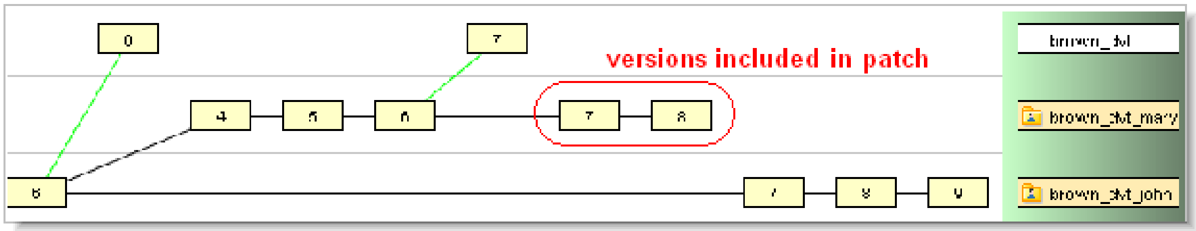
Before Mary started her recent work, she updated her workspace. This brought in a version of the file originally created by another user -- say, version *brown_dvt_john/5*. Then she proceeded to create versions 5 through 7 in workspace *brown_dvt_mary*.



Derek decides to patch version *brown_dvt_mary/7* into his workspace. He invokes the *Patch From* command from the context menu of this version. AccuRev searches backward through the element's ancestry, and includes the set of changes recently made in Mary's workspace: this set includes *brown_dvt_mary/5* through *brown_dvt_mary/7*; the patch doesn't include the two versions created in workspace *brown_dvt_john*.

Example 2

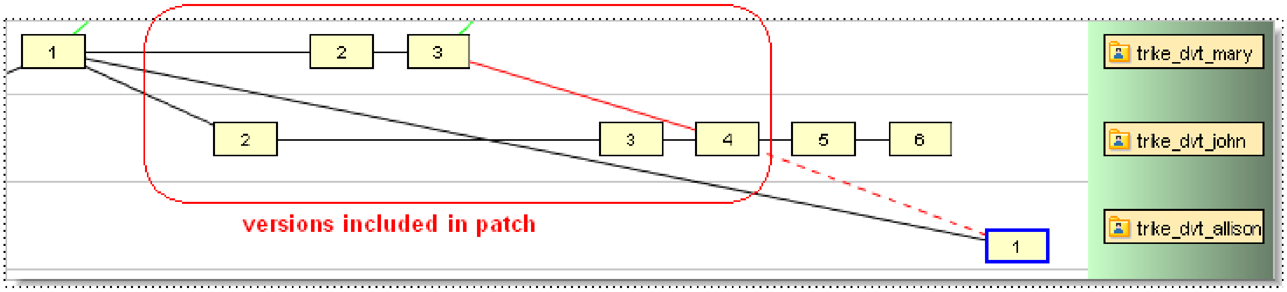
Mary started a task by bringing version *brown_dvt_john/6* into her workspace with an update. Then she created versions *brown_dvt_mary/4* through *brown_dvt_mary/6*, promoted her work to the backing stream, and then created two more versions: *brown_dvt_mary/7* and *brown_dvt_mary/8*.



When John patches version *brown_dvt_mary/8* into his workspace, AccuRev decides that only the versions since the promotion — versions 7 and 8 — contain "recent changes". The idea is that a promotion typically marks the end of a programming task, not an intermediate checkpoint.

Example 3

Allison patches version *trike_dvt_john/4* into her workspace, creating version *trike_dvt_allison/1*. Because John's version was created by a merge, the patch includes the recent changes to both merge contributors.



versions included in patch

Invoking the Patch by Drag & Drop

If you have change packages enabled and use the classic GUI rather than the CLI, the most straightforward way to initiate patch operations (if needed) is to drag and drop issues in the StreamBrowser, from an Active Issues display to the destination stream or workspace.

AccuRev determines whether the changes can be automatically merged, and if not, invokes tools such as the Change Palette or a merge display to complete the necessary operations.

Following is a summary of the behaviors you will see in different scenarios.

Dragging to a Workspace

When you drag one or more issues to a workspace, AccuRev behavior depends upon the following considerations:

- If you drag a single issue to a workspace, AccuRev determines whether or not a manual intervention is required, and either offers to auto-merge, or brings up the merge display
- If you drag one or more issues to a workspace, and AccuRev determines that they do not require a patch, it automatically performs a **Send to Workspace** operation.
- If you drag one or more issues to a workspace and AccuRev determines that merging or patching is required, it displays the Change Palette in issue mode (see *Working with the Issue Change Palette* on page 259).

  Note: If you attempt to patch into a workspace where a target element has unkept changes (has **(modified)** status), AccuRev alerts you and asks whether you wish to overwrite these unkept changes, or cancel the patch operation:

  ```
  You have made changes to some elements in this workspace.
  Do you want to discard these changes and overwrite those elements?
  Yes/Cancel
  ```

Dragging to a Dynamic Stream:

When you drag one or more issues to a dynamic stream, AccuRev behavior depends upon whether or not a patch is required:

- If a patch is required, AccuRev displays the Change Palette in issue mode
- If a patch is not required, AccuRev displays the **Promote** comment dialog box and prompts you through the **promote** process

Invoking the Patch From Command

You can invoke **Patch From** on a selected version in the following contexts:

- In the **Version Browser** (when invoked from a workspace), to patch from an arbitrary version.
- In the **History Browser** (when invoked from a workspace), to patch from the version in a particular transaction.

AccuRev processes the files you specify one-by-one. For each file, it does one or both of the following:

- Prompts you, in one or more dialogs, to resolve namespace-level conflicts.
- Invokes AccuRev's Merge tool or the other tool you've configured, in order to perform the patch operation. AccuRev's Merge tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

Notes:

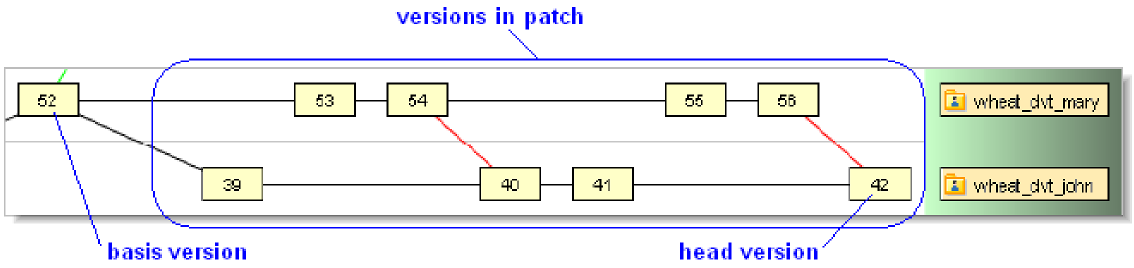
Patches involve real versions, not virtual versions

AccuRev's patch facility involves working with the contents of text files. And AccuRev records changes to files' contents as real versions in workspaces. Accordingly, AccuRev only uses real versions in its patch manipulations. If you specify a virtual version in some dynamic stream as the "patch from" version, AccuRev automatically uses the real version for which the virtual version is an alias.

What it means to be "between" the head and basis versions

In the simplest case, a patch's head version was derived from its basis version by a series of **Keep** operations (and maybe a **Rename**). In this case, it's easy to determine the set of versions that are between the head and basis versions. But what if one or more of the versions was created with **Merge**? Here's an example:

What if we **patch from** version *wheat_dvt_john/42?*



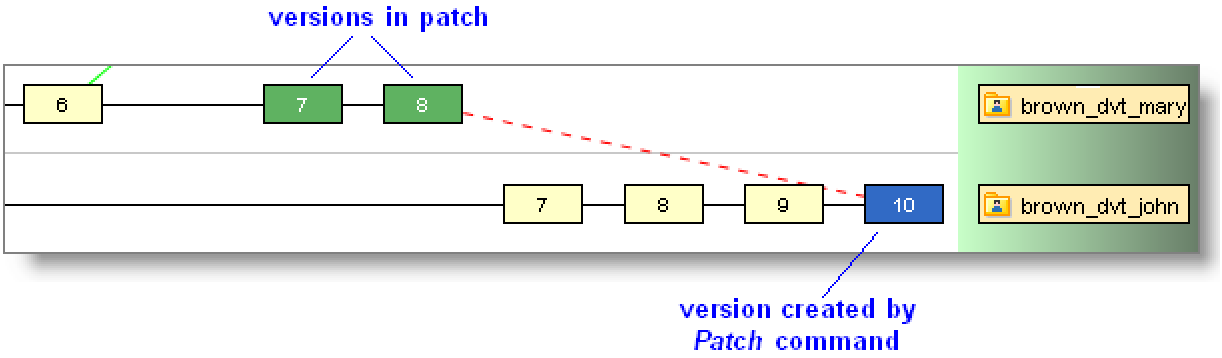
The basis version corresponding to this head version is the most recent one created in another workspace and brought into workspace *wheat_dvt_john* by an update -- that is, version *wheat_dvt_mary/52*. What versions are "between" these two versions? AccuRev determines the answer by finding this set difference:

element's version graph, up to and including **head version** — element's version graph, up to and including **basis version**

In this case, the difference between the two subsets of the element's version graph consists of eight versions, from two different workspaces.

Patch Display

In the *Version Browser*, a version created by *Patch* is connected to the "patch from" version with a dashed red line. Selecting this version highlights the versions in the patch.



The Patch Algorithm

The ***Patch From*** command performs its content-level work with the Merge tool. Submitting a different set of versions to this tool effectively implements the patch algorithm (see *The Merge, Patch, and Reverse Patch Algorithms* on page 232).

Show Patch List

The **Show Patch List** operation lists all the individual versions whose changes need to be sent from one stream (or snapshot, or workspace) to another, so that the second stream will include all the changes in the first stream.

Notes:

- *Comparison with **Stream Diff***—You can think of the Patch List tab as providing an "exploded view" of the Stream Diff (Files) tab. Whereas Stream Diff provides a simple "before" and "after" perspective on how each element changed, **Patch List** shows all the individual steps (that is, the versions) in progressing from the "before" state to the "after" state.
- *Patch List and the **Patch** command*—A Patch List table with 24 entries does not mean that you would need 24 invocations of the **Patch** command to propagate all the changes from one stream to the other. Each invocation of Patch incorporates all the changes in the specified patch A selected set changes (typically, the 'recent changes' made by one user) to a text-file element. Also, the merge-like operation that incorporates those changes in another version of the same element. See merge, basis version, head version, change package, reverse patch, which can include multiple versions. (Prior to AccuRev 3.5.5, each invocation of the **Patch** command did incorporate the changes in one version, so the correspondence to the Patch List table was exact.)

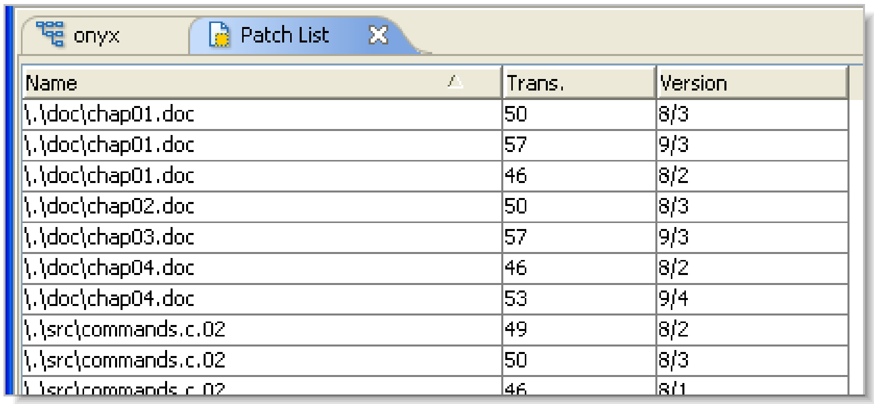
Opening a Patch List Tab

1. In the **StreamBrowser**, select a stream, snapshot, or workspace.
2. Choose **Show Patch List** from the selection's context menu, or click the button on the StreamBrowser's toolbar.

3. The mouse pointer changes to . Left-click another stream, snapshot, or workspace.

Patch List Tab Layout

The Patch List tab contains a table, each row of which details one version.



Name	Trans.	Version
\.\doc\chap01.doc	50	8/3
\.\doc\chap01.doc	57	9/3
\.\doc\chap01.doc	46	8/2
\.\doc\chap02.doc	50	8/3
\.\doc\chap03.doc	57	9/3
\.\doc\chap04.doc	46	8/2
\.\doc\chap04.doc	53	9/4
\.\src\commands.c.02	49	8/2
\.\src\commands.c.02	50	8/3
\.\src\commands.c.02	46	8/1

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.
Notaries is the element's pathname in the first stream at the current time. This may not match its pathname at the time of the transaction listed in the Transaction column.

Transaction

The transaction in which this version was created.

Version

The version-ID of the version

Working in a Patch List Tab

The Patch List tab display is informational only. See also:*Working with Tables* on page 14

The Merge, Patch, and Reverse Patch Algorithms

The AccuRev GUI uses the same tool to perform interactive merge operations, interactive patch operations, and interactive reverse patch operations (**Revert**) on the contents of a text-file element. In all these operations:

- AccuRev analyzes two contributor versions of a file, along with a third version designated as the closest common ancestor of the two contributors.
- The two contributor versions' contents are combined to produce a new version of the file, which is saved in the repository with a **Keep** command. Sometimes, the new version can be produced completely automatically; other times, you have to interactively resolve conflicts between the contributor versions.

The only differences among the several operations are in which versions are designated to be the contributors and the closest common ancestor. These differences are detailed below.

Notes:

- *Using a third-party tool for merge, patch, and reverse patch operations*

  By default, merge, patch, and reverse patch operations are performed by AccuRev's own Merge tool. If you configure a third-party text-file-merge tool, it will be used for all these operations. The algorithm used by the third-party tool is not necessarily the same as that used by AccuRev's Merge tool.

- *Merging, patching, and reverting namespace changes*

  Any merge, patch, or reverse patch operation can involve namespace changes in addition to (or instead of) content changes. AccuRev compares the pathnames of the two contributor versions, and in some cases applies a pathname change with the *Rename* command. If both a **Rename** and a **Keep** are performed, the **Rename** transaction comes first.

  See *Resolving Namespace Conflicts* on page 238 for more details.

Merge: Incorporating Content Changes

In a **Merge** or **Merge From** operation, the analysis performed on the two contributor versions goes beyond a simple diff: AccuRev determines how each *difference* represents a change from the closest common ancestor version. (To emphasize the Merge tool's perspective, we use the term “change section” to describe a location where the contributors differ from each other. In the context of the Diff tool, we use the term “difference section”.)

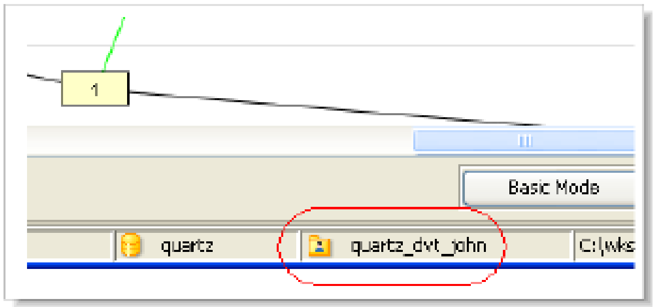
Versions in a 3-Way Merge

The versions of an element that figure in AccuRev's *3-way merge* algorithm are:

1. Workspace version (appears in lower right pane of Merge tool)

   This is also called the "to" version. The merge results will be saved as a new version in some workspace, replacing the version displayed in this pace. Which workspace is the new version created in?

   - In the most common merge scenario, you invoke **Merge** in a File Browser open on a particular workspace. The new version is created in that workspace.
   - You can invoke **Merge From** or **Patch From** in any of the version tools.

   - In this case, the new version is created in the workspace from which you launched the version tool. It's listed in the status bar at the bottom of the GUI window.
   - If you invoke **Merge** from the Change Palette, you are prompted to specify a workspace in which to perform the merge. The new version is created in that workspace.

2. Stream version (appears in lower left pane of Merge tool)

   This is also called the "from" version.

   - In the most common merge scenario, this is the version in the workspace's backing stream.
   - If you're using one of the version tools, this is the version on which you invoked **Merge From** or **Patch From**.
   - In the Change Palette, this is the version in the "source stream".

3. Closest common ancestor version (not displayed in Merge tool)

   AccuRev takes into account previous merge operations, but not previous patch operations, in determining the closest common ancestor of the workspace version and the stream version.

Notes:

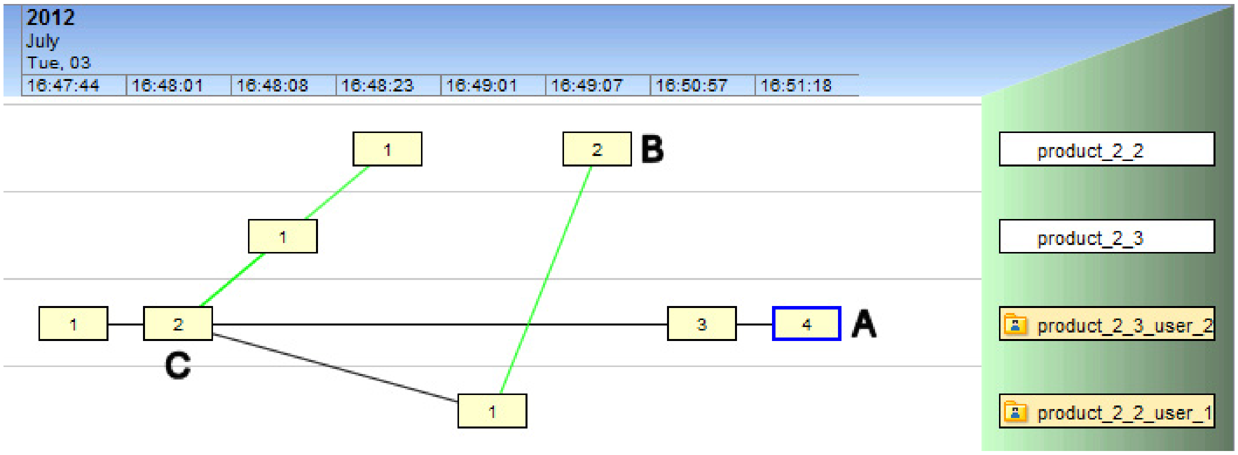
- *Workspace File vs. Workspace Version*

  AccuRev always uses the file in the workspace tree as the first contributor. If you've just saved the file with **Keep**, the file in your workspace is identical to the most recent version in your workspace stream. But if you've edited the file without Keeping it, there's a difference.

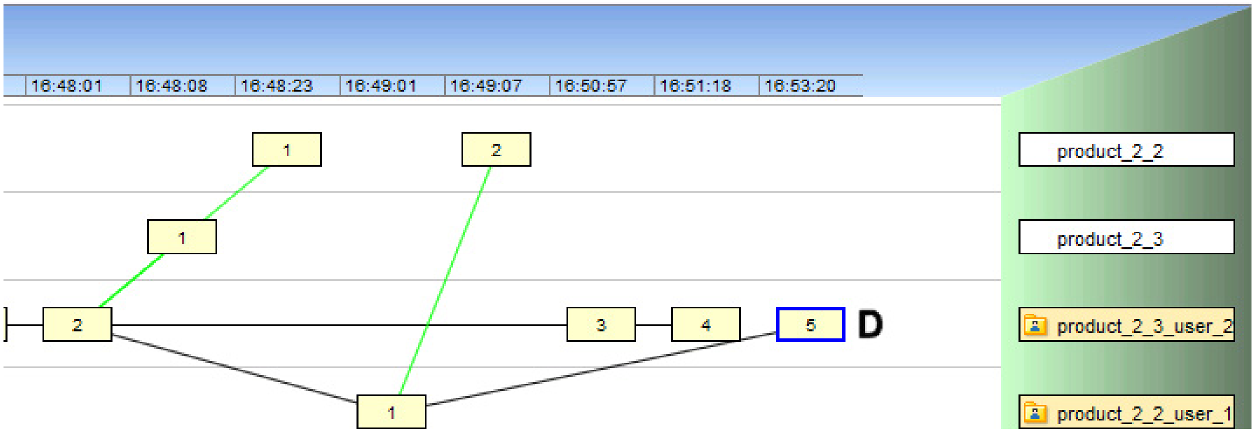
  In all cases, the version currently in the workspace stream is used in the determination of the closest common ancestor version.

- *How can I tell which is the closest common ancestor? How will the results of the merge appear in the Version Browser?*

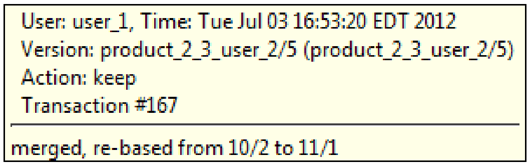
  In the Version Browser, you can identify the common ancestor by visual inspection. In the AccuRev CLI, use the command **anc -c**. The results of the merge will appear differently in the Version Browser if you are merging with a version from the backing stream than they will if you merge with a non-backing stream version. Consider the following examples:



The example above shows the most common case: merging the file with your modifications in your workspace (A) with a version with different changes in the backing stream (B). The closest common ancestor for A and B is C. This is considered a “rebase” merge, and the result appears as follows:

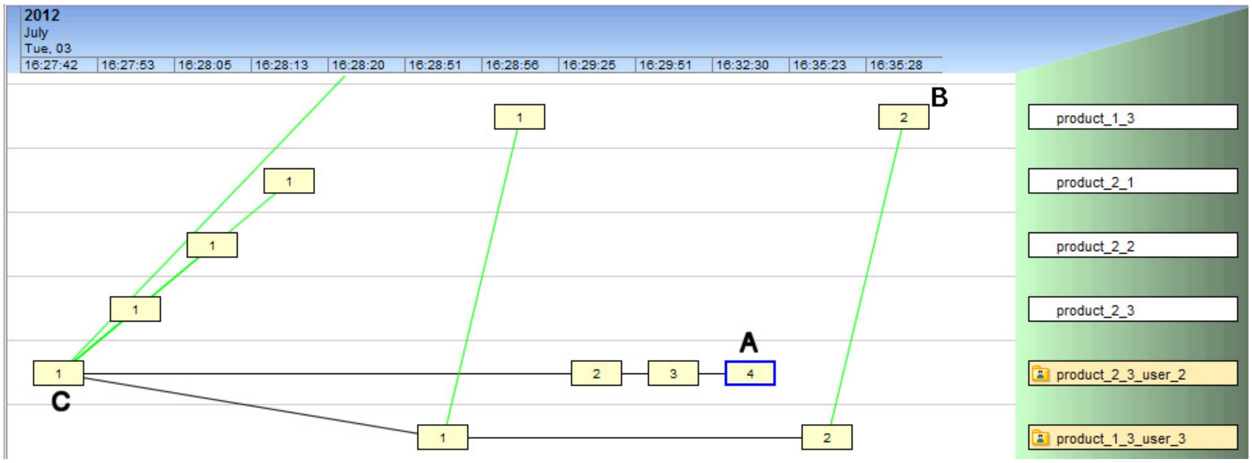


A new version is created (D) for the merged content. Note that in this case no red merge line is displayed. Instead, the merge is noted in the in the comments and in the tooltip which appears when you hover over the resulting version at D.

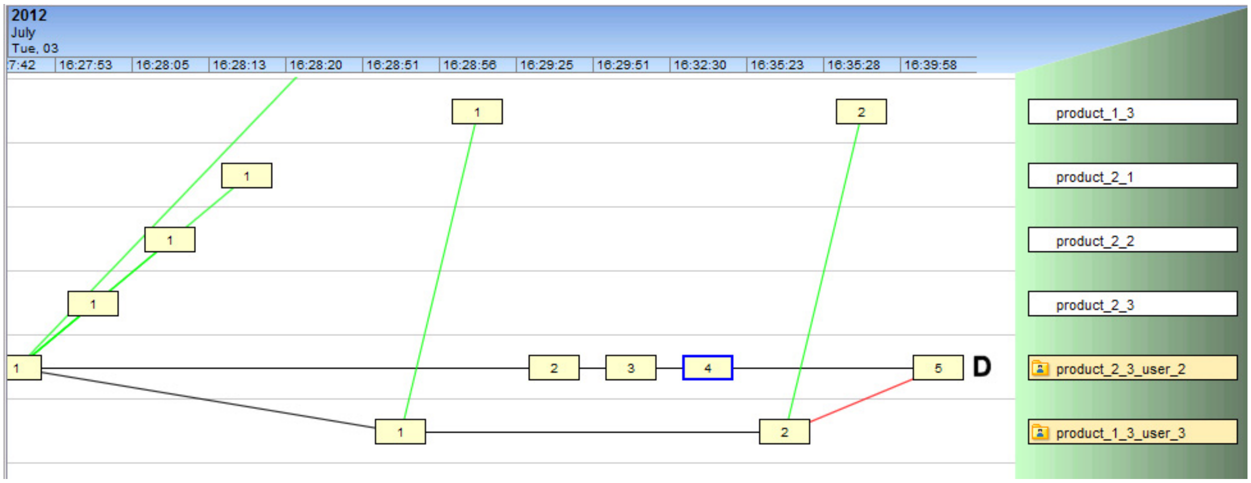

User: user_1, Time: Tue Jul 03 16:53:20 EDT 2012
Version: product_2_3_user_2/5 (product_2_3_user_2/5)
Action: keep
Transaction #167

merged, re-based from 10/2 to 11/1

If you merge with a version that is not in the backing stream, the common ancestor is still easy to determine:



Again, the file with your modifications in your workspace is at (A). The version with different changes in a different stream is at (B). The closest common ancestor for A and B is C.



And again, the result of the merge is a new version shown at D. But note that for this more complex merge, AccuRev indicates the merge process with a red line. (In versions prior to 4.9.1, all merges -- including rebase merges-- were displayed with this red line.

Handling Change Sections

AccuRev processes each change section by comparing (1) the workspace version's content, (2) the stream version's content, the closest common ancestor's content:

- **Non-conflicting change**: Only one contributor — either the 'from' version or the 'to' version — changed the content in the section; the other contributor didn't make a change. AccuRev automatically includes the change in the merged version.
- **Conflicting change**: Both contributors changed the content in the section. AccuRev highlights the section in yellow; you must resolve the conflict by selecting one contributor's change to be included in the merged version.
- **Identical change**: Both contributors changed the content in the section *in exactly the same way.* AccuRev doesn't flag this section at all; the agreed-upon change is automatically included in the merged version.

Example 1: non-conflicting change

Suppose a change section consists of 13 lines that occur in contributor #2 but not in contributor #1. To determine what kind of change this represents, the Merge tool looks at the corresponding location in the closest common ancestor version:

- If those 13 lines exist in the ancestor version, the Merge tool concludes that a change was made in contributor #1 (the lines were deleted) but no change was made in contributor #2 (the lines are still there).
- If the 13 lines do not exist in the ancestor version, the Merge tool concludes that a change was made in contributor #2 (the lines were added) but no change was made in contributor #1 (nothing was added).

In both these cases, there was a change from the common ancestor in exactly one of the contributors. The Merge tool deems this a "non-conflicting change". It incorporates the change (be it an addition, a deletion, or a revision of existing text) into the merged version.

Example 2: conflicting change

Let's take another example. A one-line error message has a slightly different wording in the two contributors:

- Contributor #1:

  ```
  #define E_COLOR498 "No color with that name was found."
  ```
- Contributor #2:

  ```
  #define E_COLOR498 "Color name unknown."
  ```

  The following line occurs at the corresponding location in the closest common ancestor version:

  ```
  #define E_COLOR498 "Huh?"
  ```

  In this situation, the Merge tool finds a change from the common ancestor in *both* contributors, not just one of them. This is a "conflicting change" (or more simply, a "conflict"). The Merge tool doesn't try to decide which contributor's change is better. It just makes it easy for *you* to make this decision when you perform the merge.

Example 3: identical change

It sometimes happens that both contributors have made *the same* change from the common ancestor version. For example, both contributors might have replaced this error message:

```
Huh?
```

with this one:

```
No such color
```

The Merge tool does not identify this as a difference section, because there's no difference between the two contributors. It silently incorporates the agreed-upon change into the merged version.

Patch: Incorporating Content Changes

In **Patch From**, the two contributor versions are the same as in a **Merge**:

- The lower right pane contains the version in your workspace.
- The lower left pane contains the stream (or "from") version -- the version, located in some other stream, that you selected when invoking **Patch From**.

But the patch algorithm does not use the actual closest common ancestor of these two contributors as the third version. Instead, it regards the stream version as being the head version of a patch, and uses the corresponding basis version as the closest common ancestor. (See *Structure of a Patch* on page 227.)

Using the basis version instead of the actual closest common ancestor effectively modifies the algorithm for handling change sections, enabling AccuRev to distinguish changes that are "in the patch" from changes made in other versions:

- If a change in the "from" version is in the patch, patching works in exactly the same way as merging. This means that it is possible (and perhaps automatic) that the "from" version's change will be included in the results of the **Patch From**.
- If a change in the "from" version is *not* in the patch, it is ignored -- the workspace version's change is automatically included in the results. (Why automatically? This case fits the definition of a non-conflicting change: "not in the patch" means that the text in the stream version is the same as the text in

the basis version (playing the role of closest common ancestor). Since only one contributor, the workspace version, has a change that differs from the version designated as the closest common ancestor, that contributor's change is accepted automatically.)

Reverse Patch: Removing Content Changes

In a **Revert**, the changes in a specified set of versions are removed from a stream's current versions of those elements:

- When reverting a transaction, AccuRev regards each version in that transaction as being the head version of a patch, and removes all the changes in that patch from the element.
- When reverting a change package, AccuRev removes all the changes in each element's change package entry.

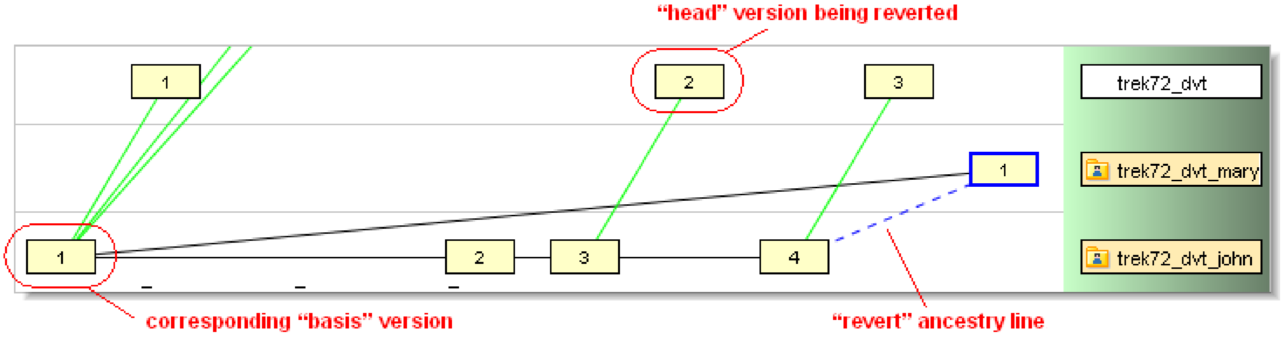
For more information on the structure of a patch and a change package entry, see *Patch and Patch From* on page 227.

For each element it processes, **Revert** uses the AccuRev Merge tool (or user-configured tool) to perform the operation that removes a set of changes from the current version. We use the term **reverse patch** to describe this process, but it's really just another instance of effectively modifying the merge algorithm by switching around the versions.

When the Merge tool is invoked by **Revert**:

- The version you selected when invoking **Revert** is designated to be the closest common ancestor.
- The lower left pane contains the version currently in the stream.
- The lower right pane contains the version that was in the stream just before the version being reverted was promoted there.

In this algorithm's switching around of the versions, the basis version of the change package entry being reverted (or the basis version corresponding to the promoted version being reverted) becomes the direct ancestor of the newly created version. The reverted element's *Version Browser* display shows this relationship:



Resolving Namespace Conflicts

Note: This topic discusses conflicts that can occur when two developers change the pathname of an element. The same kind of conflicts can occur when two developers change the target element an element link or the target pathname of a symbolic link. The same dialogs appear in both kinds of conflict situations.

Kinds of Namespace Changes

AccuRev distinguishes between these two kinds of namespace changes to an element (file, directory, or link):

- **Rename**: Changing the element's simple name (or "leaf name"):

  ```
  cmds.java --> commands.java
  ```

- **Move**: Relocating the element to another directory in the depot::

  ```
  cmd_intf/commands.java --> cmd_intf/utils/commands.java
  ```

Note: What about renaming the directory in which a file resides?

> Renaming of the parent directory is not the same as moving a file to another directory. AccuRev considers this renaming:
>
> ```
> cmd_intf --> cmd_interface
> ```
>
> ... to be change to the parent directory element, but not a change to the file elements within it. (This change to the parent directory has the *side-effect* of changing the pathnames of all elements below it.)

In a parallel development environment, namespace changes can overlap in the same way as content changes. For example, you and a colleague might rename the same file in your workspaces:

```
cmds.java --> commands.java

cmds.java --> all_commands.java
```

If your colleague promotes his change first, then the file will get **(overlap)** status in your workspace. Before you can promote the file, you must use the *Merge* command to resolve the namespace conflict.

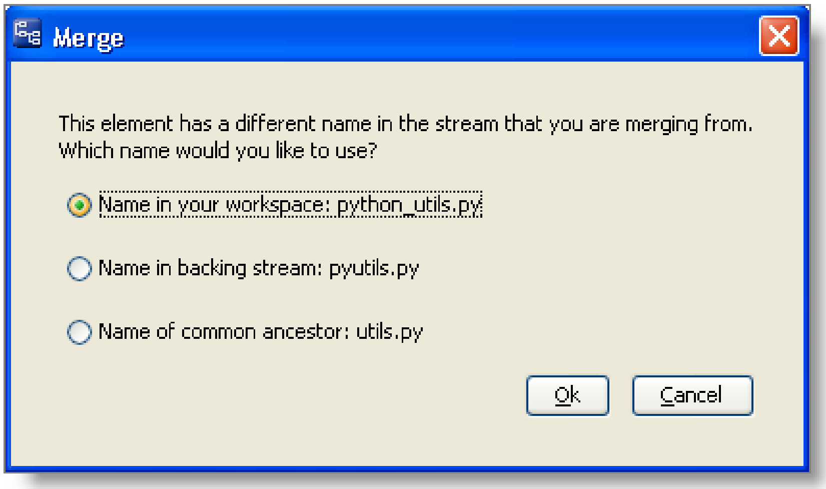
In general, your work on a file element can overlap with a colleague's work in three ways:

- You both have made a namespace change of the **rename** kind.
- You both have made a namespace change of the **move** kind.
- You both have made a content change to the file.

Any combination of these kinds of changes is possible. AccuRev first prompts you to resolve the namespace conflicts, if any, as described below. Then, it passes control to the Merge tool (or another tool that you've configured), to perform the content merge.

Resolving a "Rename" Conflict

A namespace conflict of the **rename** kind occurs when two versions' simple names differ from the simple name of their closest common ancestor (and also differ from each other).



This element has a different name in the stream that you are merging from.
Which name would you like to use?

● Name in your workspace: python_utils.py

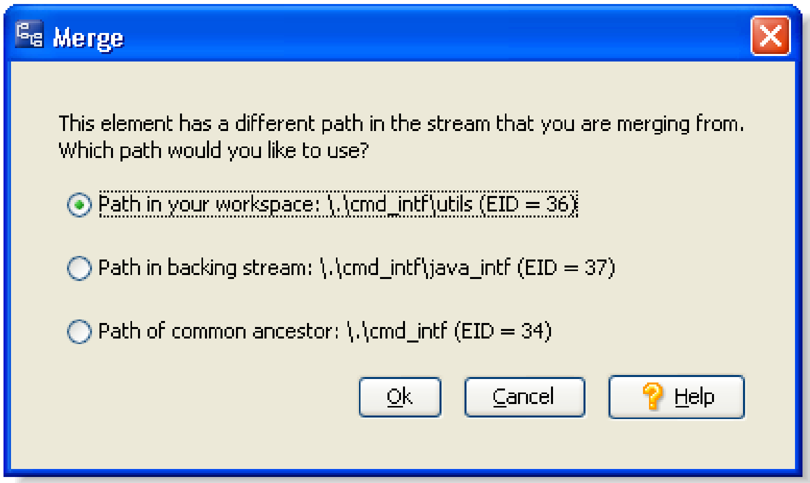
○ Name in backing stream: pyutils.py

○ Name of common ancestor: utils.py

AccuRev allows you to resolve the conflict by taking any of the three simple names.

Resolving a "Move" Conflict

A namespace conflict of the **move** kind occurs when two versions are located in different directories than their closest common ancestor (and also different directories from each other).



This element has a different path in the stream that you are merging from.
Which path would you like to use?

● Path in your workspace: \.\cmd_intf\utils (EID = 36)

○ Path in backing stream: \.\cmd_intf\java_intf (EID = 37)

○ Path of common ancestor: \.\cmd_intf (EID = 34)

AccuRev allows you to resolve the conflict by choosing any of the three parent directories as the file's location.

> **Note:** *What are the "EID" annotations?* AccuRev tracks directory elements by their immutable element-IDs (EIDs). It lists the parent directories' EIDs in this dialog to help you identify them; the directories might have different names/pathnames in different streams.

Merging Versions of a Binary File

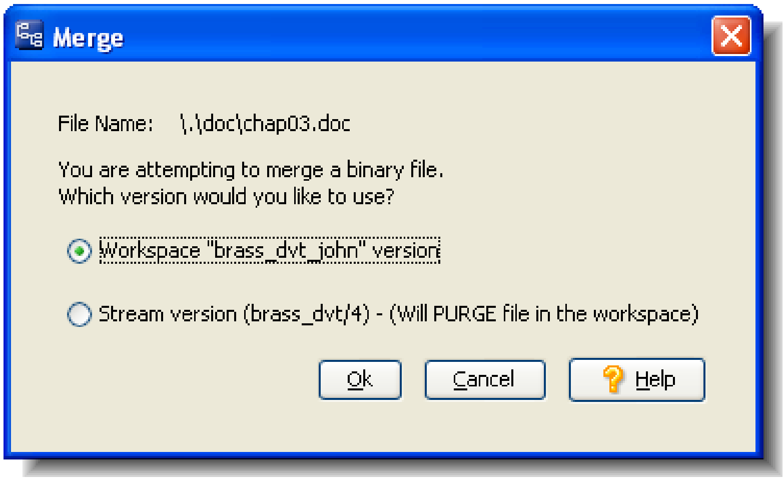
No generally accepted algorithm exists for merging the contents of binary-format files. But it is quite possible for a binary-format element to get into an overlap state in a concurrent development environment.

Example: Two team members might each revise the corporate logo file using an image editor, and then keep it. One of them then promotes the file to the backing stream, creating an overlap situation in the other workspace.

To resolve a binary-file overlap, you invoke **Merge**, just as for a text file. The Binary Merge dialog appears.

Using the Binary Merge Dialog

The Merge tool, seeing that versions to be merged are in binary format, offers the only two possible choices:



File Name: \.\doc\chap03.doc

You are attempting to merge a binary file.
Which version would you like to use?

● Workspace "brass_dvt_john" version

○ Stream version (brass_dvt/4) - (Will PURGE file in the workspace)

Workspace version

Create a new version using the file in your workspace tree, and mark that version as having been merged with the version in the backing stream. You can now **Promote** this new version to the backing stream.

Stream version

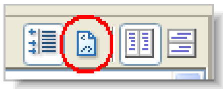
This invokes **Revert to Basis**, which resolves the situation by purging the **(overlap)**-status version from your workspace. The workspace reverts to using the version of the file that was in the backing stream at the previous update.

The file's status becomes **(stale),** because the version that previously caused the overlap is in the backing stream. To bring that version into your workspace, perform an **Update**.

Merging Versions of an HTML File

AccuRev's merge algorithm treats HTML-format files just like all other text files: merging takes place line-by-line; no attempt is made to parse the files' HTML data structures in determining difference sections.

To help you determine whether the merge process is producing a valid (and desirable) result, an HTML viewer is built into the Merge tool.



At any time, you can click the **Show HTML Result** button at the right side of the toolbar. (This button appears only when you're merging a file with a *.htm* or *.html* suffix.)

Clicking the **Show HTML Result** button displays a rendering of the merged HTML-format text in the upper pane. This button is a toggle: click it again when you're finished viewing the rendered HTML code and you want to return to performing the merge.

Merging Versions of a Link

The **Merge** command can merge changes made to element in two different contributor versions.

Basic Merge Algorithm

The basic procedure for handling changes in a merge operation is the same for *all* elements, including link elements:

- At the time of the merge, one contributor is located in a workspace, where the new version will be created); the other contributor is the version in the workspace's backing stream or some other dynamic stream. (You might have specified this second version as part of a particular transaction or change package.)
- If just one contributor makes a change from the contributors' closest common ancestor, AccuRev incorporates the change into the new, merged version automatically.
- If both contributors make a change from the contributors' closest common ancestor, AccuRev regards this as a conflict, which you must resolve during the merge process.

For a more detailed discussion, in the context of changes to the contents of a text file, see *Handling Change Sections* on page 236.

What Constitutes a Change to a Link?

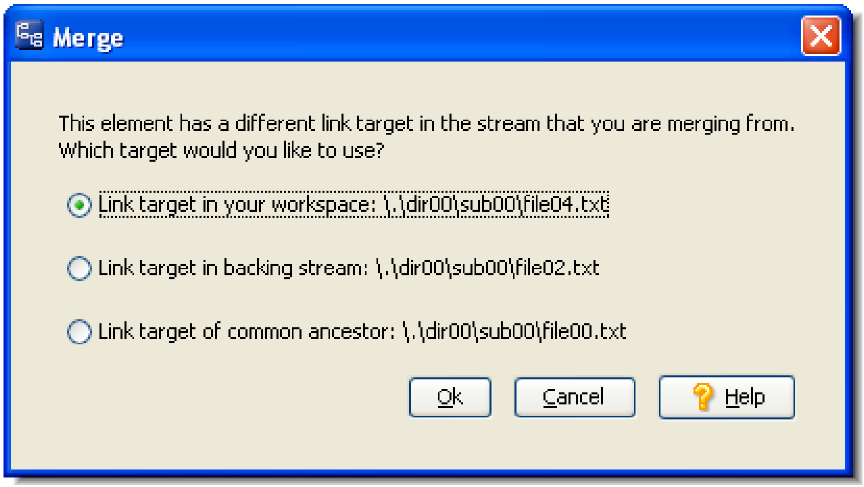
Given that the algorithm for handling changes is basically the same for all elements, the only thing special about merging versions of a link is the nature of changes to a link element.

- AccuRev tracks namespace changes to a link:
  - Initial creation of the link. *Note:* A link object is created as either an element link or a symbolic link. You cannot switch the type of an existing link object.
  - Renaming the link.
  - Moving the link to another location in the depot's directory tree.

  Each of the above changes creates a new version of the link element.
- AccuRev also tracks changes to the contents of a link -- that is, changes to the link's target element (for an element link) or to the link's target pathname (for a symbolic link). Each such change creates a new version of the link element.

Link Merge Procedure

Executing **Merge** on a link element never involves the Merge tool (or third-party tool), which is designed to process the contents of text files.



This element has a different link target in the stream that you are merging from.
Which target would you like to use?

● Link target in your workspace: \.\dir00\sub00\file04.txt

○ Link target in backing stream: \.\dir00\sub00\file02.txt

○ Link target of common ancestor: \.\dir00\sub00\file00.txt

A conflict at the content-level merge is handled by a link-specific dialog:

A conflict at the namespace-level merge involves making choices in the same one or two dialogs used for all kinds of elements.

Merging Versions of a Directory

Merge can merge changes made to a directory (folder) element in two different contributor versions.

Basic Merge Algorithm

The basic procedure for handling changes in a merge operation is the same for *all* elements, including directory elements:

- At the time of the merge, one contributor is located in a workspace, where the new version will be created); the other contributor is the version in the workspace's backing stream or some other dynamic stream. (You might have specified this second version as part of a particular transaction or change package.)
- If just one contributor makes a change from the contributors' closest common ancestor, AccuRev incorporates the change into the new, merged version automatically.
- If both contributors make a change from the contributors' closest common ancestor, AccuRev regards this as a conflict, which you must resolve during the merge process.

For a more detailed discussion, in the context of changes to the contents of a text file, see *Handling Change Sections* on page 236.

What Constitutes a Change to a Directory?

Given that the algorithm for handling changes is basically the same for all elements, the only thing special about merging versions of a directory is the nature of changes to a directory element.

- AccuRev tracks only namespace changes to a directory:
  - Initial creation of the directory.
  - Renaming the directory.
  - Moving the directory to another location in the depot's directory tree.

  Each of the above changes creates a new version of the directory element.
- AccuRev does not track changes to the contents of a directory. For example:
  - Adding a new element to a directory or renaming one of its elements not a change to the directory -- it's a change to that particular element.
  - Similarly, removing (with **Defunct**) an element from a directory is not a change to the directory -- it is a change to that particular element.

  Each of the above changes creates a new version of the element within the directory, but does not affect the directory element itself.

Directory Merge Procedure

Since AccuRev tracks only namespace changes to a directory, not content changes, execution of the Merge command on a directory element never involves the Merge tool (or third-party tool). A conflict at the namespace-level merge involves making choices in the same one or two dialogs used for all kinds of elements.

The Merge Complete Dialog

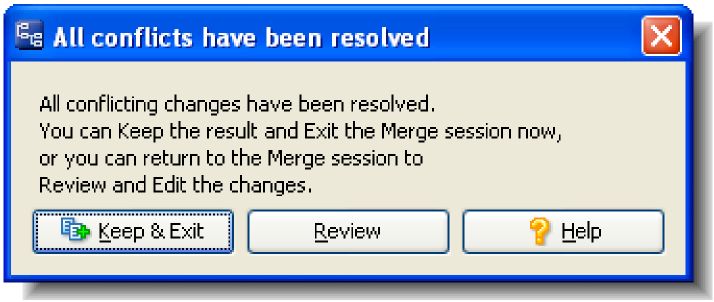
The Merge Complete ("All conflicts have been resolved") dialog appears during a *Merge tool* session, when you accept a change that decrements the Conflicts count to "0 of 0".

Using the Merge Complete Dialog

You have these choices:

Keep & Exit

Ends the Merge session, closing the Merge tab. AccuRev overwrites the workspace file involved in the **Merge** with the merged version


All conflicts have been resolved

All conflicting changes have been resolved.
You can Keep the result and Exit the Merge session now,
or you can return to the Merge session to
Review and Edit the changes.

Note: You can think of the Merge tool as a fancy text-editor. It modifies the contents of a text file in the workspace.

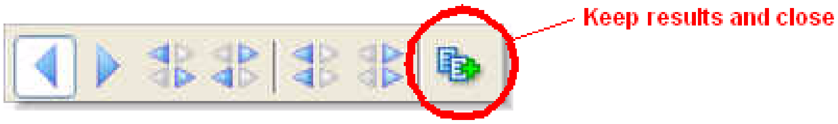
Then, it preserves this file by keeping a new version in the workspace stream. If the file had **(overlap)** status in the workspace, that status is removed.

Note: The workspace file has **(overlap)** status when you're merging your workspace version with the version in the workspace's backing stream. When you're merging a Change Palette entry, using a workspace based on the destination stream, the workspace file does not necessarily have **(overlap)** status.

You can now **Promote** the new version.

Review

Resumes the Merge session, leaving the Merge tab open so that you can review your choices and changes. You can revise your merge choices and make manual edits. When you're done, click the **Keep results and close** button on the Merge toolbar:



The Change Palette

The Change Palette is a tool for propagating changes 'outside the lines' of a depot's stream hierarchy. You use it to promote one or more versions from a dynamic stream to a stream other than its parent stream. You cannot use the Change Palette to promote versions from your workspace to an arbitrary stream. The only destination for versions in a workspace is the workspace's backing stream, via **Promote**. (But you can reparent a workspace to provide a new destination for promoted versions.)

The Change Palette helps you to manage **Merge** and subsequent **Promote** operations. You can use **Patch** instead of **Merge**, in order to send just some -- not all -- of the changes that a version contains.

There are two Change Palette modes: one for working with files and transactions, and another optimized for working with issues.

Although you can manually invoke the Change Palette by using **Send to Change Palette** in the Active Files and Active Issues windows, you typically start cross-promote operations by dragging and dropping files or issues in the stream browser. If the operation encounters conflicts that require merges or other user intervention, then AccuRev automatically displays the Change Palette to help you address those conflicts.

Note: This automatic display of the Change Palette only when you need it is new behavior as of AccuRev Release 6.0, as part of the change package enhancements introduced in that release. If you are upgrading from an earlier version of AccuRev, you will find that basic operations that do not involve conflicts no longer require the Change Palette; they now complete automatically from a simple drag & drop.

Promote vs. Cross-Promote operations

A simple **promote** operation transitions a version from being active in one workspace or stream to being active in the *parent* stream. A **cross-promote** operation, on the other hand, moves content from a dynamic stream into a *non-parent* stream or workspace. Generally speaking, there are two reasons why you would want to perform a cross-promote operation:

- To incorporate work done on an older version into a newer version
- To replicate a file that exists in one dynamic stream into another workspace or stream where it does not yet exist

Note: AccuRev considers the best practice to be cross-promoting from a stream to a workspace and not to another stream.

The effect of cross-promoting a file or a transaction is different from that of cross-promoting an issue. When you cross-promote a file or a transaction, you are transitioning the entire contents of a file or a set of files (in the case of a transaction). When you cross-promote an issue, however, you are applying a change package patch for each of the files involved in the issue and this patch may be only a portion of a file, based on changes that have occurred between the basis and head versions of the file.

When you cross promote a file, transaction, or issue, AccuRev automatically determines whether or not a patch or merge operation is required. If neither of these processes is necessary, the cross promote operation is finished. However, if AccuRev determines that a patch or merge operation is needed, the Change Palette automatically displays enabling you to take the necessary steps to complete the cross-promote.

Opening the Change Palette

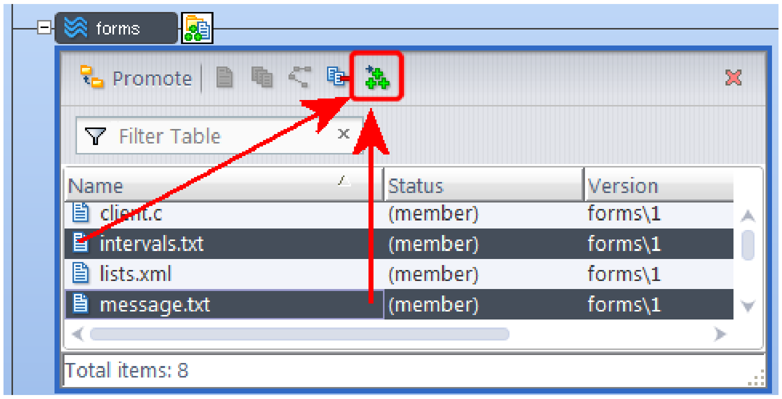
As stated above, you generally do not need to invoke the Change Palette manually: just drag and drop the element versions or issues that you wish to cross-promote, and AccuRev will determine whether or not you need to use the Change Palette to complete the operation. But if you want to send something to the Change Palette manually, here is how you do it. For files, see the next section. For issues, see *Issue Mode* on page 250.

File and Transaction Mode

First you must specify the version(s) that are to be promoted 'outside the lines' of the depot's stream hierarchy. You can do this in the variety of contexts listed below.

From a StreamBrowser: Selected items from a current development activity subwindow

The current development activity subwindow shows what is currently taking place in a stream or workspace -- *by element*, *by transaction*, or *by issue record*. In this subwindow, you can:



Name	Status	Version
client.c	(member)	forms\1
intervals.txt	(member)	forms\1
lists.xml	(member)	forms\1
message.txt	(member)	forms\1

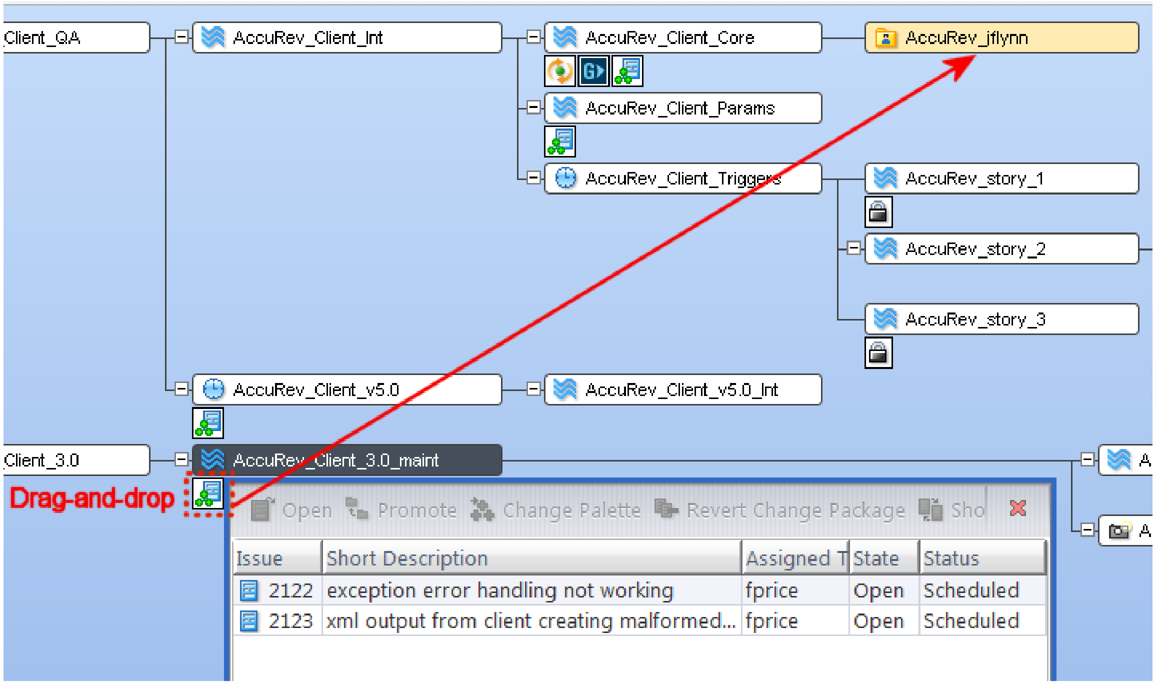
Total items: 8

1. Select any number of items (elements, transactions, or issue records).

2. Invoke **Send to Change Palette**, using the toolbar button or the selection's context menu.

From a StreamBrowser: All the versions with changes that have not been propagated to the destination

Instead of selecting individual versions, you can let AccuRev determine the complete set of versions in a source stream that have not yet been propagated to a destination stream or workspace.



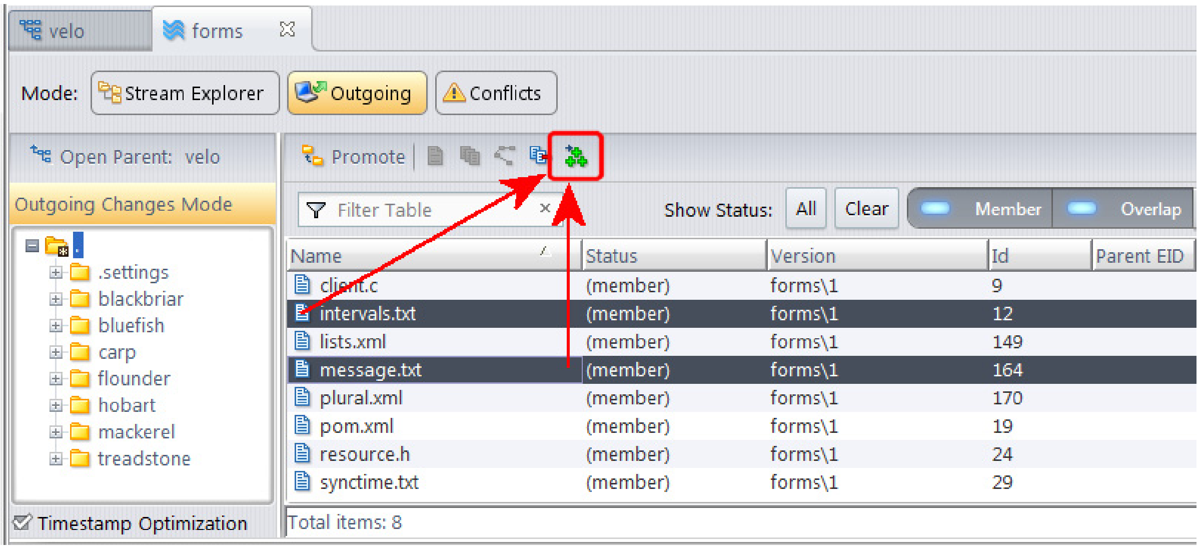
Issue	Short Description	Assigned T	State	Status
2122	exception error handling not working	fprice	Open	Scheduled
2123	xml output from client creating malformed...	fprice	Open	Scheduled

This can include versions that are not currently in the source stream's default group at all, but are inherited from a higher-level stream.

Click the control adjacent to the source stream, and drag it to the destination stream or workspace. If any of the files that are contained in the issues you have cross-promoted require patching or merging, the issue change palette appears.

From the File Browser Outgoing Changes mode, opened on a stream: Selected versions from the default group

This is essentially the same as working in a Default Group subwindow in the StreamBrowser:



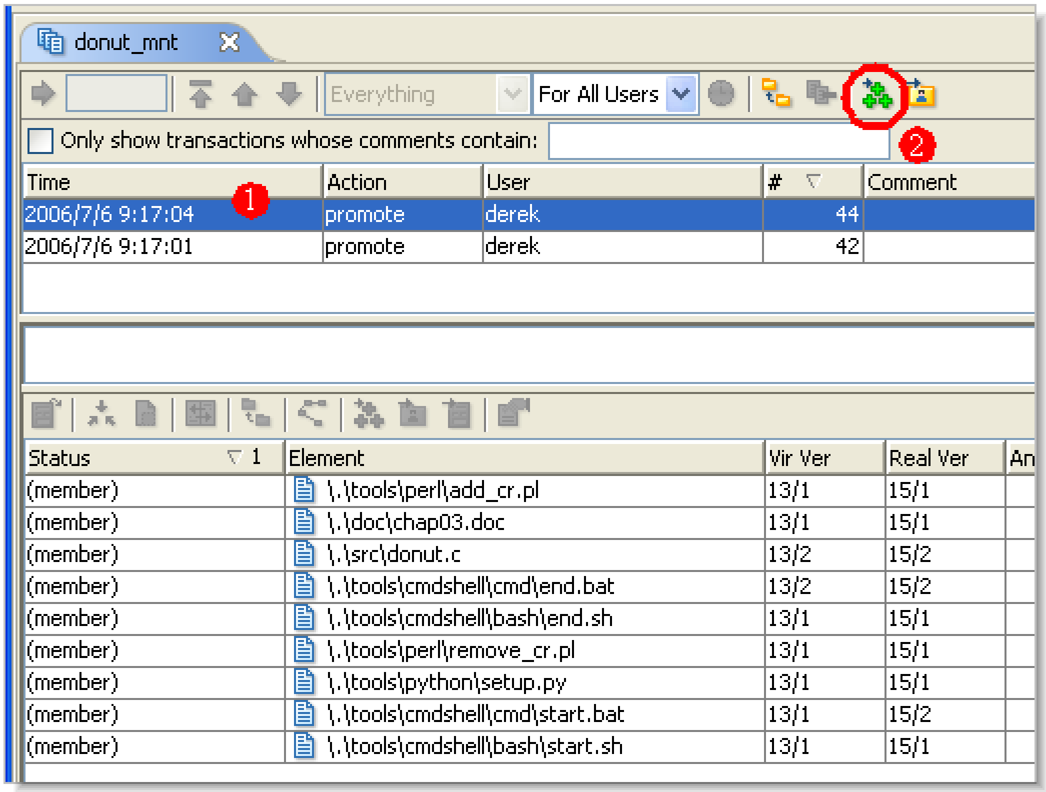
Name	Status	Version	Id	Parent EID
client.c	(member)	forms\1	9	
intervals.txt	(member)	forms\1	12	
lists.xml	(member)	forms\1	149	
message.txt	(member)	forms\1	164	
plural.xml	(member)	forms\1	170	
pom.xml	(member)	forms\1	19	
resource.h	(member)	forms\1	24	
synctime.txt	(member)	forms\1	29	

Total items: 8

1. Select the Outgoing Changes mode and Member search, if necessary.
2. Select any number of the stream's active versions.
3. Invoke **Send to Change Palette**, using the ✚ toolbar button or the selection's context menu.

From a History Browser or Active Transactions tab, opened on a stream: All the versions in specified transaction(s)

A History Browser tab shows all the transactions that created (via **Promote**) versions in a stream. An Active Transactions tab shows the subset of these transactions with at least one version still active in the stream. In either one, you can load a Change Palette with the versions created one or more selected transactions.



□ Only show transactions whose comments contain:

Time	Action	User	#	Comment
2006/7/6 9:17:04	promote	derek	44	
2006/7/6 9:17:01	promote	derek	42	

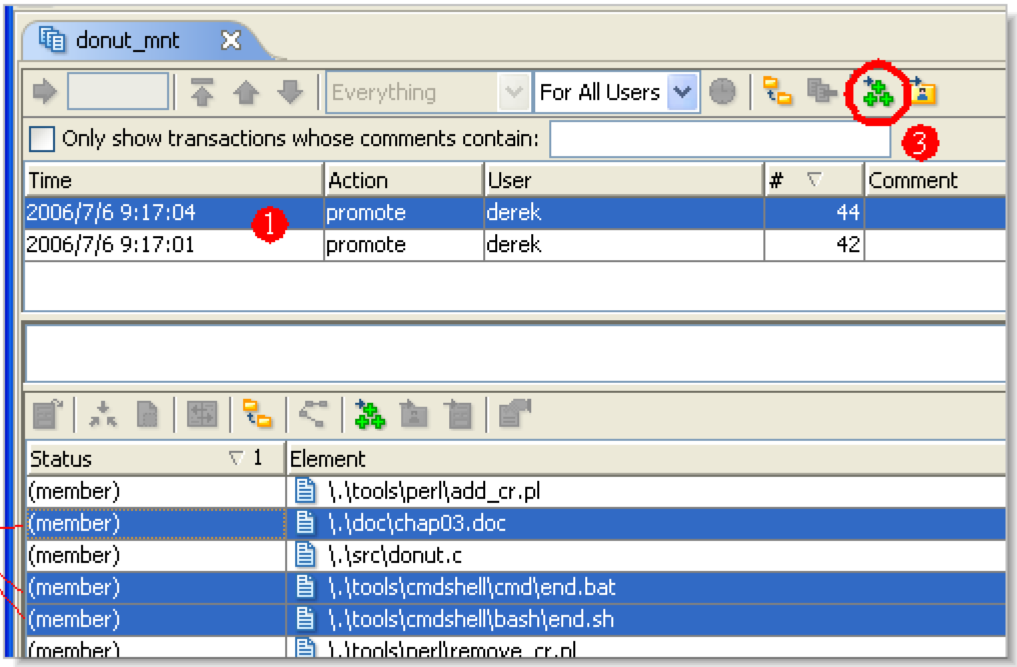
Status 1	Element	Vir Ver	Real Ver	An
(member)	\.\tools\perl\add_cr.pl	13/1	15/1	
(member)	\.\doc\chap03.doc	13/1	15/1	
(member)	\.\src\donut.c	13/2	15/2	
(member)	\.\tools\cmdshell\cmd\end.bat	13/2	15/2	
(member)	\.\tools\cmdshell\bash\end.sh	13/1	15/1	
(member)	\.\tools\perl\remove_cr.pl	13/1	15/1	
(member)	\.\tools\python\setup.py	13/1	15/1	
(member)	\.\tools\cmdshell\cmd\start.bat	13/1	15/2	
(member)	\.\tools\cmdshell\bash\start.sh	13/1	15/1	

1. Select one or more transactions in the Summary (upper) pane.

2. Invoke **Send to Change Palette**, using the toolbar button or the selection's context menu.

From a History Browser or Active Transactions tab, opened on a stream: Specified version(s) from one transaction

This is similar to the previous case: instead of loading a Change Palette with *all* the versions created in a transaction, you load selected versions from a transaction.



Time	Action	User	#	Comment
2006/7/6 9:17:04	promote	derek	44	
2006/7/6 9:17:01	promote	derek	42	

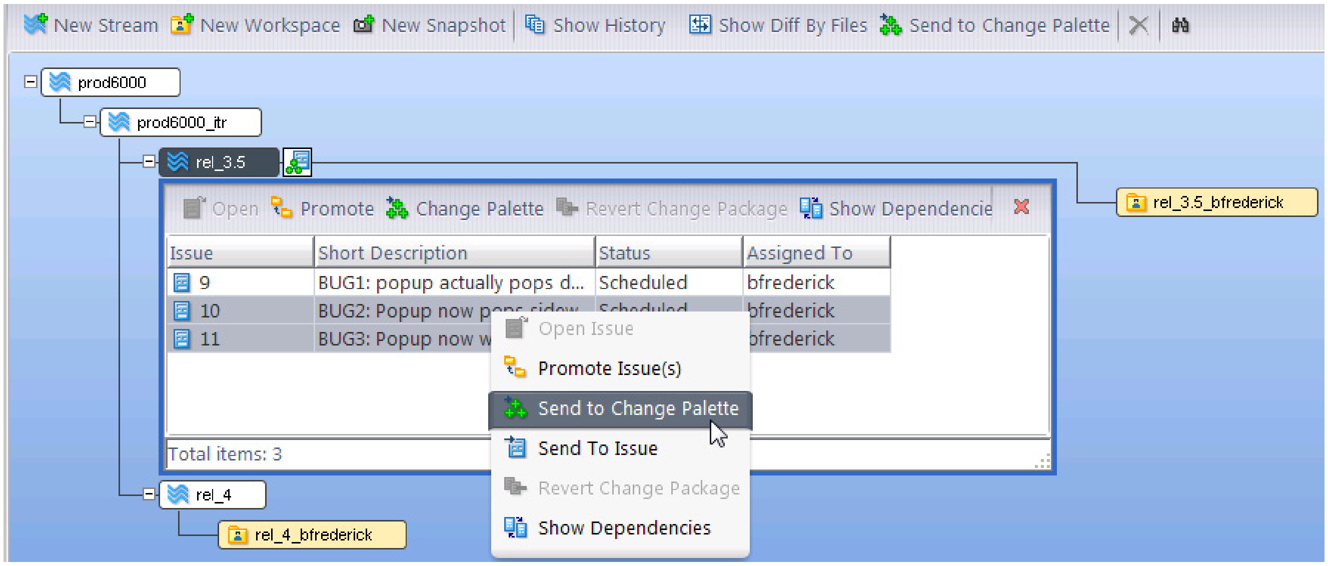
Status	Element
(member)	\.\tools\perl\add_cr.pl
(member)	\.\doc\chap03.doc
(member)	\.\src\donut.c
(member)	\.\tools\cmdshell\cmd\end.bat
(member)	\.\tools\cmdshell\bash\end.sh
(member)	\.\tools\perl\remove_cr.pl

1. Select one transaction in the Summary (upper) pane.
2. Select any number of that transaction's versions in the Versions (lower) pane.
3. Invoke ***Send to Change Palette***, using the toolbar button or the selection's context menu.

Issue Mode

From an Active Issues tab: All the versions in the change package(s) of the specified issue record(s)

An Active Issues tab shows the issue records whose change packages are "in" a particular stream.



Issue	Short Description	Status	Assigned To
9	BUG1: popup actually pops d...	Scheduled	bfrederick
10	BUG2: Popup now pops sidew...	Scheduled	bfrederick
11	BUG3: Popup now w		bfrederick

Total items: 3

Each change package contains a "patch" of data from one or more text-file elements. You can load the Change Palette with the versions in one or more issue records' change packages:

1. Select one or more issue records in the Issues (upper) pane.
2. Invoke **Send to Change Palette**, using the toolbar button or the selection's context menu.

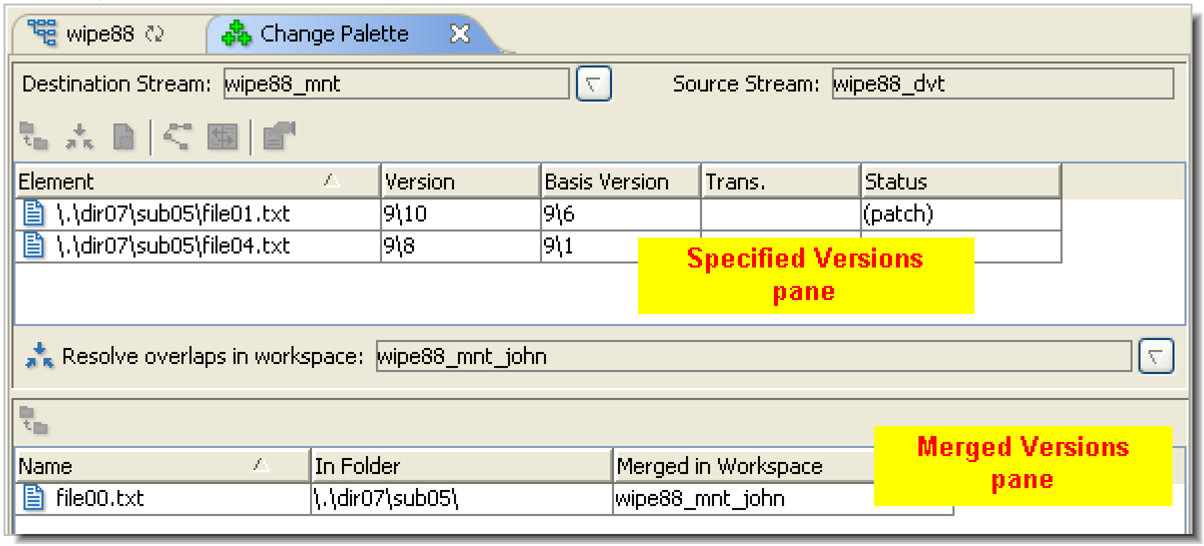
Anew Issues Change Palette appears, loaded with the versions you've specified. (See *Issue Mode* on page 254.)

Change Palette Layout

The layout of the Change Palette differs depending on whether you are working with files and transactions, or issues. The files and transactions layout is discussed in the following section. The issues layout is described in *Issue Mode* on page 254.

File and Transaction Mode

The Change Palette includes these panes:



Destination Stream: wipe88_mnt   Source Stream: wipe88_dvt

Element	Version	Basis Version	Trans.	Status
\.\dir07\sub05\file01.txt	9\10	9\6		(patch)
\.\dir07\sub05\file04.txt	9\8	9\1		

Specified Versions pane

Resolve overlaps in workspace: wipe88_mnt_john

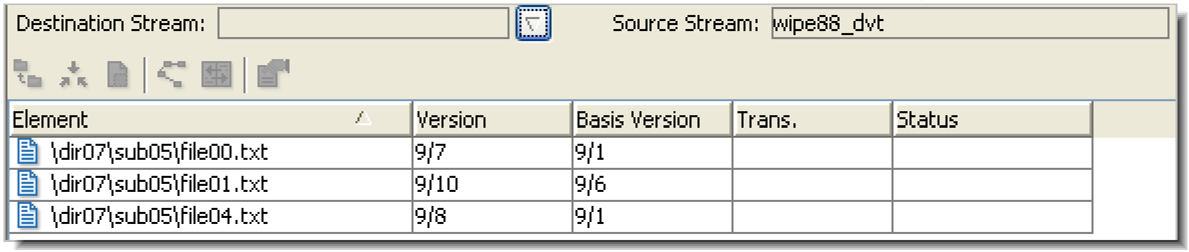
Name	In Folder	Merged in Workspace
file00.txt	\.\dir07\sub05\	wipe88_mnt_john

Merged Versions pane

- The *Specified Versions* pane initially lists all the versions you specified with **Send to Change Palette** (or the drag-and-drop operation). The versions' entries are cleared from this pane as you work with them using **Promote**, **Merge**, **Patch**, etc.
- The *Merged Versions* pane appears only after you've merged or patched versions in the Specified Versions pane. It lists the versions created by **Merge** or **Patch**, and provides a convenient way to promote these versions to the destination stream.

Layout of the Specified Versions Pane

This pane contains a table that initially lists all the versions loaded into the Change Palette.



Destination Stream:

Source Stream: wipe88_dvt

Element	Version	Basis Version	Trans.	Status
\dir07\sub05\file00.txt	9/7	9/1		
\dir07\sub05\file01.txt	9/10	9/6		
\dir07\sub05\file04.txt	9/8	9/1		

As you work with these versions, they "migrate" out of this pane:

- Entries for versions that you promote to a destination stream -- or send to a destination workspace -- are automatically cleared from the Change Palette.
- Entries for versions that you *patch* into a workspace (based on the source or destination stream) are also cleared automatically.
- Entries for versions that you *merge* into a workspace (based on the source or destination stream) are automatically cleared from the Specified Versions pane. Corresponding entries, representing the versions created by **Merge**, are added to the Merged Versions pane.

Columns in the Specified Versions Table

The Specified Versions table includes these columns:

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

Version

The version of the element that was sent to the Change Palette.

Basis Version

The version/basis-version pair defines the "patch data" that will be incorporated into the destination version if you invoke **Patch**. (See *Patch and Patch From* on page 227.)

Transaction

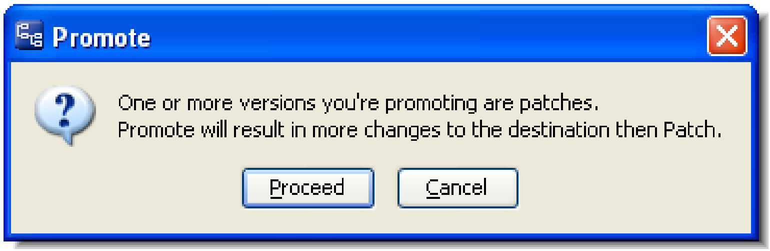
(if you invoked **Send to Change Palette** from a History Browser or Active Transactions tab) The number of the transaction from which you sent this version.

Status

The relationship between the version in the source stream and the version in the destination stream:

- **(overlap)** indicates that you must perform a **Merge** prior to promoting this element to the destination stream. You can also perform a **Patch** to propagate *some* of this version's changes to the destination stream, but this does not resolve the overlap status (and so does not enable **Promote**).
- **(patch)** indicates that the change that you sent to the Change Palette for this element (as specified by its entries in the Version and Basis Version columns) is a patch. That is, the change does *not* include the entire set of changes between the source-stream version and the closest common ancestor of the source-stream and destination-stream versions. You can perform a **Patch** operation to propagate the changes in the patch to the destination stream.

If the element does not also have **(overlap)** status, the **Promote** command is enabled for this version, too.



One or more versions you're promoting are patches.
Promote will result in more changes to the destination then Patch.

But promoting would propagate more changes to the destination stream than were loaded into the Change Palette. AccuRev issues a warning if you invoke **Promote** on this version.

- A blank indicates that there is no overlap and that the change loaded into the Change Palette is *not* a patch. This means you can safely **Promote** the version to the destination stream.

Other Indicators and Controls

The Specified Versions pane also includes:

Source Stream

The name of the stream from which you sent the versions, using **Send to Change Palette** or the drag-and-drop operation.

Destination Stream

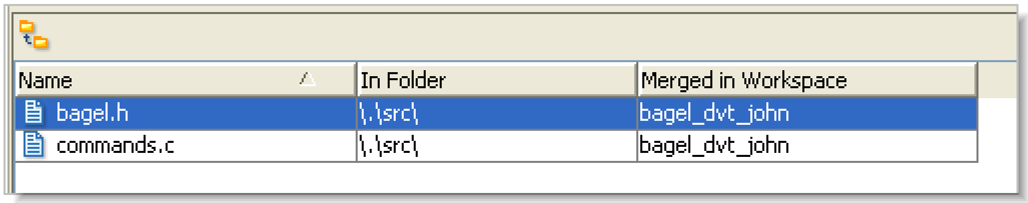
Initially blank (unless you used the drag-and-drop operation to load the Change Palette). Subsequently, it displays the name of the stream or workspace you've selected as the versions' destination.

Resolve overlaps in workspace

This field does not appear until the first time you invoke the **Merge** command. At that point, AccuRev prompts you to specify a workspace for the merge operation to take place in. It then displays the workspace's name in this field (which is also a listbox control for changing the name).

Layout of the Merged Versions Pane

This pane contains a table that lists the versions you've created by invoking the **Merge** command on versions in the Specified Versions pane. As you proceed to **Promote** these newly created versions, their entries are cleared from the Change Palette.



Name	In Folder	Merged in Workspace
bagel.h	\.\src\	bagel_dvt_john
commands.c	\.\src\	bagel_dvt_john

The Merged Versions table includes these columns:

Element (or separate "Name" and "In Folder" columns)

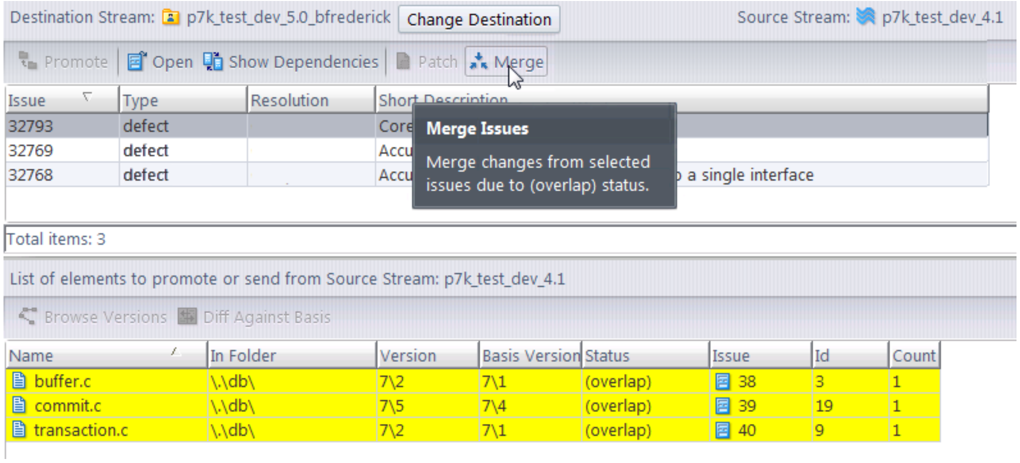
The element's pathname within the depot.

Merged in Workspace

> The name of the workspace in which the merged version was created. This is not necessarily the same for all the listed elements, since you can switch workspaces using the **Resolve overlaps in workspace** control.

Issue Mode

The layout of the Issue Change Palette differs from the one you use for files and transactions.



Merge Issues

Merge changes from selected issues due to (overlap) status.

Issue	Type	Resolution	Short Description
32793	defect		Core
32769	defect		Accu
32768	defect		Accu … a single interface

Total items: 3

List of elements to promote or send from Source Stream: p7k_test_dev_4.1

Name	In Folder	Version	Basis Version	Status	Issue	Id	Count
buffer.c	\.\db\	7\2	7\1	(overlap)	38	3	1
commit.c	\.\db\	7\5	7\4	(overlap)	39	19	1
transaction.c	\.\db\	7\2	7\1	(overlap)	40	9	1

The upper-pane displays the issues that are to be cross-promoted, and the columns are specific to your configuration of AccuRev. The lower pane displays the files that comprise the change package(s) of any issues selected in the upper-pane.

Note that the **Promote** button is disabled if AccuRev sees that the selected issues require intervention.

If you try to cross-promote to a dynamic stream, and need to perform a patch or merge, you must specify a workspace where the operation will take place. When you select **Merge**, the contents of the merge list that appear below the Issue Change Palette are based on the issues that are currently selected in the Change Palette. After performing a merge or a patch operation, AccuRev prompts you to promote the issue.

Working in the Change Palette

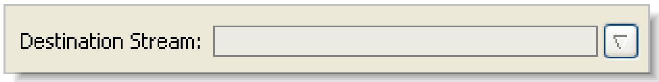
The following sections describe the tasks you perform in the Change Palette. The order of the sections follows the typical order of tasks. But the Change Palette is a long-lived "work environment", not a simple dialog box or sequential wizard. So you might perform just some of the tasks or perform them in a different order.

Also, using the Change Palette with issues has some special considerations that are covered in *Working with the Issue Change Palette* on page 259.

Specifying the Destination Stream

If you've loaded versions into the Change Palette using **Send to Change Palette**, you've already indicated the source stream, but need to specify the destination stream or workspace. (The drag-and-drop operation indicates both the source and destination.)

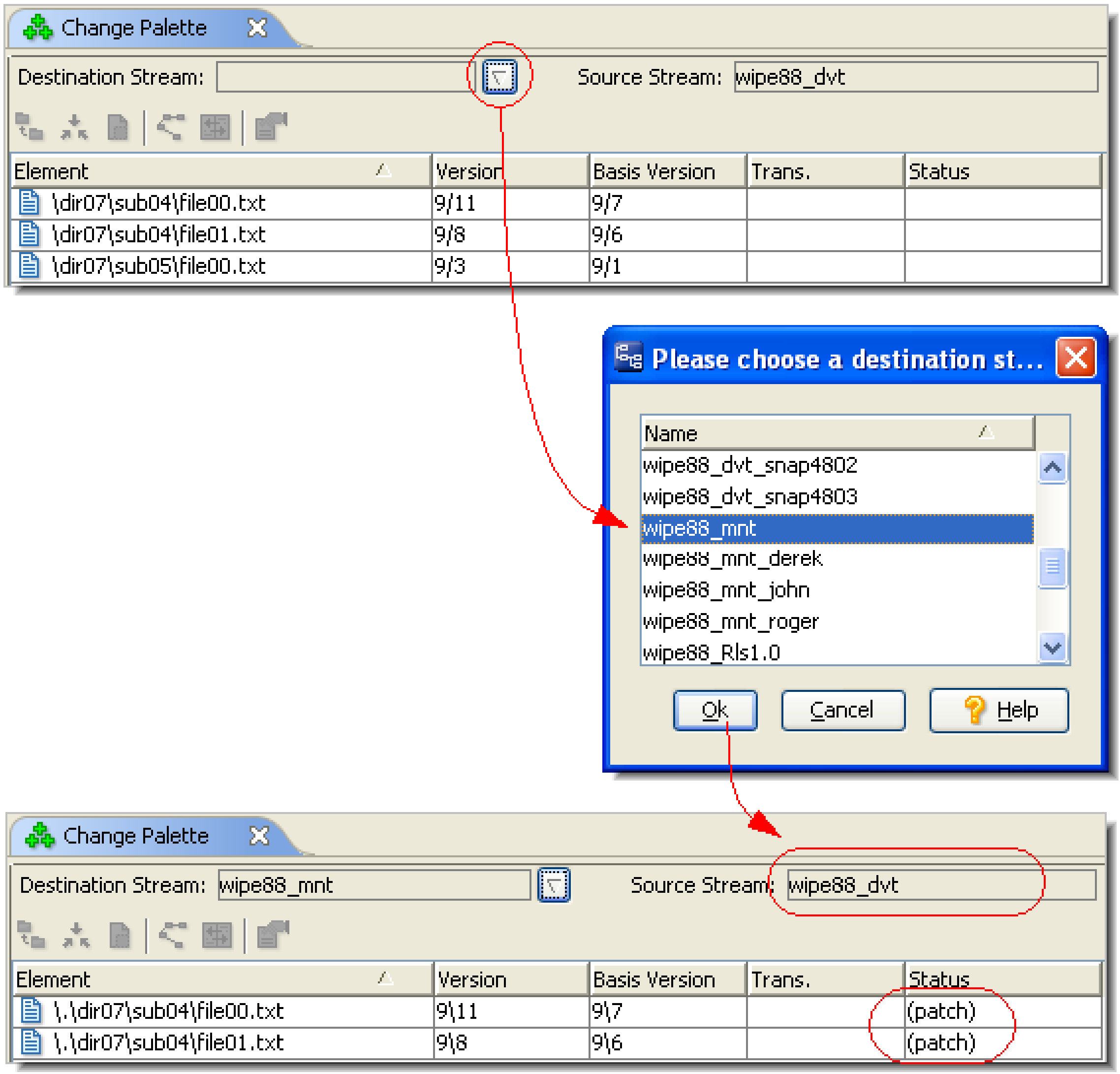
Click the arrow control next to the Destination Stream field.



Destination Stream:

A dialog appears, listing all of the depot's streams and workspaces. Select one to be the destination for subsequent commands (**Send to Workspace**, **Promote**, **Merge**, and/or **Patch**) on the versions in the Specified Versions pane.

Example



Destination Stream:

Source Stream: wipe88_dvt

Element	Version	Basis Version	Trans.	Status
\dir07\sub04\file00.txt	9/11	9/7		
\dir07\sub04\file01.txt	9/8	9/6		
\dir07\sub05\file00.txt	9/3	9/1		

Name
wipe88_dvt_snap4802
wipe88_dvt_snap4803
wipe88_mnt
wipe88_mnt_derek
wipe88_mnt_john
wipe88_mnt_roger
wipe88_Rls1.0

Destination Stream: wipe88_mnt

Source Stream: wipe88_dvt

Element	Version	Basis Version	Trans.	Status
\.\dir07\sub04\file00.txt	9\11	9\7		(patch)
\.\dir07\sub04\file01.txt	9\8	9\6		(patch)

As the example shows, AccuRev determines a status for each element, describing the relationship between the source-stream version and the destination-stream version. Entries with **(overlap)** status are highlighted in yellow.

> **Note:** If the destination is a workspace, AccuRev evaluates the version currently in the workspace stream, which is not necessarily the same as the file currently in the workspace tree.

You can change the destination stream at any time, using the arrow control next to the Destination Stream field.

Sending Versions to the Destination Stream

In the Specified Versions pane:

- If the destination is a stream, you can immediately **Promote** the versions that do not have **(overlap)** status.
- If the destination is a workspace, you can immediately **Send to Workspace** any or all versions, whether or not they have **(overlap)** status.

Select one or more eligible versions, then click the button in the pane's toolbar or choose **Promote** (or **Send to Workspace**) from the selection's context menu. When the command completes, the entry(s) for the version(s) are cleared from the Change Palette.

In the Merged Versions pane, you can **Promote** versions at any time. The versions are active in the workspace listed in the Merged in Workspace column; **Promote** sends them to this workspace's backing stream.

Notes:

- *If you've used multiple workspaces for merging...*

  In each *Promote* invocation, the set of versions must all be in the same workspace.

- *If you've merged using a workspace based on the source stream...*

  Promoting sends the merged version to the workspace's backing stream, which is not the destination stream you originally specified. You can now promote this version from the workspace's backing stream to the original destination stream (no additional merge is required). But you'll need to use another instance of the Change Palette to invoke **Promote**.

Merging the Source and Destination Versions

If an entry in the Specified Versions pane has **(overlap)** status, you cannot immediately promote the version to the destination stream. First, you must create a new version by merging the source and destination versions.

The merge operation must take place in a workspace that belongs to you and is based on either the source or destination stream. If you have not already selected a merge workspace, you must do so before performing any merges. (See *Merging the Source and Destination Versions* on page 256.)

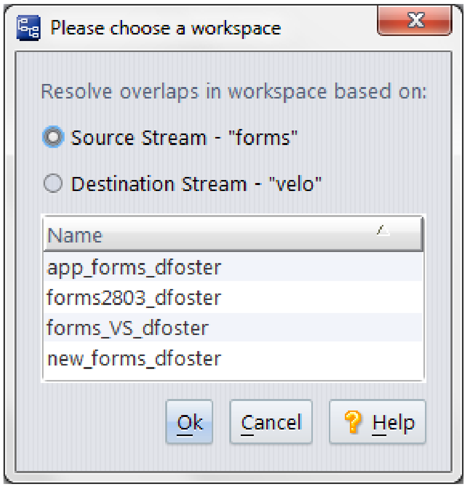
Select one or more versions with **(overlap)** status, and click the button in the pane's toolbar or choose **Merge** from the selection's context menu.

The merge operation(s) proceed for the selected element(s), as described in *Merge* on page 214.

> **Note:** If the merge workspace was not up-to-date, it's possible that the element will continue to have (overlap) status after the merge. In this case, perform a standard **Merge** in the workspace before promoting the element.

Selecting a Workspace for Performing Merges

A merged version cannot be created directly in either the source or destination stream, because these are dynamic streams — all new versions of AccuRev elements must be originally created in workspaces.



Please choose a workspace

Resolve overlaps in workspace based on:

● Source Stream - "forms"

○ Destination Stream - "velo"

Name
app_forms_dfoster
forms2803_dfoster
forms_VS_dfoster
new_forms_dfoster

The first time you invoke **Merge** in the Specified Versions pane, AccuRev prompts you to establish a merge workspace, suggesting one that is based on the destination stream. You can also choose a workspace based on the source stream.

Notes:

- *Which is preferable -- source or destination?*

  A workspace based on the destination stream is preferable. After you perform the merge, you can then accomplish the original task -- propagating a version's changes to the destination stream -- with a single **Promote**. If you use a workspace based on the source stream, you'll need to **Promote** twice: (1) from the merge workspace to the source stream; (2) from the source stream to the destination stream. The second one requires you to use another instance of the Change Palette.

- *What if no candidate workspace exists?*

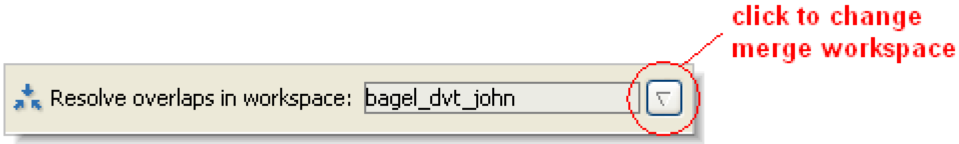
  Click **Cancel** in the choose-workspace dialog, and do either of the following:

  - Create a workspace based on one of the streams.
  - Reparent one of your existing workspaces to the source or destination stream. Be sure to **Update** the re-parented workspace before using it to perform merges in the Change Palette.

- *Dialog box used for patching changes and resolving twin status, too.*

  The Please Choose a Workspace dialog box is also used when patching changes and resolving an element’s twin status. See *Patching Changes from the Source Version into the Destination Version* on page 258 and *Resolving Twins from the Change Palette* on page 258 for more information.

After you select a merge workspace, its name is displayed at the bottom of the Specified Versions pane. You can change this setting at any time; this enables you to use different workspaces to merge different versions.



Resolve overlaps in workspace: bagel_dvt_john

Patching Changes from the Source Version into the Destination Version

If an entry in the Specified Versions pane has **(patch)** status, you can invoke **Patch** to incorporate the patch defined by the Version and Basis Version settings into the destination-stream version. Select one or more of these versions, and click the 📄 button in the pane's toolbar or choose **Patch** from the selection's context menu. AccuRev prompts you to select a workspace that belongs to you and is based on the destination stream. The patch operation is performed in this workspace, as described in *Patch and Patch From* on page 227.

When the patch operation is complete, the entry disappears from the Specified Versions pane.

Resolving Twins from the Change Palette

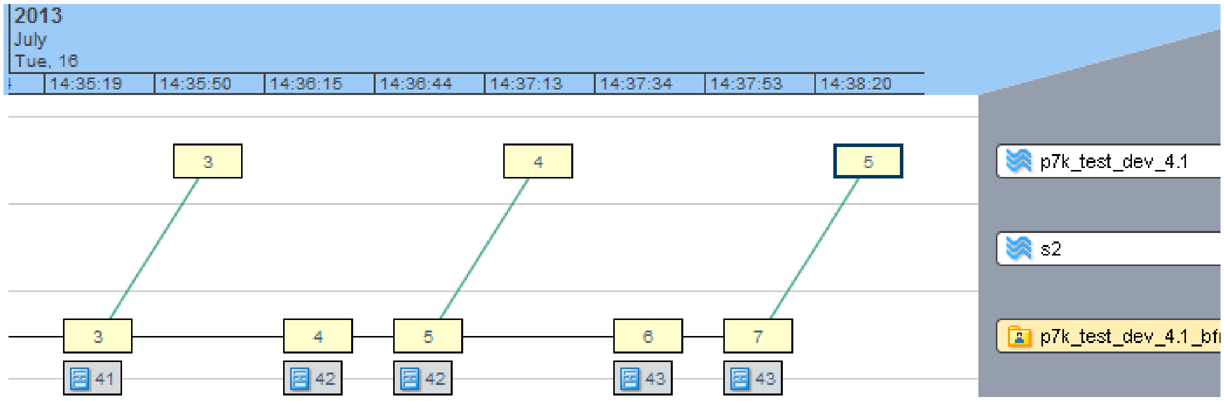
Using the Change Palette to promote elements across streams can reveal the presence of elements with a **(twin)** status, which are highlighted in yellow in the Specified Versions pane. Twin elements (or so called *twins*) can be created when a file with the same name is created in workspaces off both the source and destination streams and then promoted to their respective streams. Elements that have a **(twin)** status cannot be promoted across streams until their **(twin)** status is resolved.

AccuRev provides a twin status resolution wizard and one-step defunct operations designed to help resolve twins quickly and easily. See *Resolving (twin) Status* on page 119 for more information. Note that there are considerations when using the twin resolution wizard to resolve twins from the Change Palette.

Multiple Entries for the Same Element

In some cases, the Specified Versions pane of the Change Palette can get populated with multiple entries for the same element. Here is a simple scenario:

1. Successive changes to an element, `buffer.c`, get assigned to different AccuWork issue records. For example, this illustration shows a one-version change assigned to issue #41, then a two-version change assigned to issue #42, then a two-version change assigned to issue #43.



2. At this point, all three issues (#41, #42, and #43) are "in" the stream to which the versions were promoted.

3. Open an Active Issues window for the stream to which the versions were promoted, then select issues #41 and #43 and invoke **Send to Change Palette**. Since there is a "change package gap", consisting of versions 4 and 5, between the two change package entries for `buffer.c`, AccuRev cannot combine them into a single patch entry. Accordingly, the two entries are entered into the Change Palette

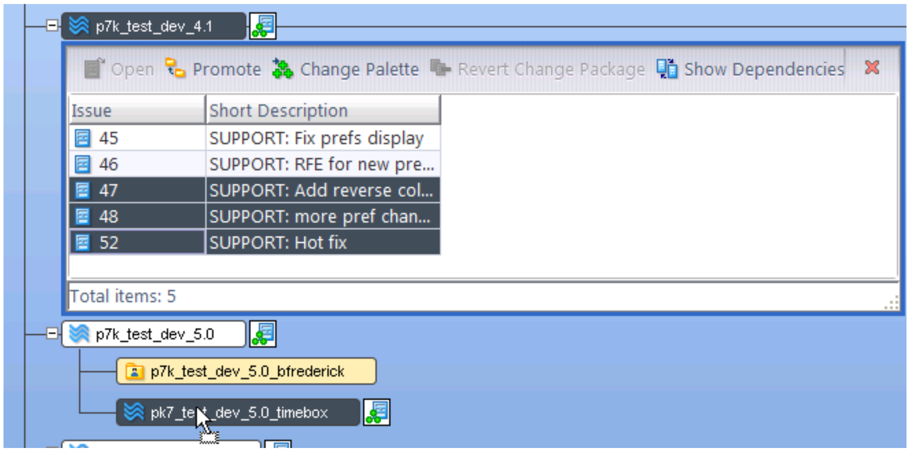
separately and, in this case, a **Patch** operation is required for one entry before the changes can be propagated to the destination stream.

Working with the Issue Change Palette

If you are working with issues rather than files, the Change Palette GUI and the procedures you use with it are somewhat different than what is described for files in the previous sections.

First of all, remember that you might not need the Change Palette at all if you are cross-promoting a single issue, or multiple issues that do not have any elements in common. Just cross-promote in the StreamBrowser by dragging and dropping the issue(s) from the Active Issue mode of the source stream, to the destination stream. The Change Palette appears if AccuRev analyzes your drag & drop and discovers that some or all of your issues require patching or merging. (See *Invoking the Patch by Drag & Drop* on page 229 for a description of the situations where the Change Palette will be automatically displayed.)

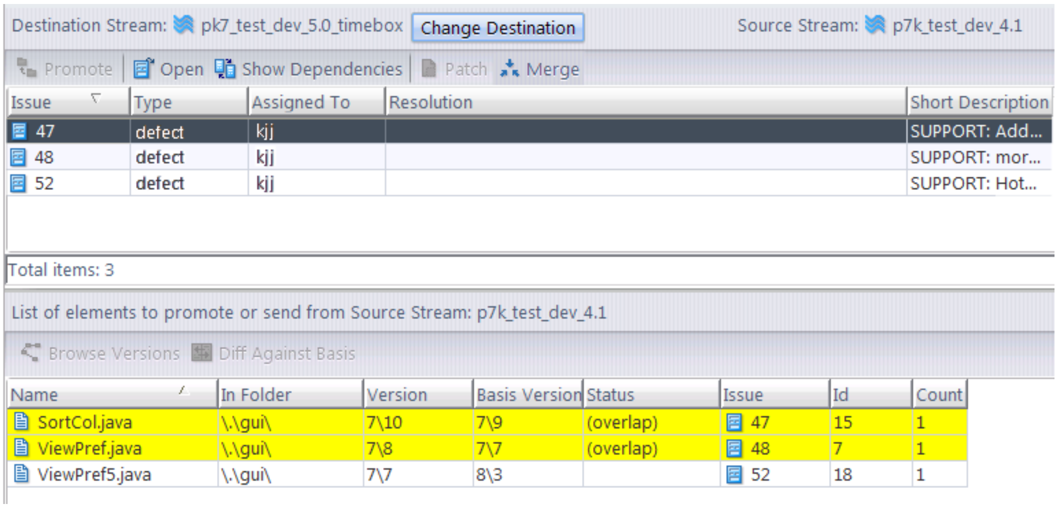
In the following example, three issues are drag & dropped from stream p7k_test_dev_4.1 to stream pk7_test_dev5.0_timebox:



Issue	Short Description
45	SUPPORT: Fix prefs display
46	SUPPORT: RFE for new pre...
47	SUPPORT: Add reverse col...
48	SUPPORT: more pref chan...
52	SUPPORT: Hot fix

Total items: 5

However, two of the issues (47 and 48) will require a merge to complete the operation, so AccuRev displays the Change Palette for you:



Issue	Type	Assigned To	Resolution	Short Description
47	defect	kjj		SUPPORT: Add...
48	defect	kjj		SUPPORT: mor...
52	defect	kjj		SUPPORT: Hot...

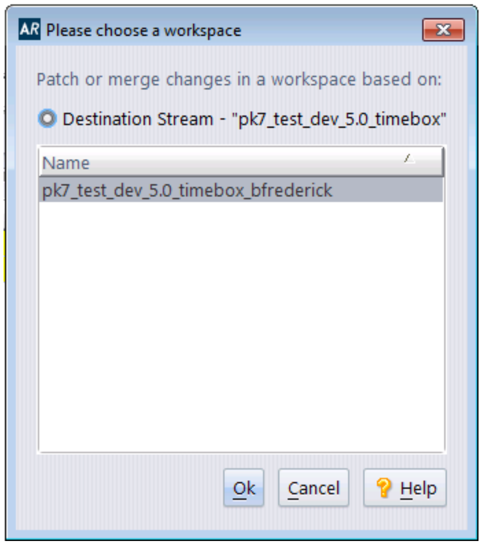
Total items: 3

List of elements to promote or send from Source Stream: p7k_test_dev_4.1

Name	In Folder	Version	Basis Version	Status	Issue	Id	Count
SortCol.java	\.\gui\	7\10	7\9	(overlap)	47	15	1
ViewPref.java	\.\gui\	7\8	7\7	(overlap)	48	7	1
ViewPref5.java	\.\gui\	7\7	8\3		52	18	1

You will need to address these overlaps that need to be merged before you will be able to promote the issue that is ready to go. The files requiring intervention are highlighted in yellow. Once you have addressed the problems, the **Promote** button is enabled. Click the **Promote** button to process the issue which requires no intervention (in this case, 52).

This leaves the issues that require merging. Select them and click the **Merge** button. If you are merging to a dynamic stream, AccuRev prompts you to specify a workspace in which to complete the patch operation:


Please choose a workspace

Patch or merge changes in a workspace based on:

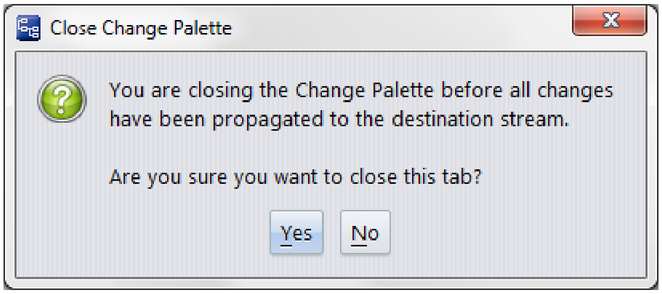
○ Destination Stream - "pk7_test_dev_5.0_timebox"

Name
pk7_test_dev_5.0_timebox_bfrederick

When you click **Ok**, will be presented with one or more Merge screens in which to complete your changes. When done merging your changes, AccuRev prompts you to promote your issues. Click the **Promote** button, and you are done.

Closing the Change Palette

If you attempt to close the Change Palette when either pane is non-empty, AccuRev prompts you for confirmation.


Close Change Palette

You are closing the Change Palette before all changes have been propagated to the destination stream.

Are you sure you want to close this tab?

Note: Remove From Change Package is available from the right mouse button context menu; it is not currently available from the tool bar.

Commands Available in the Change Palette

The commands available to you in the Change Palette vary whether you are in Issue and Transaction Mode or Issue Mode.

File and Transaction Mode

Remove from Change Palette

Remove the selected entry(s). You can clear entries from either pane. This capability helps you to keep track of your work (and the work that you've decided not to do).

Promote

(available only if the destination is a dynamic stream)

Specified Versions pane: promote the selected version(s) to the destination stream.

Merged Versions pane: promote the selected version(s) from the workspace listed in the Merged in Workspace column to its backing stream.

Send to Workspace

(available only if the destination is a workspace)

Performs a *Send to Workspace* command on the selected version(s) in the destination workspace. (see *Send to Workspace* on page 127.)

Merge

Merge the selected version(s) with the corresponding version(s) in the destination stream, creating the merged version(s) in the workspace listed in the *Resolve overlaps in workspace* field. When the command finishes, the entry(s) are cleared from the Specified Versions pane, and entry(s) for the newly created version(s) are added to the Merged Versions pane.

Patch

Performs a **Patch** command from each selected version to the corresponding version in the destination workspace, or in a workspace that you select below the destination stream. When the command finishes, the entry(s) are cleared from the Specified Versions pane.

Resolve Twin Status

Provides a wizard and other tools to help you resolve an element's **(twin)** status. See *Resolving (twin) Status* on page 119 for more information.

Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

Diff Against Basis

Compares the selected version with the version in the destination stream or workspace.

Note: When the destination is a workspace, the comparison is performed with the version in the workspace stream, not the file in the workspace tree. These are different if the element has (modified) status in the workspace.

Properties

Displays information about the selected element. The data items displayed vary with the type of element.

Issue Mode

Promote

Cross-promote any issues that are ready to be promoted and which do not require a merge or patch operation.

Open

Open a separate Edit Form to view/revise the currently selected issue record.

Show Dependencies

Opens the *Issue Dependencies* tab which you can use to propagate issue change packages to other streams and to view the set of issue records on which the selected records depend.

Patch

Performs a **Patch** from the Change Package of each selected issue to the element versions in the destination workspace, or in a workspace that you select below the destination stream.

Merge

Performs a **Merge** from the Change Package of each selected issue with the corresponding version(s) in the destination stream.

Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

Diff Against Basis

Compares the selected version with the version in the destination stream or workspace.

> **Note:** When the destination is a workspace, the comparison is performed with the version in the workspace stream, not the file in the workspace tree. These are different if the element has (modified) status in the workspace.

8. Security

This topic presents an overview of AccuRev's security-related features. We discuss and compare the following topics, and provide pointers to more detailed documentation.

Users and Groups

Each AccuRev user must have an AccuRev username (also called a "principal-name"). AccuRev maintains its own user registry in the repository; it is separate from the registry maintained by your machine's operating system (or the network).

Optionally, you can create AccuRev user groups, and add users to the groups as "members". Groups can be nested within one another; that is, a group can be a member of another group. AccuRev groups are independent of operating system groups.

AccuRev groups are used by the *Access Control List* (ACL) and *Element Access Control List* (EACL) facilities to grant or deny access to a particular resource for an entire set of users. Groups are used by the *Locks* facility to specify a set of users to which a stream lock does or does not apply.

Usernames and Groupnames

Each AccuRev group has a user-defined name. Usernames and groupnames share a "namespace" in the AccuRev repository. This means that a user and a group cannot have the same name.

User Authentication

With a few exceptions, a user cannot execute AccuRev commands unless he is authenticated as an AccuRev user. For authentication purposes, each registered AccuRev user has a username/password pair recorded in the registry. A user's password can be empty.

AccuRev automatically distinguishes two categories of users -- those who have non-empty passwords and those whose passwords are empty. The keywords *authuser* and *anyuser*, respectively, are used by the ACL facility to designate these categories.

The "AccuRev Login" User-Authentication Scheme

Most AccuRev users are authenticated by explicitly logging in to the AccuRev Server. Using the GUI or the CLI, the user logs in by entering his username and password (possibly empty). This launches a user session, which is typically of limited duration (a few hours). When the session expires, the user must login again to continue using most AccuRev commands.

A successful login command creates an encrypted *session file* that records the user's AccuRev username and password, along with the IP address of the client machine. If an AccuRev client command can be executed only by an authorized user, the command automatically sends the information in the session file to the AccuRev Server process. Thus, the user does not need to repeatedly 'remind' the AccuRev Server who (and where) he is.

If a user is already logged in, and he successfully logs in again -- as the same user or a different user -- the existing session file is overwritten.

The "Custom" User-Authentication Scheme

AccuRev also supports script-based authentication of AccuRev users. For example, customers with an LDAP authentication scheme in place at their organization could configure a script to authenticate AccuRev users via LDAP. See *Chapter 9 AccuRev Security Overview* in the *AccuRev Administrator's Guide* for further details.

Locks on Streams

Each stream can have one or more locks on it. A lock prevents a certain set of users from using the stream to create new versions of elements. That is, it prevents execution of the *Promote* command -- either promoting from the designated stream, or promoting to the designated stream, or promoting in either direction.

A lock can be absolute, applying to all users. Alternatively, you can specify that a lock applies to a particular AccuRev user, or to a particular AccuRev group. Conversely, you can specify that a lock applies to everyone *except* a particular AccuRev user, or to everyone *except* a particular AccuRev group.

Locks can also prevent reconfiguration of the contents of a stream with the include/exclude facility.

For more on locks, see *The Locks Command* on page 52.

Access Control List (ACL) Permissions

In addition to (or instead of) locks, each stream can have one or more permissions on it. Whereas a lock controls the ability to create new versions (through the *Promote* command), a permission is more general: in addition to controlling *Promote*, it controls the ability to read data from the stream, using such commands as *Annotate*, *Diff*, and *Open*. A permission also controls workspace-specific commands, such as *Update* and *Populate*.

Unlike locks, which always apply to individual streams, ACL permissions can be configured to apply to entire stream subhierarchies.

You can also create an ACL permission that applies to an entire depot. This provides a way of controlling access to all of a depot's file system data, in all streams. It also provides a way to control access to issues in a depot.

ACL Permissions and Time Considerations

ACL permissions apply to a stream regardless of any *basis time* on the stream. Similarly, ACL permissions can be placed on a *snapshot*, even though such permissions are necessarily created after the snapshot is created.

For more on ACL permissions, see the description of *The Security/ACL Subtab* on page 280.

Element ACLs (EACLs)

AccuRev also provides element-level ACLs, called "EACLs". An *EACL* is a set of permissions that controls the privileges that individual users or user groups have with respect to a particular *element*. EACLs can only be set by an AccuRev administrator using either the AccuRev GUI or the CLI.

The following table shows you where to find more information on working with EACLs in AccuRev:

If You Want To Learn...	See...
More about EACLs and the privileges you can associate with an element	The section *Element-Level Security (EACLs)* in the *AccuRev Administrator's Guide*
How to set EACLs using the AccuRev GUI	*Managing EACLs in the AccuRev GUI* on page 267
How to set EACLs using the CLI	The section *eacl* in the *AccuRev CLI User's Guide*

What Happens if You Are Denied Access?

GUI users can be denied access to elements or directories if an administrator has configured EACLs. Depending on the situation, you will either see an "Access Denied" error message, a **(no such elem)** status, or the element will be completely invisible to you, as if it does not exist:

- For commands that take element names, if you are denied access to the namespace or content, you are denied access.
- For commands that specify an eid, such as **cat**, the command will succeed if you are not denied access to that element. If you are denied access to the namespace of the element, the eid is displayed instead of the name.
- For commands that display transactions, such as **hist**, if you are denied access to the element in the transaction, the transaction is displayed, but "Access Denied" is displayed for the path name.
- For commands that take a transaction or issue, such as **promote**, if you are denied access to an element or elements in the trans or issue, the command succeeds and the elements are promoted, but the names are not displayed.
- Commands that modify workspaces, such as **update** and **pop**, will succeed even if you are denied access to some of the elements, but those elements will not be brought down to the workspace. The exception to this is if you specify the element name, such as with **pop** or **co**. In this case the command will fail as if the element doesn't exist.
- For commands such as **stat**, **dirstruct**, and **files**, where you specify an option or nothing instead of an element name, if you are denied access to either the namespace or content, nothing will be displayed for that element. For example, `stat -d` displays all the elements in the default group of the stream. If you are denied access to an element in the default group, it will not be displayed.
- When you are denied access to an element, AccuRev behaves as if the element does not exist. Depending on the specific circumstances, AccuRev will either:
  - Filter out the denied element altogether, so that there is no indication that it even exists. this is the default behavior.
  - Return an Access Denied status. This occurs when the only element in a display is denied, and a blank display would be confusing (for example, if the Show Active Files display in the StreamBrowser contains only denied elements).
  - Display a **(no such element)** status. If your access to an element or namespace changes during a session, and you are now denied access to something that was already displayed, that element or folder will appear with the **(no such element)** status.

- If you have multiple Access Denied elements in the default group, the GUI will combine these into a single Access Denied row.

For information about EACLs, see *Chapter 9 AccuRev Security Overview* in the *AccuRev Administrator's Guide*, and the command summary for *eacl* on page 104 of the *AccuRev CLI User's Guide*.

Restricting Access to Commands using Triggers

By default, any registered AccuRev user can execute any AccuRev command. Many organizations wish to restrict users' access to certain commands, such as the ability to maintain users, groups, and passwords, the ability to lock streams and create ACL permissions, and so on. Triggers provide a flexible mechanism for implementing command-based security.

Many AccuRev commands can be configured to "fire a trigger". This causes a user-defined script to execute ...

- either before the command executes (pre-operation trigger), or afterward (post-operation trigger)
- either on the client machine, or on the server machine

A pre-operation trigger can affect the execution of the command or cancel it altogether. Typically, a security-related trigger checks the identity of the user invoking the command, then decides whether or not to allow the command to proceed.

Some triggers are configured on a per-depot basis, using the *mktrig* command. These triggers monitor individual commands (*add*, *keep*, and *promote*). Three are pre-operation triggers that fire on the client machine; one is a post-operation trigger that fires on the server machine.

Other triggers are configured, on a per-depot or whole-repository basis, by placing a script in a well-known location on the server machine. These triggers monitor groups of commands, rather than individual commands.

For more on triggers, see *The Triggers Tab* on page 50, and also *Chapter 10 AccuRev Triggers* of the *AccuRev Administrator's Guide*.

Which Security Feature Should I Use?

AccuRev's security features overlap to a considerable extent. For example, when a user invokes the *promote* command, any of these mechanisms can prevent the command from proceeding:

- a lock on the source or destination stream
- an ACL permission on the destination stream, on a higher-level stream, or on the entire depot
- a read-only EACL privilege on the element being promoted
- a *pre-promote-trig* trigger script, which runs on the client machine
- a *server_preop_trig* trigger script, which runs on the server machine

How do you decide which feature to use in a given situation? There are no absolute rules, but here are some guidelines:

To script or not to script?

The trigger mechanism depends on execution of user-supplied scripts, written in Perl, Python, or some other scripting language. There's a trade-off: scripting required development time and significant expertise, but is infinitely flexible.

In many situations, you may be able to use the AccuRev software distribution's sample Perl scripts, which are designed for fill-in-the-blanks customization.

It makes sense to implement as much security as possible with locks and ACL permissions (and perhaps the sample trigger scripts), before delving into original trigger scripting.

Locks vs. ACL permissions

Roughly speaking, a lock controls the placing of data *into* a stream, whereas an ACL permission controls the reading of data *from* a stream. (There are plenty of exceptions to this general rule.)

Both locks and permissions can be targeted at specific users or groups. The ACL is more flexible: you can create any number of permissions for the same stream, but only limited number of locks.

ACLs vs. EACLs

ACLs govern access to depots and streams, with broad *all* or *none* rights. EACLs govern access to individual elements in the depot, and provide finer-grained privileges for an element: Full, Allow, Read Only, and Deny. Note that if the ACL on a stream is set to none for a given user, you cannot set EACLs for elements in that stream for that user.

Running trigger scripts: client machine vs. server machine

Running trigger scripts on the client machine conserves networking and server resources. But keep in mind that all client machines must have copies of the scripts (or must have network access to a central script repository).

Running trigger scripts on the server machine provides administrative simplicity and centralized logging.

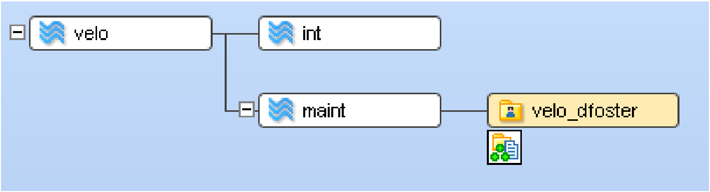
Managing EACLs in the AccuRev GUI

This section describes how to manage EACLs in the AccuRev GUI. You can also manage EACLs using the CLI. (Changes that you make using one interface are reflected in the other.) See the section *eacl* in the *AccuRev CLI User's Guide* for more information on using the CLI. See *Element-Level Security (EACLs)* on page 60 of the *AccuRev Administrator's Guide* for more detailed information on EACLs, permissions, and privileges.

Effect of Setting Permissions on an Element

By default, all users have the Allow privilege for elements under AccuRev control. Generally speaking, you use the procedures described in this section to assign a more restrictive privilege (like Read Only or Deny, for example) to an element, or to restore a user's default privileges.

When you specify a privilege for an element, that setting is reflected throughout the depot. Consider the following simple depot:



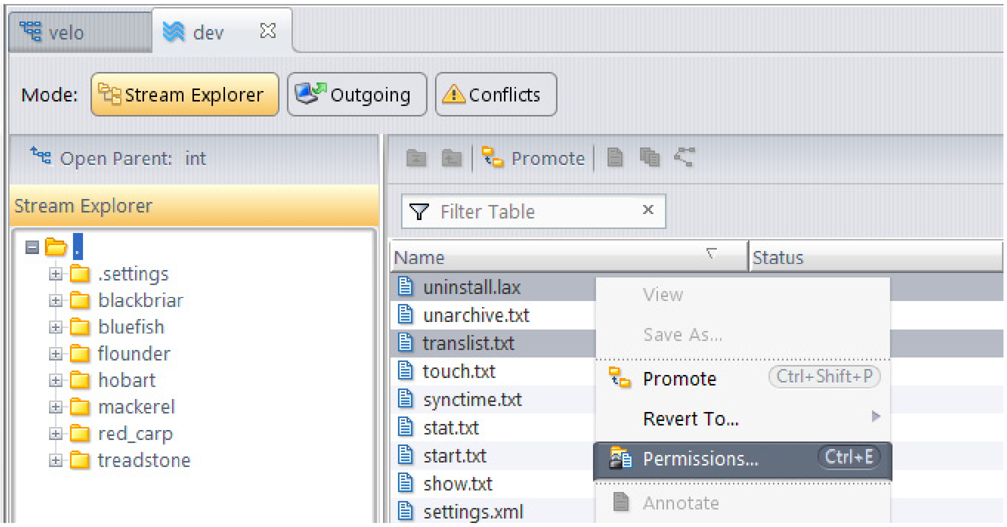
If you change the privilege for an element in the **velo_dfoster** workspace, that change is automatically reflected in all the other streams in the depot; the same is true for changes made to an element's privileges in, say, the **int** stream. In other words, you do not have to promote an element or update a workspace to propagate changes to its privileges throughout the depot.

Setting Permissions for an Element

Use this procedure to set permissions for an element. If you want to control access to a stream, use ACLs.

To set permissions for an element:

1. In the File Browser, select the elements whose permissions you want to set.



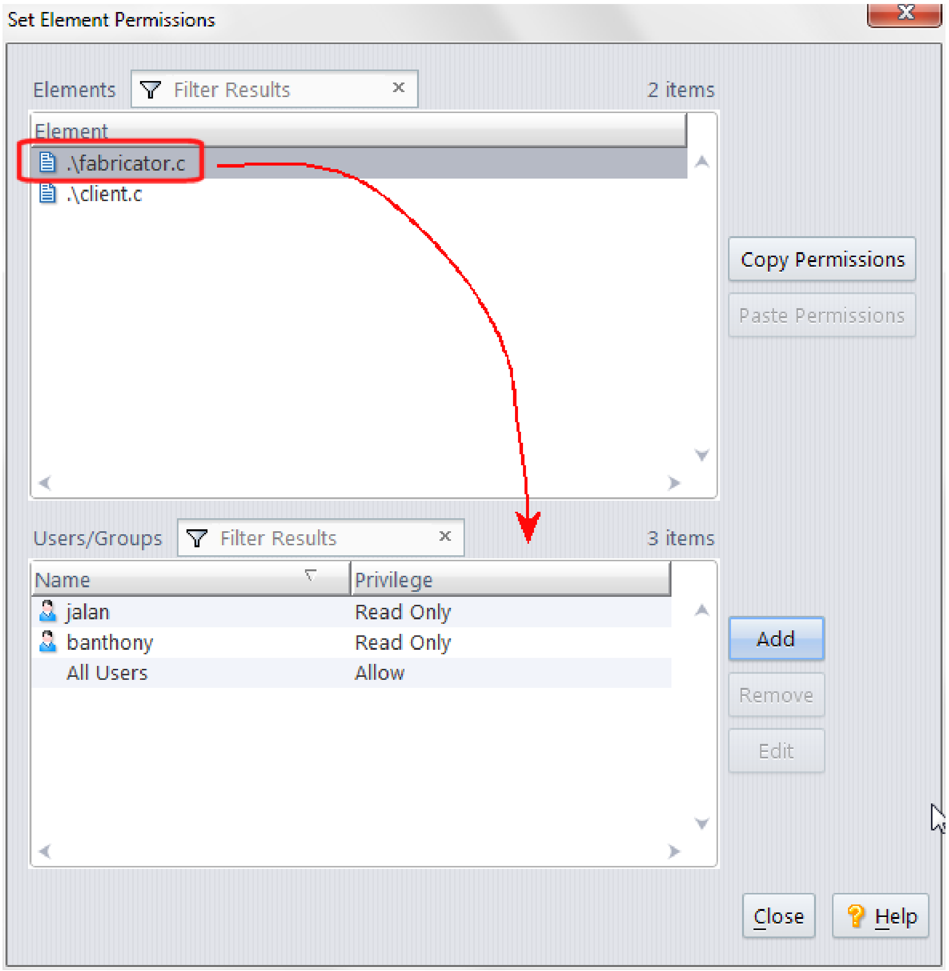
2. Right-click one of the selected files and choose **Permissions...** from the shortcut menu.

   Alternatives: Use the **Ctrl+E** accelerator key, or choose **Actions** > **Permissions...** from the main menu.

   The Set Element Permissions dialog box appears.

3. In the Elements list box, select the element whose privileges you want to modify.

   Tip: You can select only one element at a time.



Set Element Permissions

Elements — 2 items

Element
.\fabricator.c
.\client.c

Users/Groups — 3 items

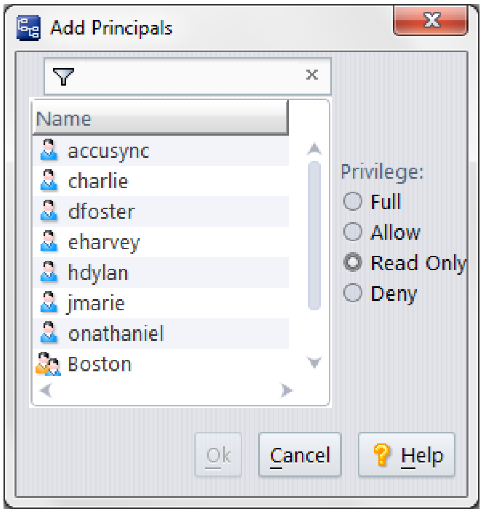
Name	Privilege
jalan	Read Only
banthony	Read Only
All Users	Allow

The Users/Groups list box displays the principals who currently have access to the element you selected, and the privileges specified for that element.

Note: The "All Users" name means just that -- all users and groups defined in the depot. Exceptions (like users *banthony* and *jalan*) are called out separately. In this example, all users have Allow privileges for **forms.xml**, except users *banthony* and *jalan*, who have Read Only access for that element.

4. Click the **Add** button.

   The Add Principals dialog box appears.



Note that the list does not include any principals for whom privileges have already been defined for the selected elements. In this example, users *banthony* and *jalan* do not appear because privileges have already been set for them for the selected element. (Use the Edit action to change existing privileges. See *Editing Permissions* on page 272 for more information.)

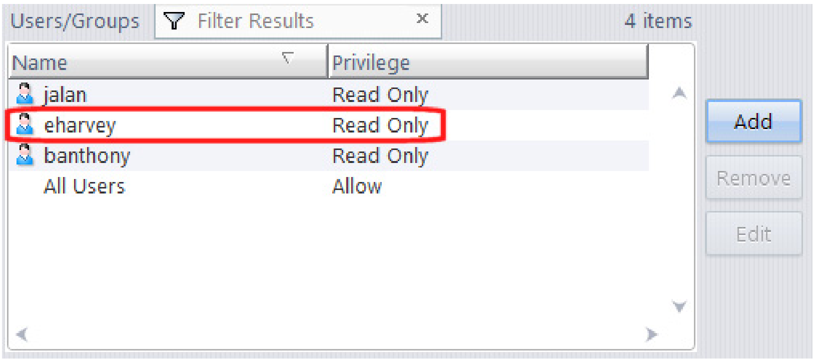
5. Select the principal whose privilege you want to set for the element selected in *Step 3*.

   Note: Changes that you make to permissions take effect as soon as you click the **Ok** button in the Add Principals dialog box.

6. Select the privilege you want to assign and click the **Ok** button.

   If you selected a directory in *Step 1*, AccuRev asks whether you want to apply the privilege recursively to all elements in that directory. See *Setting Permissions on a Directory* on page 271 for more information on this topic.

   The change you made appears in the Users/Groups list box. Here, for example, we see that the privilege for user *eharvey* has been changed from Allow to Read Only for the element selected in *Step 3*:



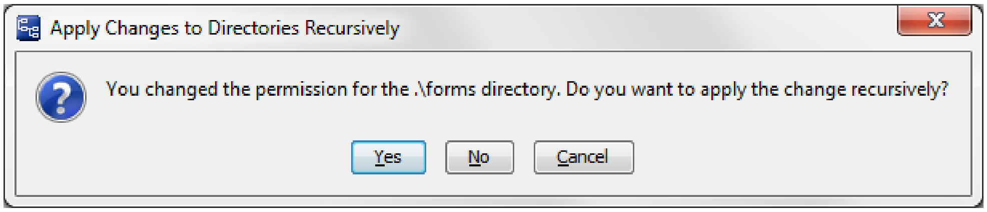
Name	Privilege
jalan	Read Only
eharvey	Read Only
banthony	Read Only
All Users	Allow

7. If you selected more than one element in *Step 1*, return to *Step 3* and repeat this procedure for the remaining elements.

   Otherwise, click **Close**.

Setting Permissions on a Directory

When you set permissions on a directory, AccuRev displays a dialog box that allows you to specify whether or not you want these permissions to be applied recursively:



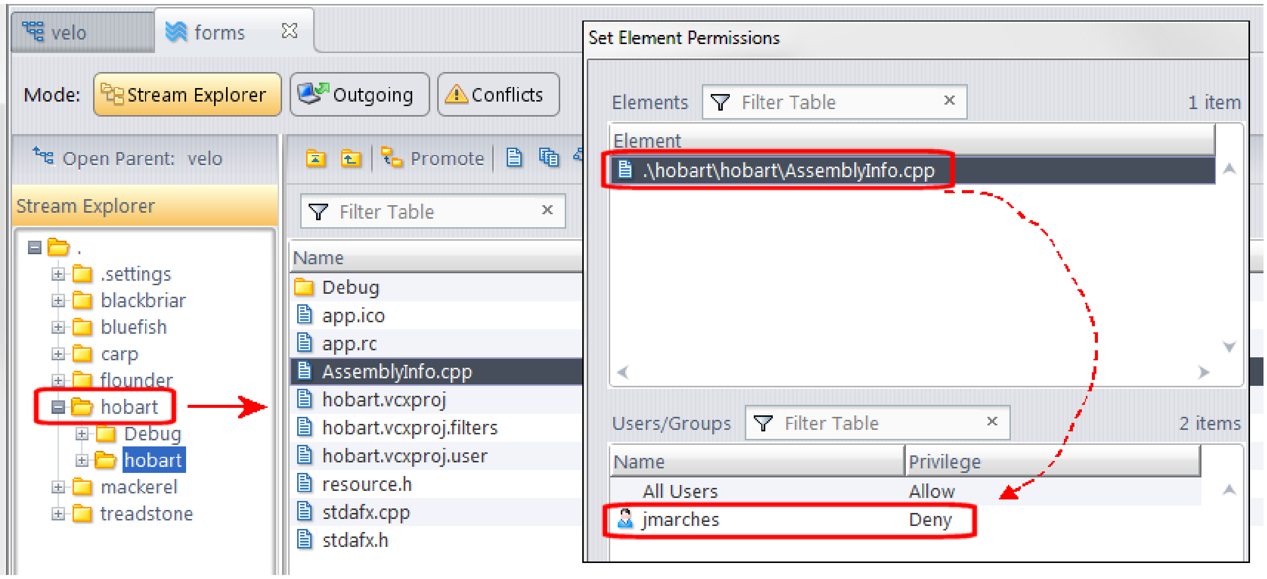
Apply Changes to Directories Recursively

You changed the permission for the .\forms directory. Do you want to apply the change recursively?

Choose:

- **Yes** to apply the permission to the selected directory, all child directories, and all elements within those directories.
- **No** to apply the permission to the selected directory only.
- **Cancel** or click the X button on the dialog box to cancel the operation without making any changes.

Consider the following example. Here, the Deny permission was set recursively on the **/hobart** directory for user *jmarches*. You can see that the Deny permission was set for elements within the **/hobart/hobart** directory, as with the **AssemblyInfo.cpp** file:



Set Element Permissions

Elements — 1 item

Element
.\hobart\hobart\AssemblyInfo.cpp

Users/Groups — 2 items

Name	Privilege
All Users	Allow
jmarches	Deny

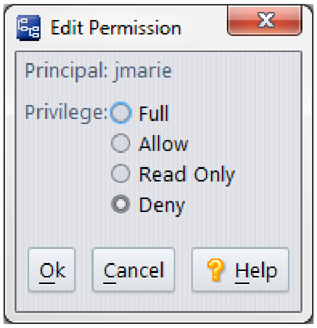
Editing Permissions

Use this procedure when you want to edit the permissions associated with an element.

1. Display the Set Element Permissions dialog box. See *Setting Permissions for an Element* on page 268 if you need help with this step.
2. In the Elements list box, select the element whose permissions you want to edit.

   The Principals list box displays all of the users and groups who currently have a privilege set for that element.
3. In the Principals list box, select the user or group whose privilege you want to modify.
4. Click the **Edit** button.

   The Edit Permission dialog box appears. The user's current privilege is selected by default.



Edit Permission

Principal: jmarie

Privilege: ○ Full
○ Allow
○ Read Only
● Deny

5. Select the desired privilege and click **Ok**.

   The changes you make are applied immediately.

Copying Permissions

Use this procedure when you want to copy the permissions associated with one element and apply them to another element.

Note: When you paste, any existing permissions associated with the target element are overwritten.

1. Display the Set Element Permissions dialog box. See *Setting Permissions for an Element* on page 268 if you need help with this step.
2. In the Elements list box, select the element whose permissions you want to copy.
3. Click the **Copy Permissions** button to copy that element's permissions.
4. In the Elements list box, select the element to which you want to apply the permissions you copied in *Step 3*.
5. Click the **Paste Permissions** button to paste the copied permissions to the selected element.

   The changes you make are applied immediately.

Removing Permissions

Use this procedure to remove permissions for an element from one or more users.

Note: When you remove a privilege that you explicitly set for a user or group, that privilege reverts to the value set for any group to which that user or group belongs. Note, however, that the default behavior for an element whose ACL does have any permissions defined for a user is to deny access to that user.

1. Display the Set Element Permissions dialog box. See *Setting Permissions for an Element* on page 268 if you need help with this step.
2. In the Elements list box, select the element whose permissions you want to modify.

   The Users/Groups list box displays all of the users and groups who currently have a privilege set for that element.
3. In the Users/Groups list box, select the user or group whose privilege you want to remove for this element.

   Tip: Use **Shift**+click or **Ctrl**+click to select multiple users or groups.
4. Click the **Remove** button.

File Locking

Parallel development is flexible and powerful, but it is not appropriate for every situation. Some organizations don't like the extra steps involved in merging, even though merging is largely automated. Some files cannot be merged, because they are in binary format. (The merge algorithm handles text files only, not binary files such as bitmap images and office-automation documents.)

Serial Development and Parallel Development

Accordingly, AccuRev supports both *serial development*, through its **exclusive file locking** feature, and *parallel development*. Exclusive file locking can be implemented at the depot level, the workspace level, or the element level (when you place the file under version control or when you create a new version).

The serial development model places more restrictions on users in the edit stage, but it eliminates the merge stage altogether.

Example:

Here's a standard scenario, in which all the workspaces are in serial-development mode:

1. A user starts working on a file by specifying it in a *Send to Workspace* ("checkout") or A*nchor* command. The file changes from being read-only to writable.
2. AccuRev places an exclusive file lock on the file. This prevents the file from being processed with *Send to Workspace*, *Anchor*, or *Keep* in other workspaces.
3. The user can edit and K*eep* any number of private versions of the file in his workspace. Then, the user P*romote*'s his most recently kept version to the backing stream. The exclusive file lock guarantees that no *Merge* will be required before this promotion.
4. After *Promote* records the new version in the backing stream, things return to the initial state: AccuRev releases the exclusive file lock, and the file returns to read-only status in the user's workspace.

5. A user in any workspace can now *Send to Workspace* or A*nchor* the file, which starts the exclusive-file-locking cycle again.

If exclusive file locking applies to a file element, and the element is not currently active in any sibling workspace:

- It is maintained in a read-only state in your workspace tree.
- If the element is *active* in a sibling workspace, you cannot make the element active in your workspace.
- If the element is not active in a sibling workspace, you can make it active in your workspace with the *Send to Workspace* or *Anchor* command. This makes the file in your workspace tree writable. (You must do this before you can use *Keep* to create a new version.) You can also make an element active with *Revert*, *Rename*, *Defunct*, *Paste* (after a *Copy*), or the CLI command **undefunct**.

  Note: In this context, workspaces are considered siblings if they promote to the same stream. If the stream hierarchy includes pass-through streams, workspaces can be siblings even if they have different parents.

An exclusive file lock on a file element is released when active development on the file ends in that workspace:

- A *promote* command sends your private changes to an element from your workspace stream to the backing stream.
- A *purge* command discards your private changes to an element.

Either way, the workspace returns to using a backing-stream version of the element.

The Limited Effect of an Exclusive File Lock

Exclusive file locking does not freeze an element completely:

- The lock applies only within the scope of a particular backing stream. It doesn't affect other backing streams and the workspaces based on them.
- The lock acquired through workspace-level or depot-level exclusive file locking applies only to workspaces in serial-development mode. Users in parallel-development-mode workspaces can make changes and promote the changes to the backing stream.
- A lock placed on an individual file element in a workspace applies to all sibling workspaces, but not to other workspaces.
- The lock doesn't prevent the current version in the backing stream from being promoted to higher-level streams.

Exclusive file locking does not prevent any user from modifying any file with a text editor or IDE. AccuRev encourages users in serial-development-mode workspaces to "ask permission first": it maintains files in a read-only state, and makes a file writable when a user executes a *co* or *anchor* command on it. But users can modify a file "without asking permission", by changing the access mode (UNIX: *chmod* command, Windows: *attrib* command or *Properties* window) and then editing it. Such "unauthorized" changes can't be sent to the AccuRev depot, though: the exclusive file lock disallows a *Send to Workspace, Anchor,* or *Keep*.

Anchor-Required Workspaces

AccuRev also offers a less-restrictive variant of exclusive file locking. **Anchor-required** workspaces allow parallel development, with multiple users modifying the same file at the same time (in their own workspaces). In an anchor-required workspace, *all* elements are maintained in a read-only state when you are not actively working on them. Using such a workspace is similar to working with exclusive file locking, except that you are not constrained by elements' activity in sibling workspaces:

- You can make a file element active with the *Send to Workspace* or *Anchor* command. This makes the file writable. You must do this before you can use *Keep* to create a new version.
- You can also make an element active with *Revert*, *Rename*, *Defunct*, *Paste* (after a *Copy*), or the CLI command **undefunct**.

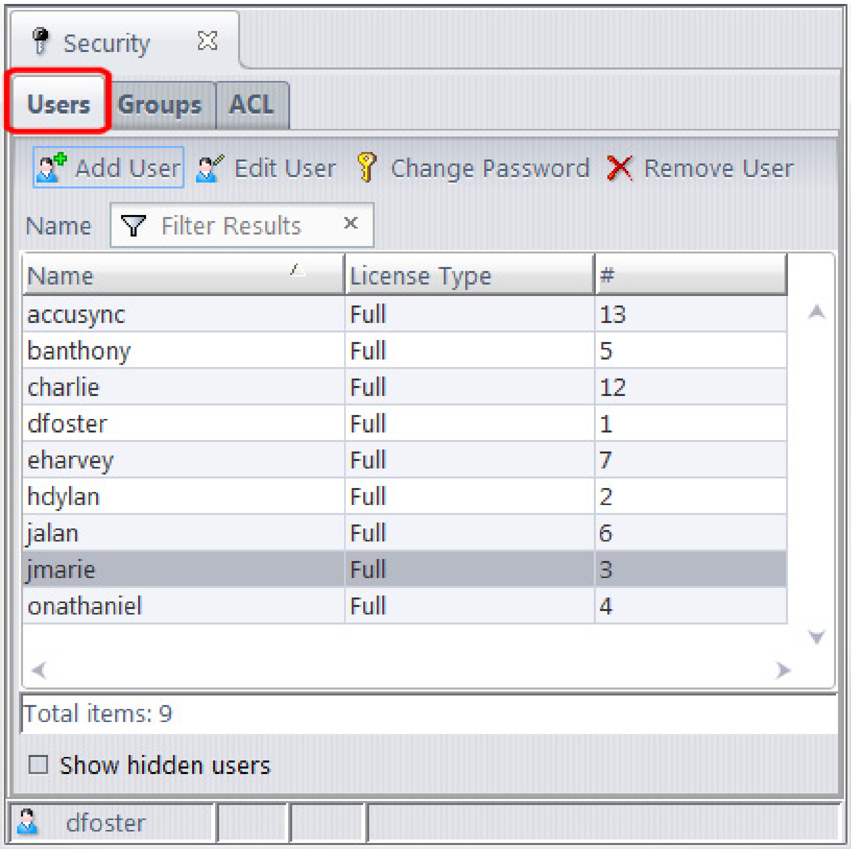
The Security/Users Subtab

The Users subtab of the Security tab displays the repository's set of registered AccuRev *users*, and provides commands for maintaining the set of users. Various security (and non-security) features involve users:

- Before executing any AccuRev commands, you must *Login* as a particular user.
- Access Control List (ACL) entries can apply to particular users or to particular groups.
- AccuWork issue records can contain fields whose value must be a particular AccuRev user.

Opening a Security/Users Subtab

Choose *Admin* > *Security* from the GUI main menu to display the Security tab. Then, click the **Users** subtab.



Name	License Type	#
accusync	Full	13
banthony	Full	5
charlie	Full	12
dfoster	Full	1
eharvey	Full	7
hdylan	Full	2
jalan	Full	6
jmarie	Full	3
onathaniel	Full	4

Total items: 9

☐ Show hidden users

Security/Users Subtab Layout

The Users subtab contains a table with these columns:

Status

(appears if *Show hidden users* is checked) **Inactive** indicates that the username has been removed, and can't be used in a *Login* command, or in the definition of an *ACL* entry or a *lock*. A removed user can subsequently be *Reactivated*.

Name

The username. You can change a username (for example, from **derek** to **dpost**) without disrupting AccuRev's bookkeeping, because it tracks user identities by their unique, immutable integer IDs.

A user cannot have the same name as any other user, or the same name as any group. See *User-Specified Names for AccuRev Entities* on page 23.

License Type

The type of license associated with the user. Values include Full (AccuRev, AccuWork, and GitCentric), AccuWork, and GitCentric Only.

#

The integer that uniquely identifies this user. This number cannot be changed. **Note:** The same "namespace" of integer IDs is shared by users and groups.

Working in a Security/Users Subtab

You can use this subtab to activate/deactivate users, and to perform operations that maintain the user registry.

Controlling Which Users are Displayed

☑Show hidden users

If checked, the table includes users that have been deactivated with the *Remove* command. This also adds a Status column.

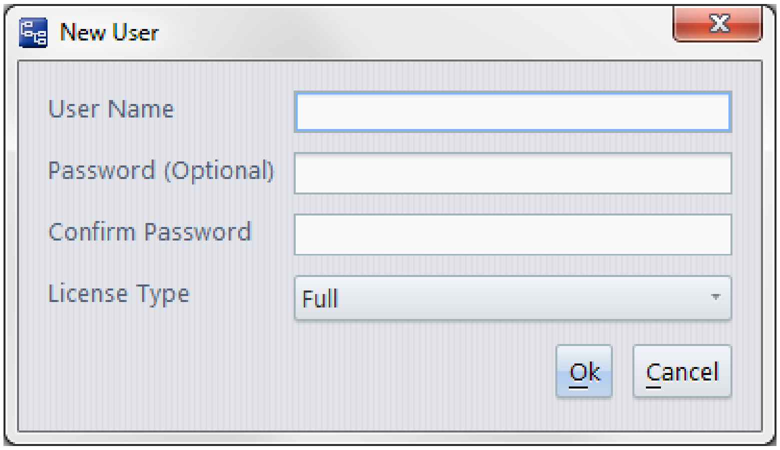
Creating a New User

Note: In order to successfully add a user, there must be at least one available user license.

To create a new user:

1. Click the **Add User** button () on the toolbar.

   The New User dialog box appears.



User Name

Password (Optional)

Confirm Password

License Type: Full

2. In the **User Name** field, specify the new user’s name.
3. In the **Password (Optional)** field, optionally, specify a password; reenter the password in the **Confirm Password** field.
4. Specify the type of license you want to associate with this user.
5. Click **Ok**.

Operating on a Selected User

You can choose any of the following commands from the context menu of a selected user:

Edit User

Use this command to change either the user’s name or license type. See *User-Specified Names for AccuRev Entities* on page 23.

Change Password

Change this user's password. **Note:** AccuRev allows you to change any user’s password, not just your own. But your organization might use a trigger to disallow all but a few administrative users from changing another user's password.

See *User-Specified Names for AccuRev Entities* on page 23.

Reactivate User

Reactivate a user who has been deactivated with *Remove User*. In order to succeed, there must be at least one available user license.

Remove User

Deactivate the selected user. No one can Login as that user, but the user still appears in the repository's historical records, such as the History Browser. You can use the *Reactivate User* command to bring back a removed user.

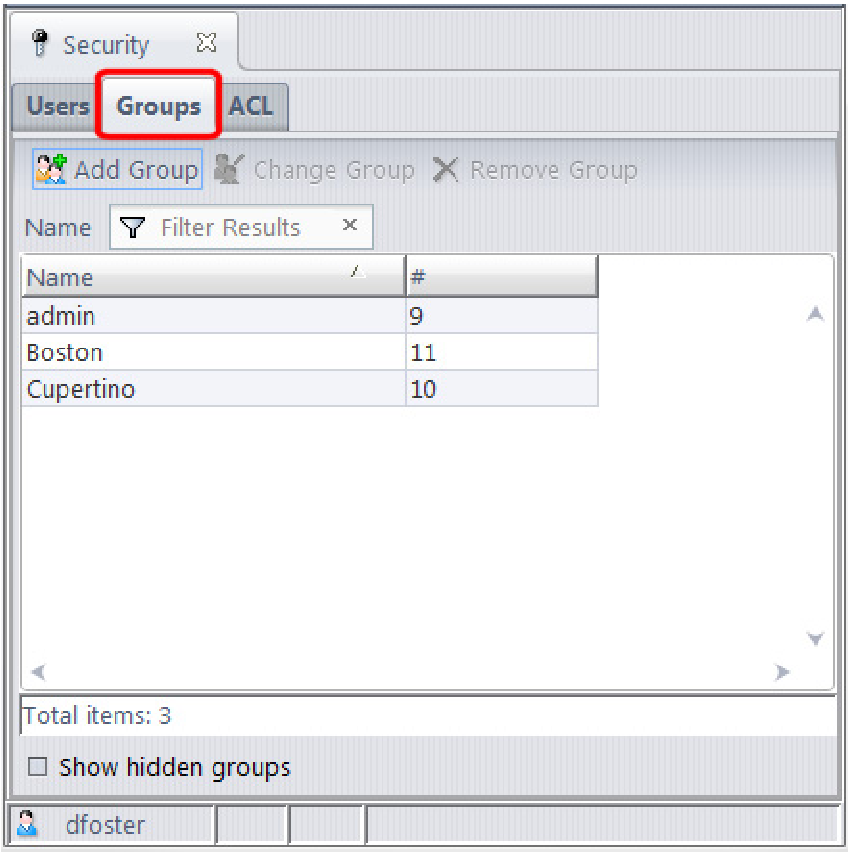
The Security/Groups Subtab

The Groups subtab of the Security tab displays the repository's set of user *groups*. Individual *users* can be members of any number of groups. Likewise, a group can belong to -- that is, be a subgroup of -- any number of other groups.

Access Control List (ACL) entries can apply to particular users or to particular groups.

Opening a Security/Groups Subtab

Choose *Admin* > *Security* from the GUI main menu to display the Security tab. Then, click the **Groups** subtab.



Name	#
admin	9
Boston	11
Cupertino	10

Security/Groups Subtab Layout

Status

(appears if *Show hidden groups* is checked) **Inactive** indicates that the group has been *Removed*, and can't be used in the definition of an *ACL* entry or a *lock*. A removed group can subsequently be *Reactivated*.

Name

The group name. You can change a group name (for example, from **sptgrp** to **sport_group**) without disrupting AccuRev's bookkeeping, because it tracks group identities by their unique, immutable integer IDs.

A group name can be up to 96 characters long. A group cannot have the same name as any other group, or the same name as any user. See User-Specified Names for AccuRev Entities on page 23.

#

The integer that uniquely identifies this group. This number cannot be changed. **Note:** The same "namespace" of integer IDs is shared by users and groups.

Working in a Security/Groups Subtab

You can use this subtab activate/deactivate groups, and to perform operations that maintain groups and their memberships.

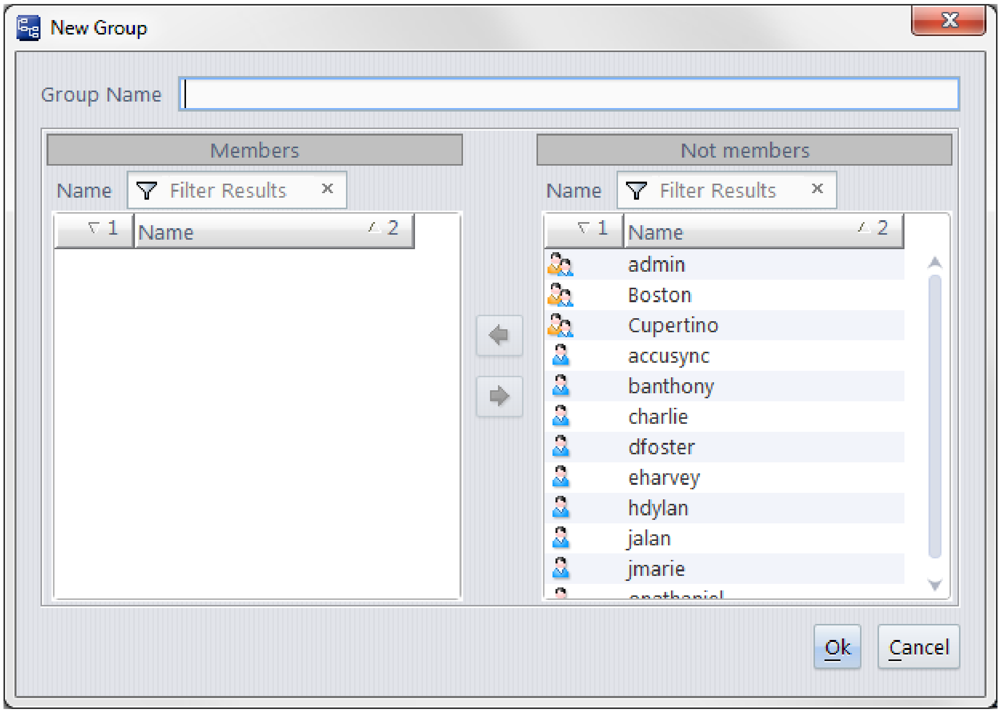
Controlling Which Groups are Displayed

☑Show hidden groups

If checked, the table includes group that have been deactivated with the *Remove* command. This also adds a Status column.

Creating a New Group

Use the **Add Group** toolbar button to create a new group.



Add Group

A New Group dialog appears, in which you specify the group's name and its members.

To specify a group's membership, select one or more users and/or groups, and click the ← or → button to move them between the **Members** and **Not members** lists.

A user can be a member of any number of groups. Likewise, a group can belong to (be a subgroup of) any number of groups.

Tip: You can filter both lists. Filters use an "includes" logic to display only those principal names that include the characters you enter.

Operating on a Selected Group

You can choose any of the following commands from the context menu of a selected group:

Change Group

Change the name that AccuRev associates with this group identity, or change the group's membership. A Change Group dialog appears, works the same way as the New Group dialog.

A user can be a member of any number of groups. Likewise, a group can belong to (be a subgroup of) any number of groups.

Reactivate Group

Reactivate a group that has been deactivated with *Remove Group*.

Remove Group

Deactivate the selected group.

The Security/ACL Subtab

The ACL subtab of the Security tab displays the repository's set of access control list (ACL) entries, also called **permissions**. Each stream and depot in the repository can have any number of permissions. **Note:** For this command, "stream" can be a dynamic stream, a workspace stream, or a snapshot.

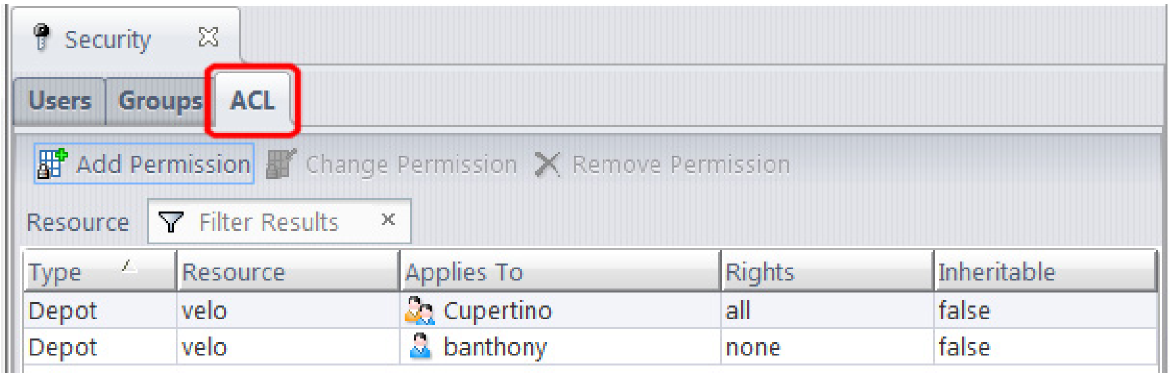
Each permission controls the ability of a particular user, or a particular set of users, to access that stream or depot.

Opening a Security/ACL Subtab

Choose *Admin* > *Security* from the GUI main menu to display the Security tab. Then, click the **ACL** subtab.

Security/ACL Subtab Layout

The table in the ACL subtab includes these columns:



Type	Resource	Applies To	Rights	Inheritable
Depot	velo	Cupertino	all	false
Depot	velo	banthony	none	false

Type

The type of resource to which the permission applies: **Stream** or **Depot**.

Resource

The name of the stream or depot to which the permission applies. The permission also applies to (is inherited by) the entire subhierarchy below that resource, but it can be overridden at a lower level.

Note: Because permissions are inherited down a depot's stream hierarchy, a permission on a depot is almost equivalent to one on the depot's base stream. The only difference is that a depot permission also affects access to the depot's issues.

Applies To

The user or set of users to which the permission applies: all users with passwords (identified as "authuser" in the AccuRev CLI), all users without passwords ("anyuser" in the CLI), a particular user, or a particular group.

Rights

all: the user(s) can access the resource.

none: the user(s) cannot access the resource.

Inheritable

Depot permission:

yes: the permission applies to the depot, and also to the depot's entire stream hierarchy.

no: the permission applies only to the depot, not to any of the depot's version-controlled elements.

Stream permission:

yes: the permission applies to the entire subhierarchy below the stream in the Resource column.

no: the permission applies only to the stream in the Resource column, not to any lower-level stream.

How Individual Commands Use the ACL

The following commands check ACL entries on one or more dynamic streams, workspace streams, or snapshots before proceeding. In the following, 'stream' can be a dynamic stream, workspace stream, or snapshot. If a version is being accessed from stream A, and that version is cross-linked to stream B, AccuRev checks the ACL permissions on stream A only, not on stream B.

- *Anchor*, *Defunct*, *Populate*, *Revert to Basis*, *Revert to Most Recent Version*, *Update* and File Browser searches check the current workspace.
- *Annotate*, *View*, and *Send to Workspace* check the stream of the version being accessed. *Send to Workspace* also checks the current workspace.
- *Promote* checks the stream to which the version(s) are being promoted.
- *Include from Stream* and *Clear Rule* check both streams involved in the cross-link.
- The various *Diff Against ...* commands check the streams of both versions being compared.
- *Merge* checks the workspace/stream in which you're doing the merge, as well as any stream explicitly mentioned on the command line.

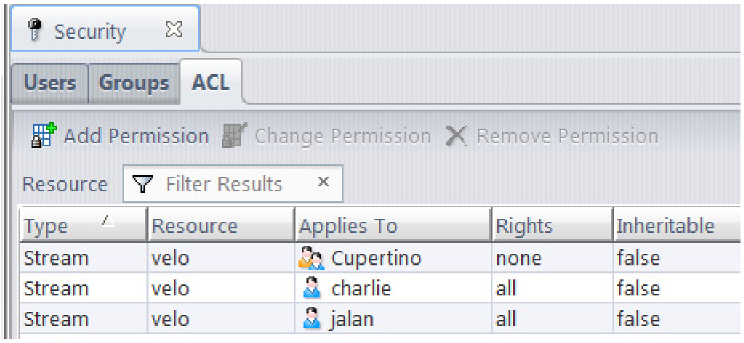
- *Remove* and *Reactivate* stream check the stream being changed.
- *New Workspace*, *New Stream*, and *New Snapshot* check the specified backing stream for the workspace/stream/snapshot being created.
- *Change Stream* checks the stream being changed (and, if appropriate, its new backing stream).
- *View Streams* checks the depot.
- In a *Stream Browser* tab, *Show History* checks the selected stream. In a *Depots* tab, *Show History* checks the selected depot.

Conflicting Permissions

Two or more permissions on a resource can apply to the same user, or to the same stream. In such cases, an **all** permission overrides one or more **none** permissions. This makes it easy to implement "all but" access controls.

Example 1: "all but" at the user level

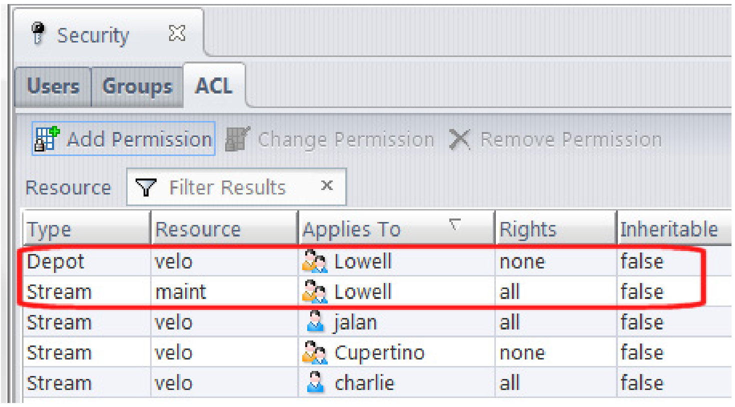
The permissions shown here prevent everyone in the **Cupertino** group -- except for users **charlie** and **jalan** -- from accessing stream **velo**.



Type	Resource	Applies To	Rights	Inheritable
Stream	velo	Cupertino	none	false
Stream	velo	charlie	all	false
Stream	velo	jalan	all	false

Example 2: "all but" at the stream level

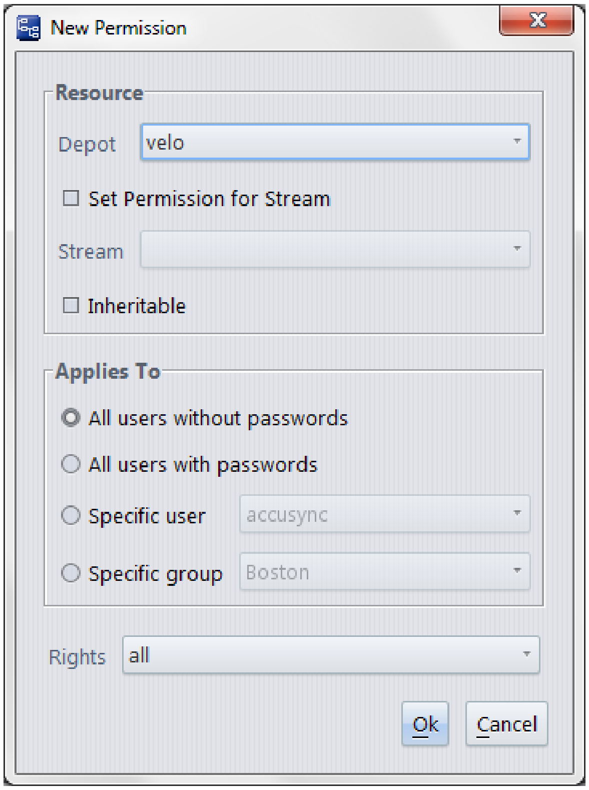
The permissions shown here restrict the **Lowell** group to working in the **maint** stream subhierarchy. Members of this group cannot work with other streams in the **velo** depot.



Type	Resource	Applies To	Rights	Inheritable
Depot	velo	Lowell	none	false
Stream	maint	Lowell	all	false
Stream	velo	jalan	all	false
Stream	velo	Cupertino	none	false
Stream	velo	charlie	all	false

Working in a Security/Access-Control Subtab

Use the following commands to maintain the set of permissions in the repository's access control list.



New Permission

Resource

Depot: velo

☐ Set Permission for Stream

Stream:

☐ Inheritable

Applies To

● All users without passwords

○ All users with passwords

○ Specific user: accusync

○ Specific group: Boston

Rights: all

Add ACL Entry

Change ACL Entry

The same dialog enables you to create a new ACL permission, or to modify an existing one.

Resource: Select one of the repository's depots from the *Depot* listbox.

- If you want the permission to apply to the depot itself, leave the *Set Permission for Stream* checkbox cleared.
- If you want the permission to apply to one of the depot's stream, check the *Set Permission for Stream* checkbox, and choose a stream from the *Stream* listbox.
- If you want the permission to apply to the entire subhierarchy below the specified stream, check the *Inheritable* checkbox.

Security Group: Use the radio buttons (and listboxes) to specify an individual user or a single group to which the permission will apply. (You can't specify multiple users or groups, but you can produce the same effect by creating multiple permissions on the same resource.)

Permission: Select **all** or **none** from the listbox.

Remove ACL Entry

Deletes the selected entries from the access control list.

The Available AccuRev Servers Dialog

(and the New Server and Edit Server dialogs)

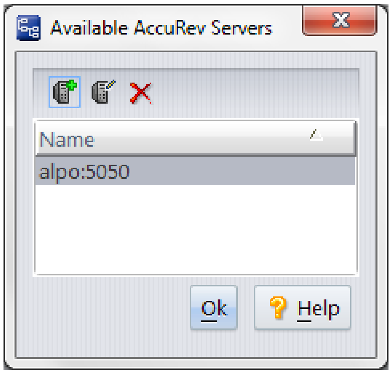
The Available AccuRev Servers dialog provides a way to maintain the list of AccuRev Servers to which you can *login*. Each *client* installation of AccuRev maintains its own AccuRev Server list, in file *<AccuRev-install-dir>/bin/acclient.cnf*.

Invoking the Available AccuRev Servers Dialog

Choose *Tools* > *Login* from the GUI main menu. Then click the 🗄 button in the *Login* dialog.

Using the Available AccuRev Servers Dialog

To switch to another AccuRev Server, select its entry, and click *Close*.



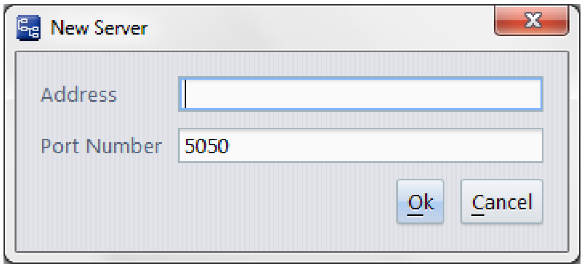
Click the Name column header to reverse the order in which the AccuRev Servers are listed.

Toolbar Commands

The toolbar includes the following commands for maintaining the AccuRev Server list:

🗄 Add Server

Identify an instance of the AccuRev Server, to which anyone using this client installation will be able to log in. A dialog appears, in which you identify the Server



Address

Port Number 5050

The Address can be a name (e.g. *abbot* or *abbot.myfirm.com*) or an IP address (123.156.189.047). Be sure to specify the correct port number for that Server. No attempt is made to check the validity of the new entry.

Edit Server

Modify the address and/or port number of the selected entry in the list. No attempt is made to check the validity of the modified entry.

✕ Remove Server

Remove the selected entry from the list. This does not affect that AccuRev Server process; it merely removes the Server from this client installation's list.

Note: If you remove the server that you are currently logged in to from the list of available servers, then after clicking **OK** and dismissing the Available AccuRev Servers dialog box, you must log in to a different server instead of clicking the **Cancel** button in the Log in to AccuRev Server dialog box. Otherwise, the current server will remain in the list of available AccuRev servers.

9. AccuWork

AccuWork is AccuRev's issue management facility. This chapter describes AccuWork features for end users, and introduces AccuWork's powerful *Change Package* capabilities. For detailed information on Change Packages, see *Chapter 10 Change Packages*.

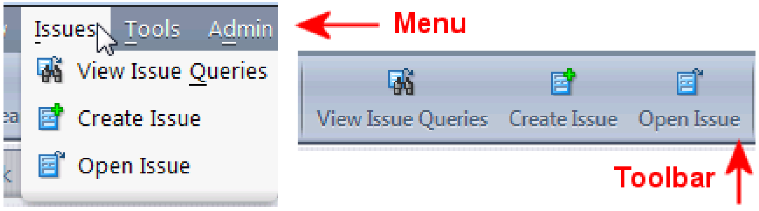
Configuring AccuWork

Before you can use AccuWork, it must be configured by an administrator. This is covered in *Appendix A The Schema Editor (Administrators Only)*.

Note that the specific layout and available fields that you see will depend on how your Administrator decides to configure the AccuWork forms.

Accessing AccuWork

The main way to access AccuWork functionality is from the either the **Issues** menu or the AccuRev toolbar:



- **View Issue Queries** lets you create, run, and edit queries, and work with the issues those queries return. See *Creating AccuWork Queries* on page 309 for more information.
- **Create Issue** lets you create an issue in AccuWork. See *Creating an Issue* on page 287 for more information.
- **Open Issue** lets you open an existing AccuWork issue. See *Modifying an Existing Issue* on page 288 for more information.

Working with Issues

This section describes how to create, edit, and clone AccuWork issues. See *Issue Form* on page 294 if you are interested in details about the AccuWork Issue form itself.

Creating an Issue

To create an issue in AccuWork:

1. Click the **Create Issue** button ( ) on the AccuRev main toolbar.

   Alternative: Select **Issues** > **Create Issue** from the AccuRev main menu.

The Issue form appears.

2. Enter data in all of the required fields.

   Note: While required fields have red labels, whether or not a field is required is determined by your AccuRev administrator and how he or she has configured your site's AccuWork schema. See *Issue Form* on page 294 for more information on this topic.

3. Enter any additional information you have.

4. When you are done, click the **Save** button (💾).

Modifying an Existing Issue

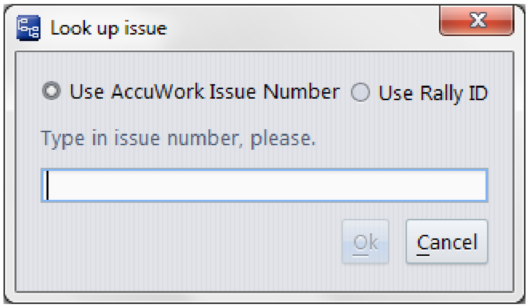
To modify an existing issue:

1. Click the **Open Issue** button (📋) on the AccuRev main toolbar.

   Tip: The **Open Issue** button is also available on the Query Results pane toolbar. See *Query Tab Layout* on page 310 for more information.

   Alternative: Select **Issues** > **Open Issue** from the AccuRev main menu.

   AccuWork displays the Look Up Issue dialog, which prompts you to enter the issue record's ID. This can be an integer (in the case of AccuWork) or some other unique identifier (some third-party ITSs use alphanumeric IDs, for example).



Look up issue

● Use AccuWork Issue Number ○ Use Rally ID

Type in issue number, please.

   Tip: If you have implemented third-party issue tracking system (ITS) keys in your AccuWork schema, this dialog box displays a pair of radio buttons that let you indicate whether you want to use AccuWork issue numbers or third-party ITS keys when looking up issues. See *Using Third-Party Keys in the AccuRev GUI* on page 408.

2. If applicable, change the default ITS.

3. Enter the issue number and click **Ok**.

   The Issue form for the issue you specified appears. See *Issue Form* on page 294 for more information on the form.

Notes:

Multiple depots

A typical AccuRev repository contains multiple depots. Each depot can have its own AccuWork configuration. In each depot, the issue records are numbered sequentially: 1, 2, 3, ... The **Create Issue** or **Look Up** operation applies to the current depot.

Alternate lookup field

AccuWork can be configured to use a lookup field other than the default (field name *issueNum*, field label *Issue*). In this case, the label for the configured field appears in the prompt window, and you must enter a value that exactly matches (except for uppercase/lowercase) some issue record's value of that field.

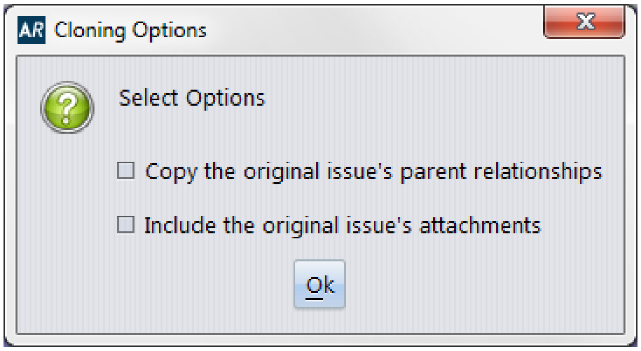
Cloning an Issue

Cloning an issue creates a new issue record based on the issue you have chosen to clone. You might want to clone an issue when it contains a lot of information that is relevant to the new issue you want to create and you do not wish to recreate this information manually.

To clone an issue:

1. Select the issue you want to clone. You can select an issue by:
   - Opening it. See *Modifying an Existing Issue* on page 288 for more information.
   - Choosing it in the Query Results pane. See *Working in the Query Results Pane* on page 314.
2. Click the **Clone Issue** button () on the Issue form or Query Results pane toolbar.

   If the original issue has any attachments or parent issues, AccuWork asks you whether or not you want to create the new issue with this information in the Cloning Options dialog box:

   □ Copy the original issue's parent relationships

   □ Include the original issue's attachments

   Make your choices and click **Ok**.

   In either case, the new issue appears on its own tab. The **Short Description** field, if present in your schema, includes a prefix of **[Clone of <*issue number*>** to help distinguish it from the original issue.
3. Make any changes required and then click either the **Save** button () or the **Save & Close** button () to create the issue.

What Information is Cloned

When you clone an issue, under the covers AccuWork:

1. Initializes the new issue based on the "initialization of new issue" conditions specified on the Validation tab of the Schema Editor.
2. Copies all values from the original issue record to the clone.
3. Initializes the cloned issue based on the "initialization of cloned issue" conditions, if any, specified on the Validation tab of the Schema Editor.
4. Performs any other issue validation specified on the Validation tab of the schema editor.

The cloned issue is created as a new issue only after you click the **Save** button (💾) or the **Save & Close** button (💾✖).

By default, AccuWork clones all of the fields defined in your schema except issueNum (including third-party ITS keys, if you are using them) and transNum. The new issue record is created with the same values as those in the original issue.

For example, imagine you have submittedBy and dateSubmitted fields in your schema, which record the username of the user who created an issue and the date it was created. By default, the issue record AccuWork creates during cloning will also get those values -- even if the clone operation was performed by a different user at another date and time. If you want, you can use the "initialization of cloned issue" validation condition to set new values for these fields -- and any other fields you wish. In this case, you might want to set submittedBy to "CUR_USER" and dateSubmitted to "CUR_DATE". Other examples for values you might wish to reset are an issue's state or status. See *Validation Tab* on page 392 to learn more about using the schema to specify issue record initialization and validation.

> **Note:** Remember that the content of Issue forms, including some subtabs, is determined by how your AccuRev administrator has configured your AccuWork schema -- you might not see an Attachments tab on the Issue form at your site, for example. For general information about the Issue form, see *Issue Form* on page 294.

Information That is Never Cloned

In addition to the issueNum and transNum values, AccuWork never copies the following:

- Change package information
- Issue history
- Subtasks
- Relationships
    - Tracking issues
    - Dependent issues
    - Duplicate issues

In addition, the value of the shortDescription field (if present in your schema) is given a prefix of **[Clone of <*issue number*>]** to help distinguish it from the original issue. You can, of course, change the shortDescription to whatever you like.

As mentioned previously, attachments and parent/subtask relationships are created in the cloned issue only if you choose.

Creating Subtasks

AccuWork allows you to define parent/subtask relationships between issues. You might want to do this when you have a single issue (or story) that is made up of two or more individual tasks that you want to be able to manage separately. In AccuWork, you can create separate issues to represent those tasks and associate them, as subtasks, with the parent issue. For example, imagine an issue such as *User wants reporting*. In fleshing out this story, you might discover that it really consists of a number of individual requirements, each of which you want to manage as its own task. Using parent/subtasks, you might end up with the following issues:

- 23425 -- *User wants reporting* (parent issue)
  - 23426 -- *Support PDF, CSV, and HTML report formats* (subtask issue)
  - 23448 -- *Support online help* (subtask issue)
  - 23517 -- *Support custom reports* (subtask issue)

As the issue numbers suggest, parent/subtask issue relationships can be created at any time.

You can create subtasks graphically, using familiar drag-and-drop gestures, or manually, using the Subtasks panel on the Issue form.

Creating Subtasks Using Drag-and-Drop

To create a subtask using drag-and-drop:

1. In the Query Results pane of the Queries tab, locate the issues for which you want to create a parent/subtask relationship. (See *The AccuWork Query Editor* on page 316 if you need help understanding the Queries tab.)
2. Drag the issue you want to create as a subtask, and drop it on the issue you want to define as the subtask's parent.

   Tip: You always drag-and-drop the issue you want to create as the subtask. The parent issue is always the target.

Creating Subtasks Using the Issue Form

To create a subtask using the issue form:

1. Open the Issue form for the issue you want to create as either the parent or subtask.
2. Click the **Subtasks** tab.
3. If you opened the Issue form for the issue you want to create as the parent, navigate to the Subtasks toolbar.

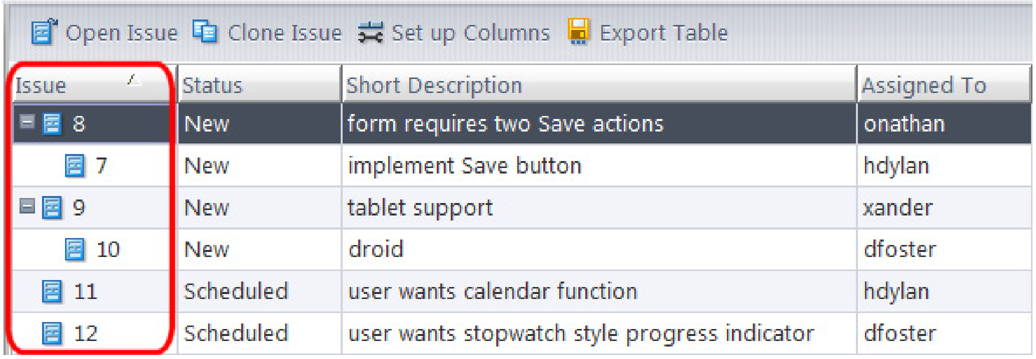
   If you opened the Issue form for the issue you want to create as the subtask, navigate to the Parents toolbar.
4. To use an existing issue to specify the parent/subtask relationship, click the **Link issue...** button ().

   AccuWork displays the Link with Issue dialog box. Enter the appropriate issue number and click **Ok** to create the relationship.
5. To create a *new* issue to specify the parent/subtask relationship, click the **Link new issue...** () button.

AccuWork displays a new Issue form. When you are done, click the **Save** or **Save & Close** button. AccuWork both creates the new issue and the parent/subtask relationship.

How Subtask Issues are Sorted

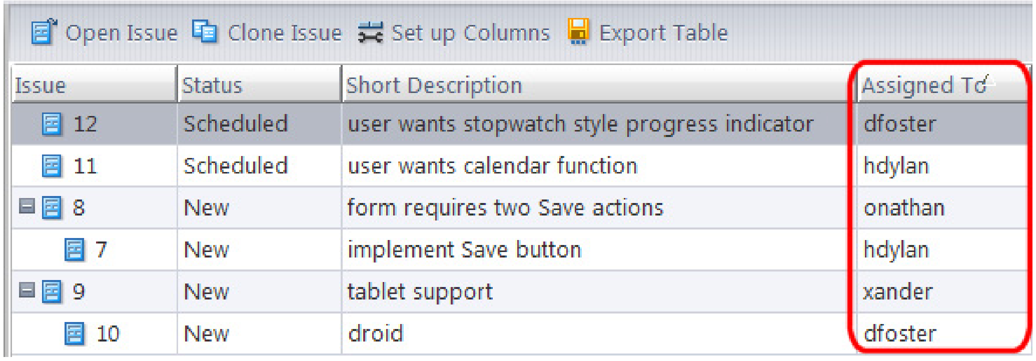
AccuWork always sorts issues using data in the *parent* issue, even if the subtask issue. Consider the following simple example, which shows the default sort order for the Query Results pane, by **Issue** number in ascending order:



Issue	Status	Short Description	Assigned To
⊟ 8	New	form requires two Save actions	onathan
7	New	implement Save button	hdylan
⊟ 9	New	tablet support	xander
10	New	droid	dfoster
11	Scheduled	user wants calendar function	hdylan
12	Scheduled	user wants stopwatch style progress indicator	dfoster

Even though issue 7 is the lowest issue in the table, it is not listed first because it is a subtask of issue 8.

Similarly, if we next sort on **Assigned To**, issue 12 is listed first because its value, *dfoster*, is lowest in the alphabet. Note that it also appears after the value *xander*, because that issue (10) is a subtask of issue 9 with which *xander* is associated.

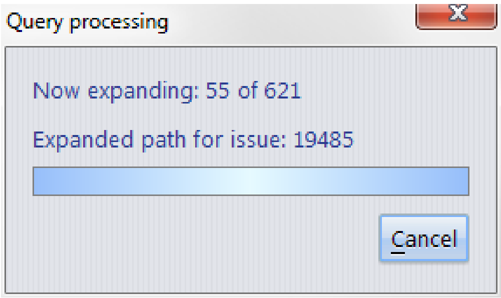


Issue	Status	Short Description	Assigned To
12	Scheduled	user wants stopwatch style progress indicator	dfoster
11	Scheduled	user wants calendar function	hdylan
⊟ 8	New	form requires two Save actions	onathan
7	New	implement Save button	hdylan
⊟ 9	New	tablet support	xander
10	New	droid	dfoster

While the sort behavior of subtask issues is intuitive, it might not be immediately apparent why sorting on a particular column, like **Assigned To** in the previous illustration, yields the results it does -- especially if the **Issue** column is obscured from view.

Query Results with Large Numbers of Subtasks

When you execute a query that has a large number of issues that are subtasks in the result set, AccuRev presents a dialog box that displays its progress in expanding the paths to all of those subtasks that match the query you ran. An example of this dialog box is shown here:



Query processing

Now expanding: 55 of 621

Expanded path for issue: 19485

- The **Now expanding:** field shows the total number of issues whose paths need to be expanded to display the subtask, and how many have already been expanded. In this example, AccuRev has expanded 55 of the 621 paths to subtask issues.
- The **Expanded path for issue:** field identifies the last issue number (or third-party key) that has been expanded.

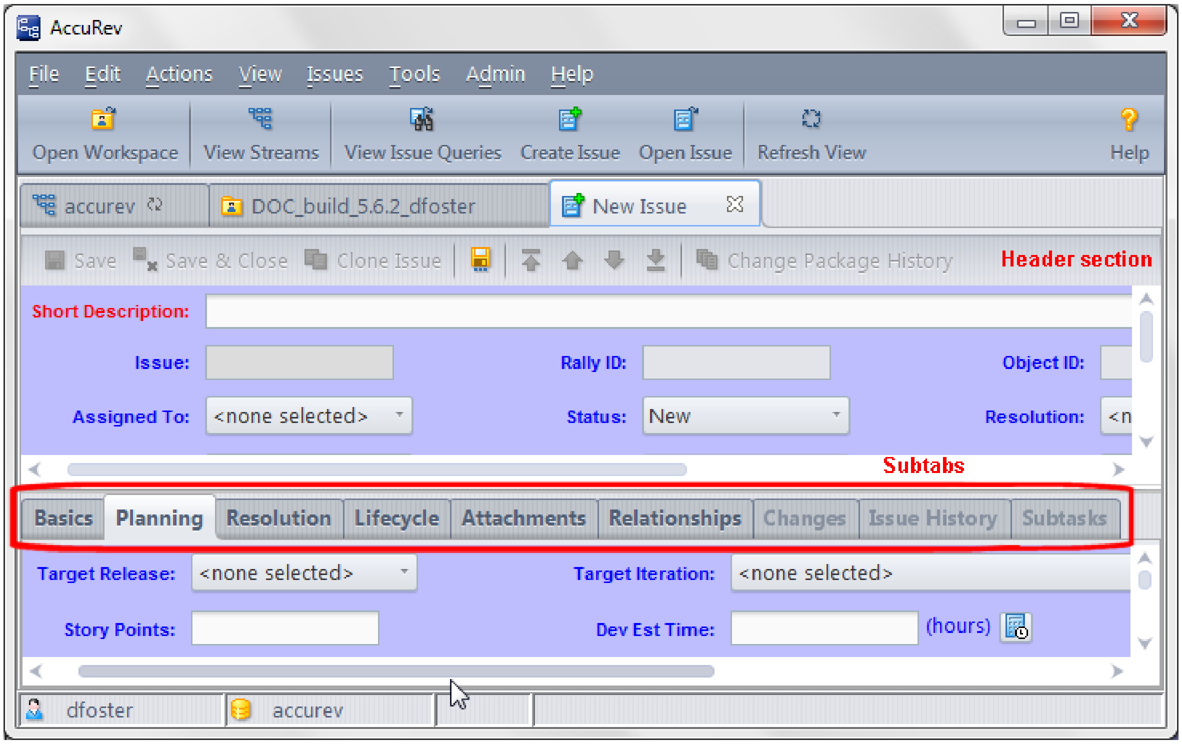
Processing a substantial number of subtask issues can take a long time. You can click **Cancel** at any time to stop the subtask issue path-expansion processing. AccuRev will display expanded paths for any issues it has processed at that time, and you can manually expand the paths for any remaining issues if you choose. Cancelling this processing does not affect the result set in any way.

Issue Form

This section describes the layout and features of the AccuWork Issue form. To learn how to create, clone, and modify AccuWork issues, see *Working with Issues* on page 287.

All AccuWork Issue forms have these features in common:

- The Issue form consists of one or more pages, which appear as subtabs within the overall GUI tab. Typically, there's a header section, which always remains visible as you switch from page to page, using the mouse.



Header section

Short Description:

Issue:

Rally ID:

Object ID:

Assigned To: <none selected>

Status: New

Resolution: <n

Subtabs

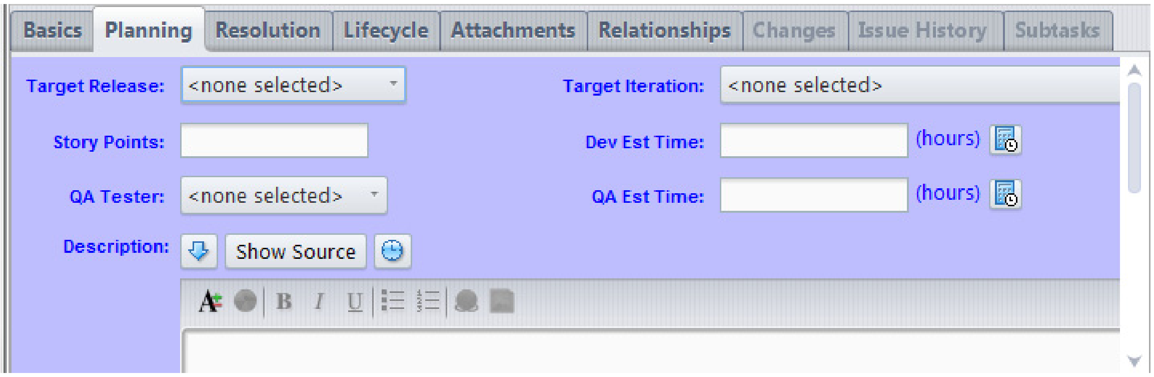
Target Release: <none selected>

Target Iteration: <none selected>

Story Points:

Dev Est Time: (hours)

- Each field is represented by a simple text box, a multiple-line text box, a multiple-choice listbox, or a more complex user-interface "widget". To the left of this widget is a descriptive label for the field. (A red label indicates a required field.)


Basics Planning Resolution Lifecycle Attachments Relationships Changes Issue History Subtasks
Show Source
B I U

Target Release: <none selected>

Target Iteration: <none selected>

Story Points:

Dev Est Time: (hours)

QA Tester: <none selected>

QA Est Time: (hours)

Description:

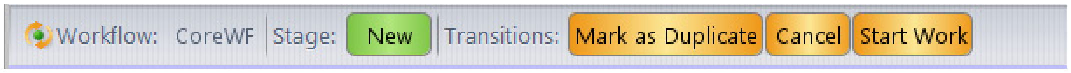
Special Subtabs

Every Issue form has two special subtabs, which appear and work differently than the standard fill-in-the-blanks "pages":

- The **Changes** subtab displays the issue record's change package, and provides options for analyzing and manipulating the change package's entries. See *The Issue Form -- "Changes" Subtab* on page 304 for more information.
- The **Issue History** subtab displays a complete breakdown of how the issue record's field values have changed over time. See *The Issue Form -- "Issue History" Subtab* on page 306 for more information.

Workflow Toolbar

If your site is configured to use the optional AccuRev workflow feature, your Issue form may display the following workflow toolbar:



AccuRev Workflow is documented in the *AccuRev® Web Interface User's Guide*, Version 2012.1 or later, but a brief summary of this toolbar is:

Workflow:

A label that identifies the name of the workflow to which the current issue is subject.

Stage:

A label icon that identifies the issue's current workflow stage. (This is not clickable.)

Transitions:

One or more buttons that execute any transitions that are valid for the issue.

Note: If an issue doesn't match any workflow stages, or is subject to multiple workflows, AccuRev displays a message to that effect where the buttons would otherwise appear. If an issue's fields don't match with the logic specified for a transition, the transition button appears, but it is grayed out until the user changes the fields required to bring the issue into compliance.

Working in an Issue Form

You can enter and change values in any number of fields, and visit different pages in a multiple-page form. Here are some notes, describing aspects of AccuWork Issue forms that may differ from other fill-in-the-blanks forms:

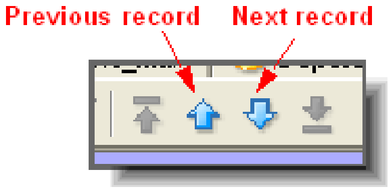
1. *Navigating fields and pages*

   To move from field to field, use the mouse or press **Tab** and **Shift-Tab**. To move from page to page in a multiple-page Issue form, click the subtab name (there is no keyboard accelerator).

   The **Enter** key confirms a menu choice or enters a line-break in a multiple-line text field. It does not perform field navigation. Within a multiple-line text field, the **Tab** key inserts a TAB character, rather than jumping to the next field.

2. *Browsing multiple issue records*

While you're working with issue record, browse arrows are enabled in the Issue form's toolbar.



This makes it easy to view a set of consecutive issue records. If you've made some changes that you haven't yet saved, AccuWork prompts you to save or discard those changes before switching to the previous or next one.

3. *Field initializations*

   When an Issue form appears after you invoke ***Create Issue***, you may notice that some fields already have values. AccuWork can include any number of such field initializations.

4. *Required fields*

   If a field's label appears in **red**, it is a required field. You cannot **Save** the issue record until every required field has a *value*. (A text field has a value if it's non-empty. A multiple-choice listbox field has a value if it's not "<none selected>".) Some of those fields may have gotten their values through an automated field initialization.

   If you attempt to save the issue record when one or more required fields still need a value, a pop-up window prompts you to supply the value(s).

5. *Other field validations*

   Field initializations and required fields are implemented through AccuWork's field validations (or "edit checks") facility. As you work in an Issue form, changing field values, you may notice these field-validation effects:

   - A text field, multiple-choice listbox field, or timestamp field has automatically been set to a specified value.
   - The set of required fields has changed.
   - A listbox's set of choices has changed.
   - The entire issue record, a particular page (subtab), or a particular field has been set to read-only status.

6. *An issue record's unique identifier*

   Each issue record has an integer identifier, its issue number. This number is unique within the depot (but not within the entire repository). AccuWork assigns the issue number automatically the first time the issue record is saved, and it cannot be changed thereafter. The issue number is stored in field **issueNum**. By default, the label for this field in an Issue form is **Issue**, but the edit-form designer can change the label for this field.

Saving or Discarding Your Work

Your entries in an Issue form are not stored in the issue record until you click the **Save** or **Save & Close** button in the Issue form toolbar. If you don't wish to save your work, just close the edit-form tab as you would any GUI tab.

When you save an issue record:

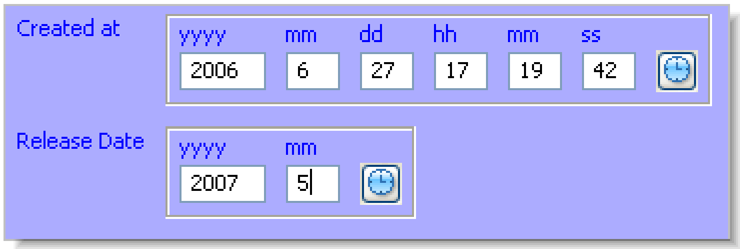
- (for a new issue record) AccuWork assigns the next available integer issue number to the new issue record. This number cannot be modified.
- The issue record is stored in the depot. A transaction of kind **dispatch** records the creation or modification of the issue record. No matter how the issue record is modified subsequently, you can revisit the issue record in its current form on the **Issue History** subtab.
- If you clicked **Save**, the Issue form remains open, with the newly assigned issue number replacing the "New Issue" label and appearing in the Issue field.

  Note: "Issue" is the default label for the issueNum field, but it might have a different label, depending on how the schema is configured.
- If you clicked **Save & Close**, the Issue form closes.

Working with Special Fields

AccuWork supports special field types, which have special user-interface "widgets" on an Issue form. These are described in the following sections.

Timestamp field



Created at

yyyy	mm	dd	hh	mm	ss
2006	6	27	17	19	42

Release Date

yyyy	mm
2007	5

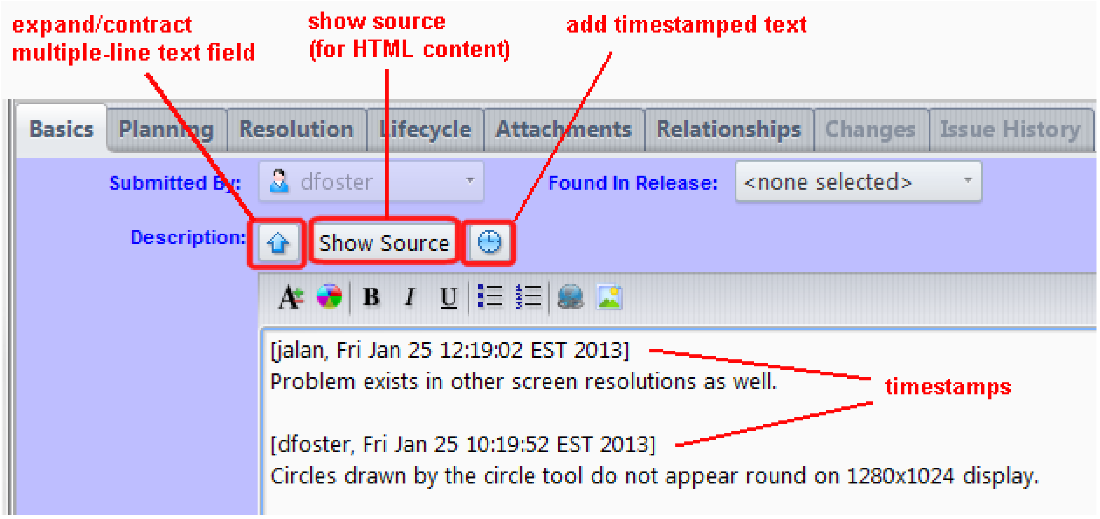
granularity: one second

granularity: one month

You can fill in the individual subfields manually, or use the 🕒 **Select Date** button to display choices ("tomorrow", "2 days ago", etc.) that fill in the subfields automatically. Once these subfields are filled in, you can revise them individually or by clicking the **Select Date** button again.

Multiple-Line Text and Log fields

A text field can have multiple lines, and can also specify both the width and the height of the textbox on the Issue form.



Submitted By: dfoster

Found In Release: <none selected>

Description:

[jalan, Fri Jan 25 12:19:02 EST 2013]
Problem exists in other screen resolutions as well.

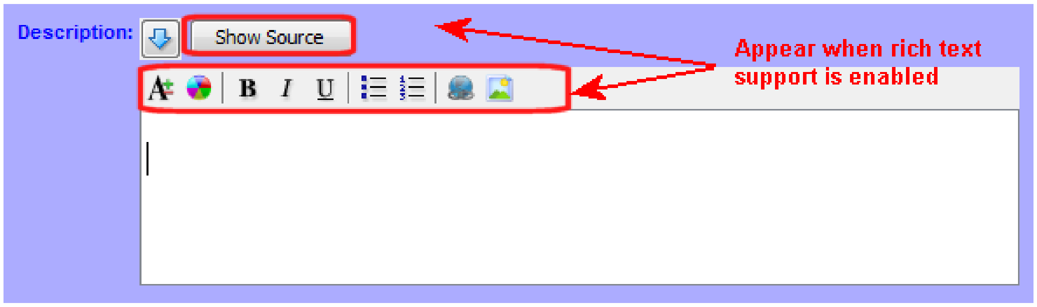
[dfoster, Fri Jan 25 10:19:52 EST 2013]
Circles drawn by the circle tool do not appear round on 1280x1024 display.

When you use the form you can use the ⇩ control at the top of the textbox to temporarily expand the height. Click the control again (it is now ⇧) to decrease the height. A scroll bar appears on the textbox whenever the current contents of the field are not completely visible.

A log field is a variant of a multiple-line text field. You can type directly into such a field. And if you click the 📝 **Add Timestamped Text** control, a timestamp text line is inserted at the beginning or end of the field.

Rich text fields

If support for rich text has been enabled by your AccuRev administrator, AccuWork displays tools to help you work with HTML, as shown in the following example of a Description field:



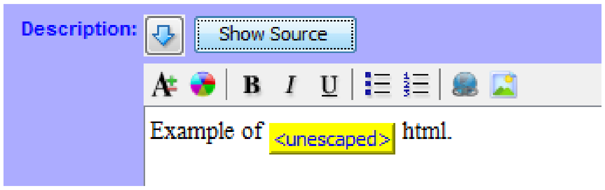
The following table summarizes rich text tools and what they do.

Tool	Description
Show Source button	Displays the HTML source for formatted text. When clicked, the button text changes to **Hide Source**. Clicking the **Hide Source** button reverts the display to formatted text only.

Tool	Description
Toolbar	Contains tools for text formatting and working with elements like table and images ▪ Font size ▪ Font color ▪ Bold text ▪ Italic text ▪ Underlined text ▪ Unordered list ▪ Ordered list ▪ Hyperlink ▪ Image

Fields for which rich text support has been enabled display formatted text (numbered lists and bolded text, for example) by default. For example, if you create an issue in the AccuRev Web UI or a third-party issue tracking tool that supports rich text, the formatted text is displayed in AccuWork.

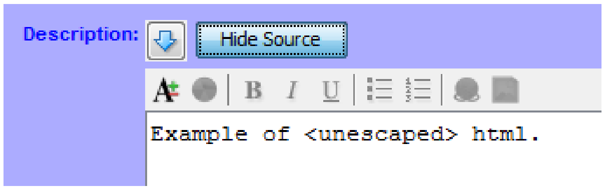
Note: If a field contains unescaped bracketed text in, say, a third-party issue tracking system, AccuWork renders the brackets and the string they contain as a button, as is the case with the string <*unescaped*> in the example shown here:



Example of <unescaped> html.

To resolve issues like this:

1. Click the **Show Source** button to display the HTML source.

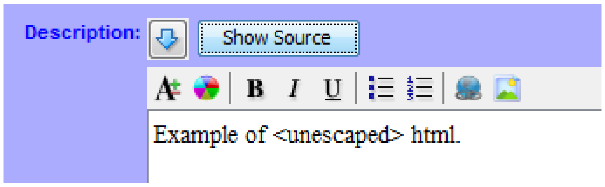


```
Example of <unescaped> html.
```

2. Copy the HTML source.

3. Click the **Hide Source** button to revert to the rich text display.

4. Replace the rich text with the HTML source you copied in Step 2.



Example of <unescaped> html.

Rich text support is available for fields with a data type of either *Text* or *Log* and is enabled on a per-field basis. See *Enabling Rich Text for Text and Log Fields* on page 384 for more information.

Timespan field

A timespan field is AccuWork's only numeric field type. You can type any value into the field, but when you leave the field, AccuWork automatically erases a non-numeric value. Negative numbers and decimals are allowed. If you enter an integer (e.g. **45**), it is automatically converted to a decimal (**45.0**).

Attachments field

An Issue form can contain one or more attachment fields. In each such field, you can specify one or more files and/or Internet addresses (URLs) to be attached to the current issue record. AccuWork displays the attachments data as a table.

In addition to specifying the location of a file or Internet resource, you enter a name and optional comment. AccuWork automatically fills in your username, the date, and the size of the attached file. (Internet URLs are assigned a size of 0.) If the edit-form field is not large enough to show all these attachment parameters, use the scroll bar to see all the data. You can also resize and rearrange the columns of an attachments table.

An attachment field includes its own toolbar, with these buttons:

Field	Description
New Attachment	Define a new file attachment for this issue record. A New File Attachment dialog appears, in which you can specify one or more files, a name for the attachment, and a comment string. You can specify multiple files at once -- each one becomes a separate file attachment. In this case, the Attachment Name input field is disabled; each filename is automatically assigned as the attachment name. The comment string that you specify is assigned to each file attachment.
New URL	Define a new attachment to be the address (URL) of an Internet resource. A New URL Attachment dialog appears, in which you specify a URL, a name for the attachment, and a comment string. AccuWork inserts the string *http://* in the URL field. You can erase this if you want to specify a location accessed by another Internet protocol, such as *ftp://* .
Open Attachment	Open the existing attachment that is currently selected, using the appropriate program.
Save Attachment As	Create a copy of the currently selected attachment on the client machine.
Properties	Launch a Properties window, displaying the definition of the currently selected attachment. You can use this window to change the attachment's Name or Comment value.
Delete Attachment	Remove the attachment from the issue record (and from the depot). A URL attachment can be deleted at any time. A newly added file attachment can be deleted if you haven't yet saved the issue record.

When you save the issue record, each file is copied to the depot, so that the data always remains available through the issue record, even if an original file is deleted.

Relationship field

AccuWork supports several kinds of relationships between a pair of issue records:

- **Duplicate**: You can specify that issue record B duplicates issue record A, so that no work need be done on B. (Perhaps the same bug was reported twice.) Any number of records (B, C, D, ...) can duplicate a given record (A). The default AccuWork issue schema requires that a Duplicate relationship be established when an issue record's State field is set to the value **Duplicate**. See Viewing and

Maintaining an Issue Record's "Duplicate" Relationships on page 302 for more information.

- **Dependency**: You can specify that issue record B depends on issue record A (for example, from a project management viewpoint). AccuRev imposes no further semantics on such dependencies; this is a good application for user-defined scripts or practices.

  IMPORTANT NOTE: Issue dependencies you define manually on the Issue form are independent of change package dependency relationships among issue records, which are maintained automatically by AccuRev. See *Change Packages, Segments, and Dependencies* on page 334 for more information.

- **Subtask**: You can specify that issue record B defines a subtask of the work to be performed for issue record A. In this case, AccuRev considers issue record A to be "in" a particular stream only if issue record B is "in" that stream, also.

- **Tracking**: Tracking issues are sometimes required after you cross-promote and patch multiple issues to a destination stream, and you might choose to use them if for some reason you wish to have a patch operation (for example) tracked by its own unique issue rather than by a variant of an existing issue. See *Tracking Issues* on page 375 for more information.

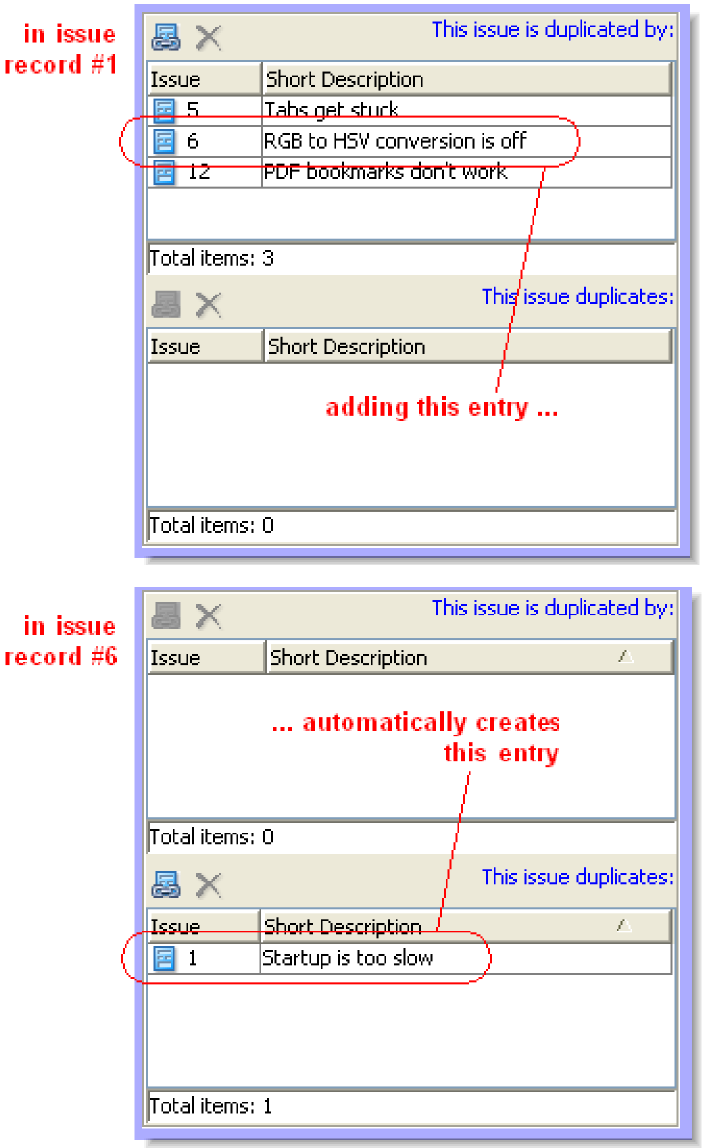
You cannot create "relationship chains": for example, if B duplicates A, you cannot make C a duplicate of B.

Issue record relationships can be viewed only in the AccuWork GUI. The Issue form tables that display relationships are not included when you export an issue record (for viewing or printing). These tables are not included when you use the AccuWork command line interface (CLI) to dump the contents of an issue record. (See the *AccuWork™ Command-Line Interface* chapter in the AccuRev *CLI User's Guide*.)

When you create or delete a relationship between issue records, the change to both issue records is saved immediately. There is no need to invoke **Save** on the issue record. (The 💾 **Save** button is not enabled in the edit-form toolbar.)

Viewing and Maintaining an Issue Record's "Duplicate" Relationships

The "duplicate" relationship is not symmetrical: "A duplicates B" is not the same as "B duplicates A".



This issue is duplicated by:

Issue	Short Description
5	Tabs get stuck
6	RGB to HSV conversion is off
12	PDF bookmarks don't work

Total items: 3

This issue duplicates:

Issue	Short Description

Total items: 0

This issue is duplicated by:

Issue	Short Description

Total items: 0

This issue duplicates:

Issue	Short Description
1	Startup is too slow

Total items: 1

Accordingly, an edit-form field displaying a duplicate-type relationship consists of two tables: the top table shows issue records that are duplicated by the current record; the bottom table shows issue records that the current record duplicates.

Because of the "relationship chains" restriction, issue records can appear in only one of these tables, not both.

On the other hand, "A duplicates B" *is* the same as "B is duplicated by A". This means that when you add an entry to the *top* table of one issue record, AccuWork automatically adds it to the *bottom* table of the other issue record (and vice-versa).

A Duplicate relationship field has its own toolbar, with these buttons:

Link Issue

Create a relationship link with another issue record. AccuWork prompts you to enter an issue record number.

✗ Remove Link

Remove the selected relationship link.

Note: Duplicate Relationship and Duplicate State

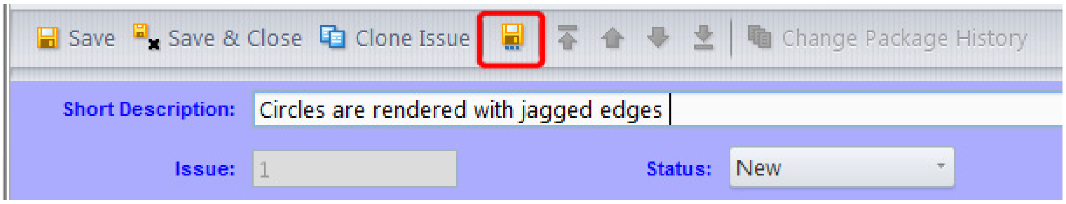
In AccuWork's default issue schema:

- There is a State field in the header section, one of whose values is **Duplicate**. (The Status value must be **Closed** to enable the setting of the State value to **Duplicate**.)
- There is a Duplicates field in the Relationships subtab.
- An edit-form validation specifies that when you set the State field's value to **Duplicate**, you must indicate which issue record is duplicated, by creating an entry in the Duplicates field.

Exporting an Issue Record

You can export issue records in either HTML or XML.

1. Click the **Export** button in the form's toolbar.

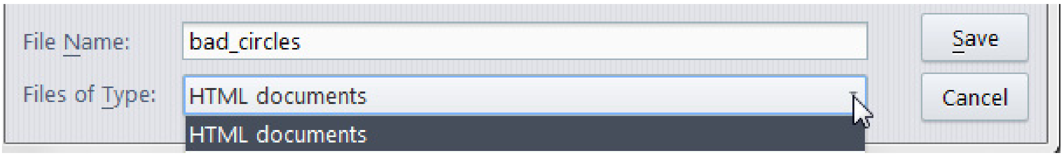


Short Description: Circles are rendered with jagged edges

Issue: 1

Status: New

2. Select the file type in the Save As dialog that appears.



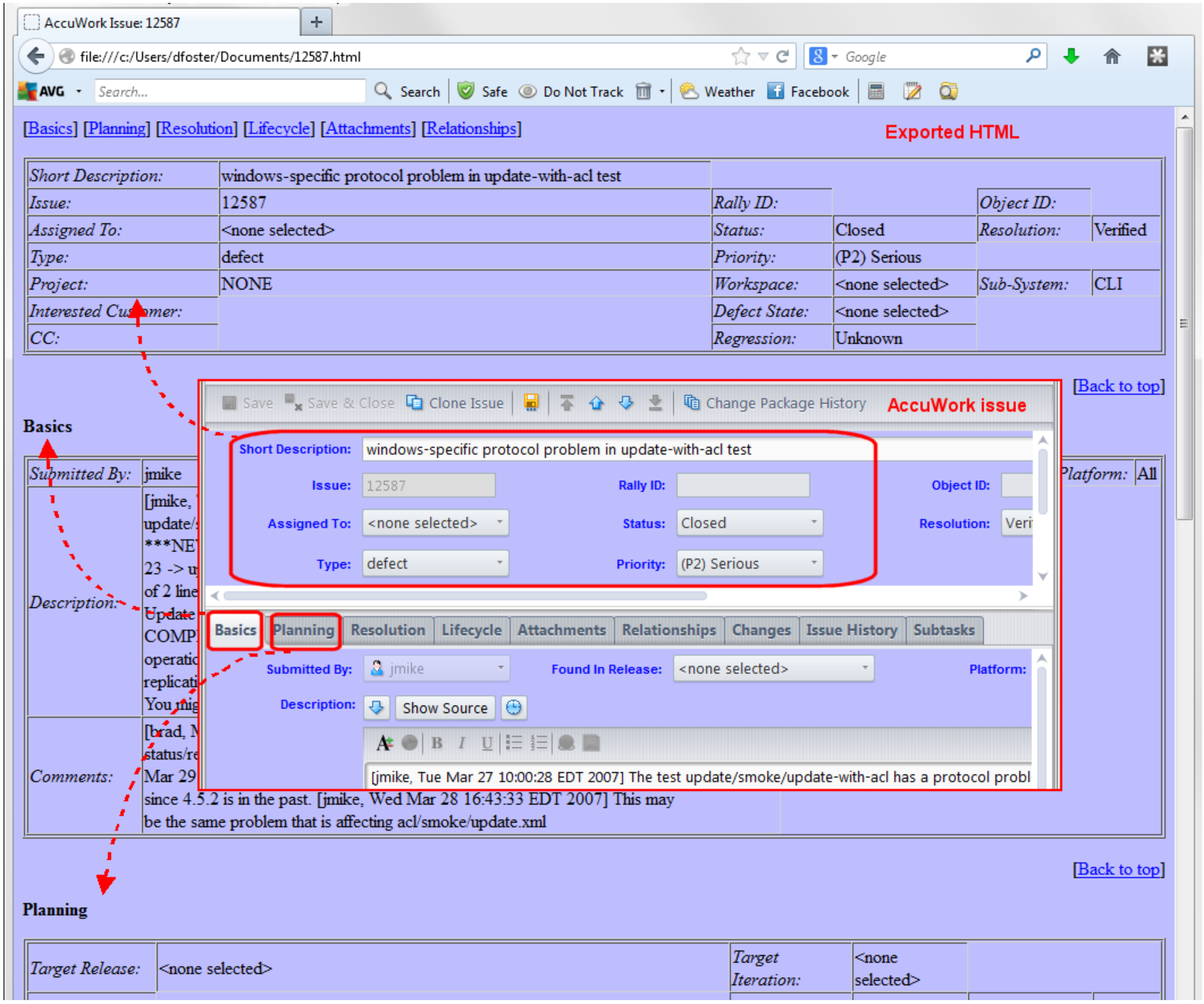
File Name: bad_circles

Files of Type: HTML documents

HTML documents

3. Specify a pathname for the export file. You don't need to specify the *.html* or *.xml* suffix -- AccuWork adds it automatically.

When creating an HTML file, AccuWork outputs all the content of the issue record and approximates the form layout, too. Even if the Issue form has multiple pages, you need to "print" only once. All of the pages are combined into a single HTML or XML file:]



[Basics] [Planning] [Resolution] [Lifecycle] [Attachments] [Relationships]

Short Description:	windows-specific protocol problem in update-with-acl test				
Issue:	12587	*Rally ID:*		*Object ID:*	
Assigned To:	<none selected>	*Status:*	Closed	*Resolution:*	Verified
Type:	defect	*Priority:*	(P2) Serious		
Project:	NONE	*Workspace:*	<none selected>	*Sub-System:*	CLI
Interested Customer:		*Defect State:*	<none selected>		
CC:		*Regression:*	Unknown		

[Back to top]

Basics

Submitted By:	jmike	*Platform:* All
Description:	[jmike, update/ ***NE 23 -> u of 2 line Update COMP operatic replicati You mig	
Comments:	[brad, N status/re Mar 29 since 4.5.2 is in the past. [jmike, Wed Mar 28 16:43:33 EDT 2007] This may be the same problem that is affecting acl/smoke/update.xml	

Short Description: windows-specific protocol problem in update-with-acl test

Issue: 12587 Rally ID: Object ID:

Assigned To: <none selected> Status: Closed Resolution: Veri

Type: defect Priority: (P2) Serious

Submitted By: jmike Found In Release: <none selected> Platform:

Description:

[jmike, Tue Mar 27 10:00:28 EDT 2007] The test update/smoke/update-with-acl has a protocol probl

[Back to top]

Planning

Target Release:	<none selected>	*Target Iteration:*	<none selected>

On Windows machines, AccuWork automatically invokes a web browser on the HTML file it creates. If you wish, resize the browser window to optimize the look of the issue record. HTML documents automatically adjust to changes in window width. Then, use the browser's print option to create a printout of the issue record.

The Issue Form -- "Changes" Subtab

The Changes subtab of an issue record's Issue form shows the change package of the issue record. You can invoke options to see the details of the change package's changes, maintain its entries, and view the transaction and version history of its elements.

For a detailed discussion of change packages, see *Chapter 10 Change Packages*.

"Changes" Subtab Layout

The Changes subtab contains a table, each row displaying one "change package entry". A given element can have at most one entry in a change package -- but this entry can "grow" over time, as development work continues on the element. The table includes these columns:

Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

Version

The head version of the change package entry.

Basis Version

The basis version of the change package entry.

The head version and basis version are the "endpoints" of the change package entry. That is, the entry consists of all the versions between these two versions. (The head version is included; the basis version isn't.)

Working in the "Changes" Subtab

The Changes subtab has options for modifying change package contents:

- **Remove** deletes one or more selected entries from the change package.
- **Send to Issue** copies one or more selected entries to another issue record's change package.

But the change package facility is designed so that most modifications to change packages occur as "side effects" to the **Promote** operation, by the change-package-level integration between AccuRev and AccuWork. (*Change-Package-Level Integration between AccuRev and AccuWork* on page 400.)

See also: *Working with Tables* on page 14

Options Available in the "Changes" Subtab

Remove

Deletes the selected entry(s) from the change package.

This change takes effect immediately -- you cannot discard the change by closing the Issue form without invoking **Save**. (Removing change package entries does not enable the **Save** button in the edit-form toolbar.)

Diff Against Basis

(text-file elements only) Compares the entry's head version with the basis version. This shows exactly what changes to this element are included in the change package.

Show History

Open a History Browser tab, containing the transactions involving the selected element.

Browse Versions

Open a Version Browser tab, showing all the versions of the selected element, and their interrelationships (ancestry).

Send to Issue

Copies the selected entry(s) to another issue record's change package. If there's an existing entry for an element in a destination change package, an attempt is made to combine the existing entry with the entry you specified. The change(s) take place immediately in the destination issue record(s). There is no way to cancel the change(s) -- but you can open the issue record(s) and **Remove** the entry(s).

The default query is executed, and the results are displayed in a dialog (see *Send to Issue : Associating Changes with an Issue* on page 354). You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

Properties

Displays information about the selected element. The data items displayed vary with the type of element.

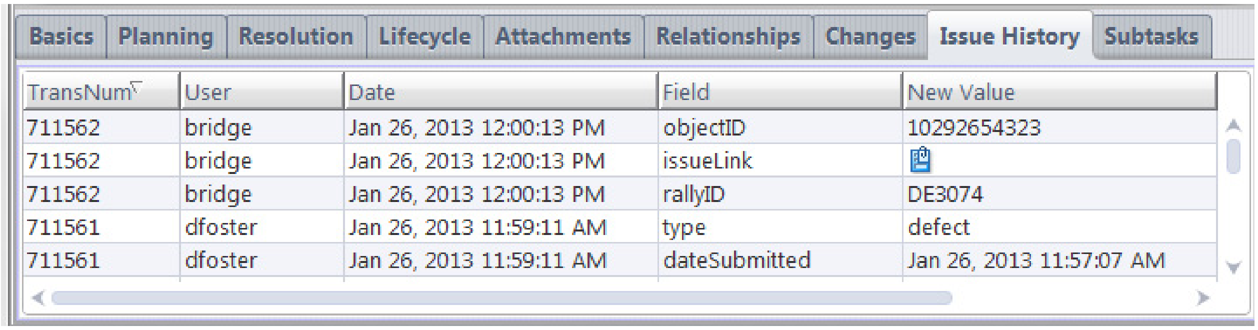
The Issue Form -- "Issue History" Subtab

The Issue form of an AccuWork issue record always includes a subtab named Issue History, which is maintained automatically by AccuWork. This subtab shows, on a field-by-field basis, how the issue record has changed over time.

You can use this subtab to view past "versions" of the issue record.

"Issue History" Subtab Layout

The Issue History subtab contains a table, each row of which shows how one field was changed (or initially created) by a particular save of the issue record.



TransNum	User	Date	Field	New Value
711562	bridge	Jan 26, 2013 12:00:13 PM	objectID	10292654323
711562	bridge	Jan 26, 2013 12:00:13 PM	issueLink	📋
711562	bridge	Jan 26, 2013 12:00:13 PM	rallyID	DE3074
711561	dfoster	Jan 26, 2013 11:59:11 AM	type	defect
711561	dfoster	Jan 26, 2013 11:59:11 AM	dateSubmitted	Jan 26, 2013 11:57:07 AM

The table contains these columns:

TransNum

The number of the ***Save*** transaction. (The transaction kind is **dispatch**.)

User

The user who made the change to the issue record.

Date

The time at which the change tool place.

Field

The field whose value changed (or was initially created).

New Value

The value to which this transaction changed this field.

Old Value

The value of this field prior to the ***Save***. (Empty for the issue record's initial ***Save*** transaction.)

Working in an "Issue History" Subtab

If you double-click any row of the Issue History table, AccuWork opens a read-only Issue form, showing the state of the issue record at the time of that particular **Save** operation. (Exception: the issue record's change package does not participate in this feature.)

The New File Attachment Dialog

The Browse for Attachment dialog allows you to browse for files to attach to an issue record when in a field of type Attachments.

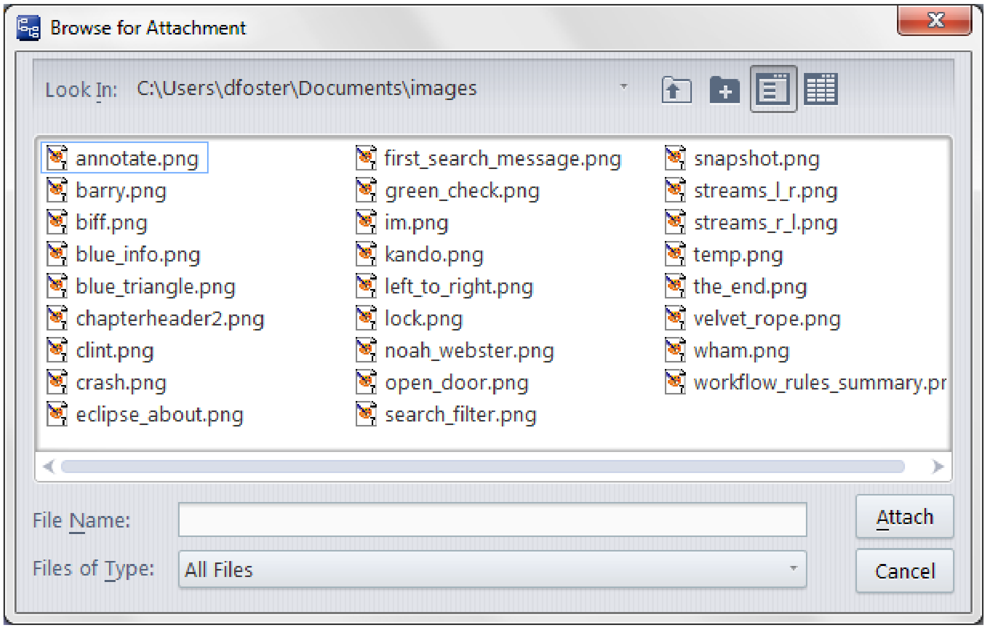
Invoking the New File Attachment Option

Click the **New File Attachment** icon in the toolbar of the Attachments field in an issue record's Issue form.

Using the Browse for Attachment Dialog

The dialog is a standard file-open dialog for your operating system.

Browse to a directory, choose one or more files, and click **Attach**.



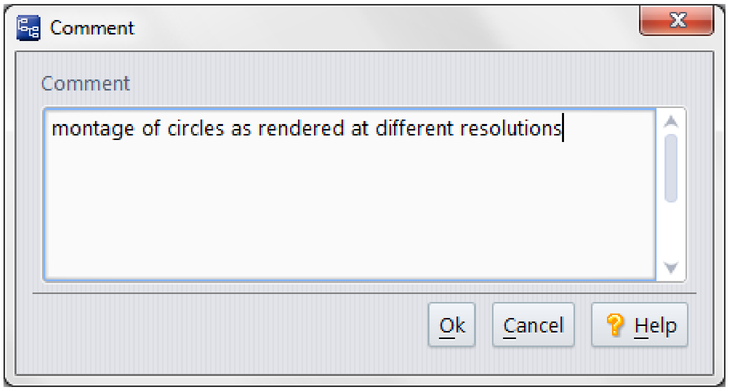
Look In: C:\Users\dfoster\Documents\images

annotate.png
barry.png
biff.png
blue_info.png
blue_triangle.png
chapterheader2.png
clint.png
crash.png
eclipse_about.png
first_search_message.png
green_check.png
im.png
kando.png
left_to_right.png
lock.png
noah_webster.png
open_door.png
search_filter.png
snapshot.png
streams_l_r.png
streams_r_l.png
temp.png
the_end.png
velvet_rope.png
wham.png
workflow_rules_summary.pr

File Name:

Files of Type: All Files

Then enter comments (if desired), in the Comments dialog that displays. Click **OK** to attach the file(s) to the issue record.



montage of circles as rendered at different resolutions

The New URL Attachment Dialog

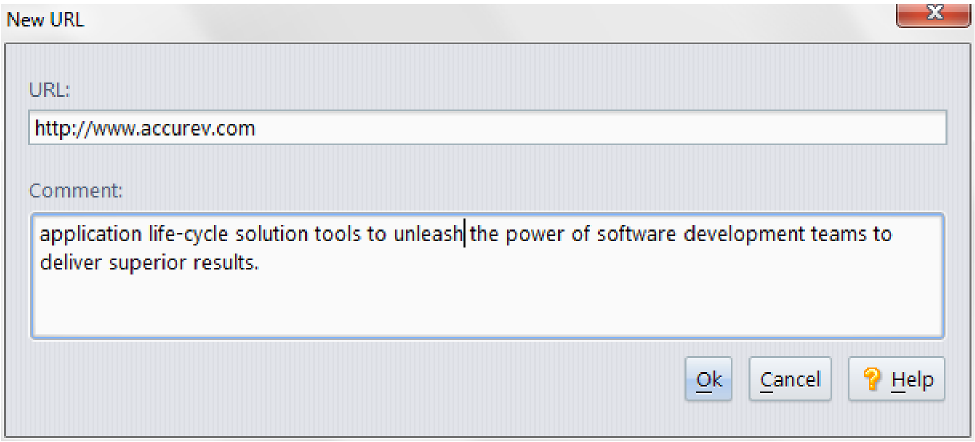
The New URL Attachment dialog defines one or more URL attachments in an issue record field of type Attachments.

Invoking the New URL Attachment Option

Click the **New URL Attachment** icon in the toolbar of the Attachments field in an issue record's Issue form.

Using the New URL Dialog

Fill in these fields to define one or more attachments:



URL: http://www.accurev.com

Comment: application life-cycle solution tools to unleash the power of software development teams to deliver superior results.

URL

A full Web address.

Comment

Any text string.

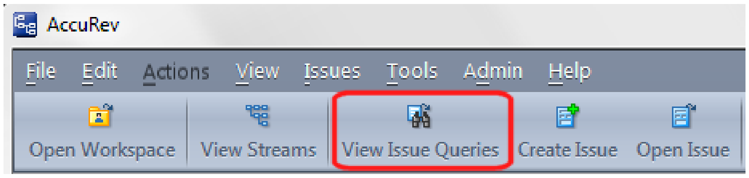
Creating AccuWork Queries

An AccuWork query retrieves a set of records according to user-defined selection criteria. AccuWork has a point-and-click interface for creating and editing queries; it enables you to create simple queries quickly, and to create sophisticated queries in a straightforward, reliable way.

By default, the queries that you create are private queries, which cannot be seen by other users. You can declare any query to be a public query. Such queries are visible to all users, who can use and copy them, but cannot modify your original.

Opening a Queries Tab

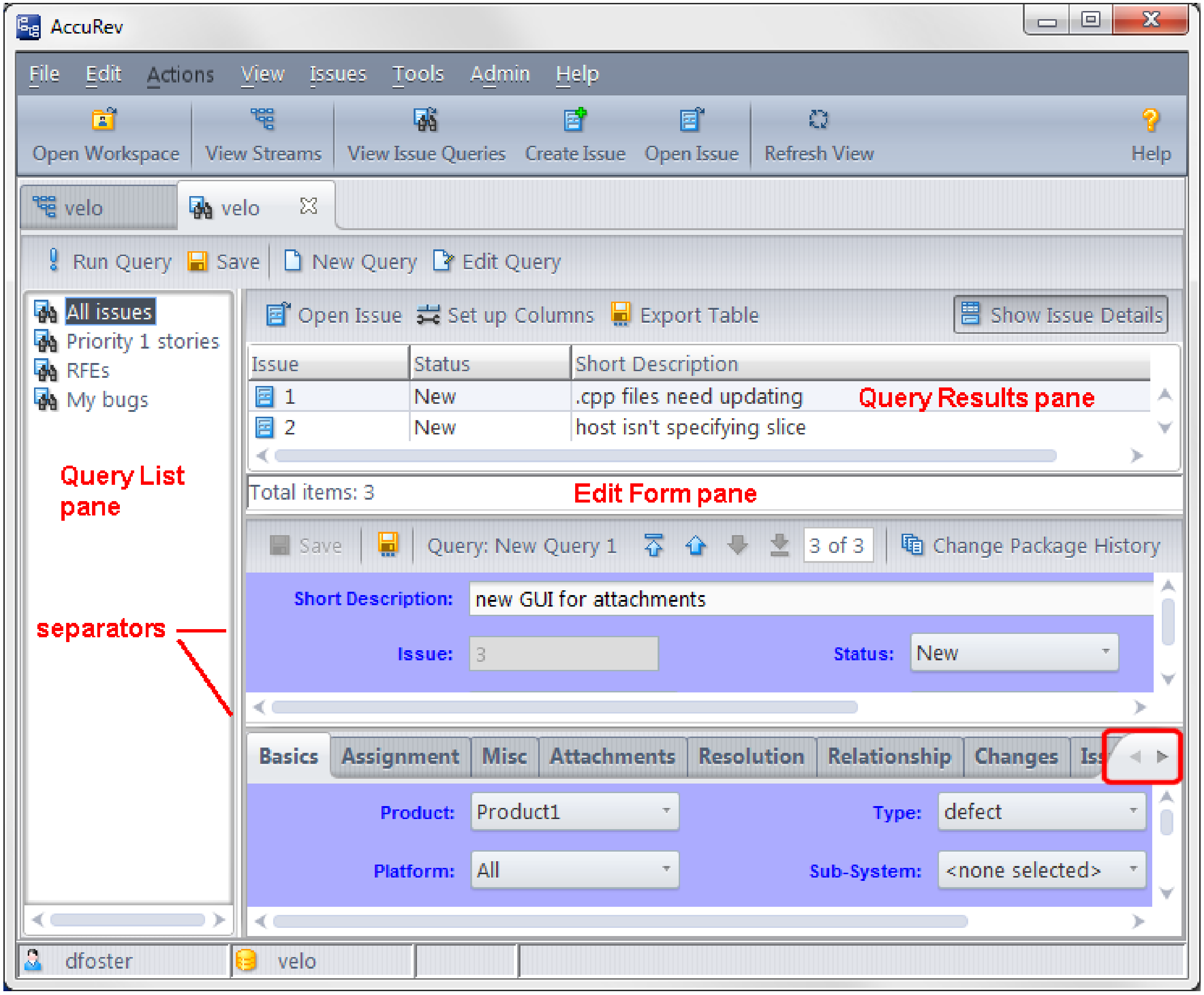
Choose the **Issues > View Issue Queries** option from the GUI main menu, or click the **View Issue Queries** button in the GUI main toolbar.



AccuRev prompts you to choose a depot, and then displays the **Queries** tab.

Query Tab Layout

The Query tab contains three panes, each with its own toolbar:



Issue	Status	Short Description
1	New	.cpp files need updating
2	New	host isn't specifying slice

Total items: 3

Short Description: new GUI for attachments

Issue: 3

Status: New

Product: Product1

Type: defect

Platform: All

Sub-System: <none selected>

Query List pane

Query Results pane

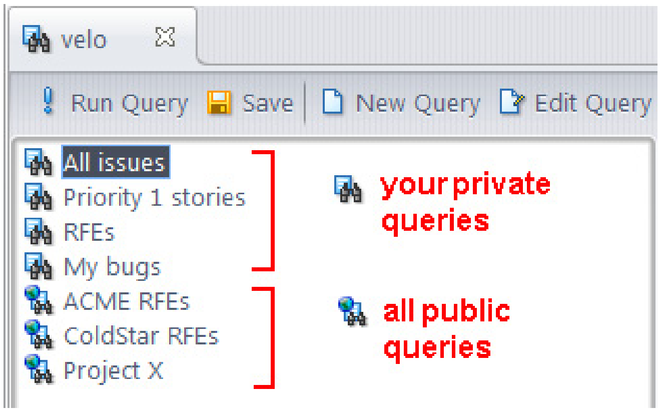
Edit Form pane

separators

- The Query List pane lists the names of all your existing private queries, along with public queries available to all users. Using the toolbar in this pane, you compose new queries, view/revise/rearrange/execute existing queries, and delete queries.
- When you execute a query, the set of issue records selected by the query are displayed as a results table in the Query Results pane. If you've set a default query for this depot, it's executed automatically when you open the Queries tab.
- If you click the 🗎 **Show Issue Details** button in the toolbar of the Query Results pane, an Issue form pane appears below it. This pane is a fully functional Issue form, which you can use to view and modify the issue record currently selected in the Query Results pane. A user preference (**Tools > Preferences**) configures this pane to appear automatically when you open a Queries tab.

Query List Pane Layout

The Query List pane contains a list of the private queries you've defined for the current depot, along with any public queries defined by you and/or other users.



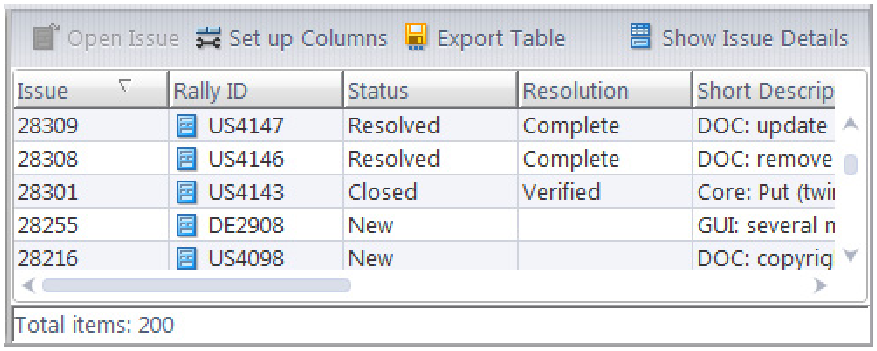
The listing is not alphabetical -- whenever you create a new query, it's simply added to the end of the list.

Your queries (both private and public) are always listed above other users' public queries.

See *Working in the Query List Pane* on page 312.

Query Results Pane Layout

The Query Results pane contains a table that displays the results of a query.



Issue	Rally ID	Status	Resolution	Short Descrip
28309	US4147	Resolved	Complete	DOC: update
28308	US4146	Resolved	Complete	DOC: remove
28301	US4143	Closed	Verified	Core: Put (twi
28255	DE2908	New		GUI: several n
28216	US4098	New		DOC: copyrig

Total items: 200

Each row of the table displays one issue record; each column displays a particular issue-record field.

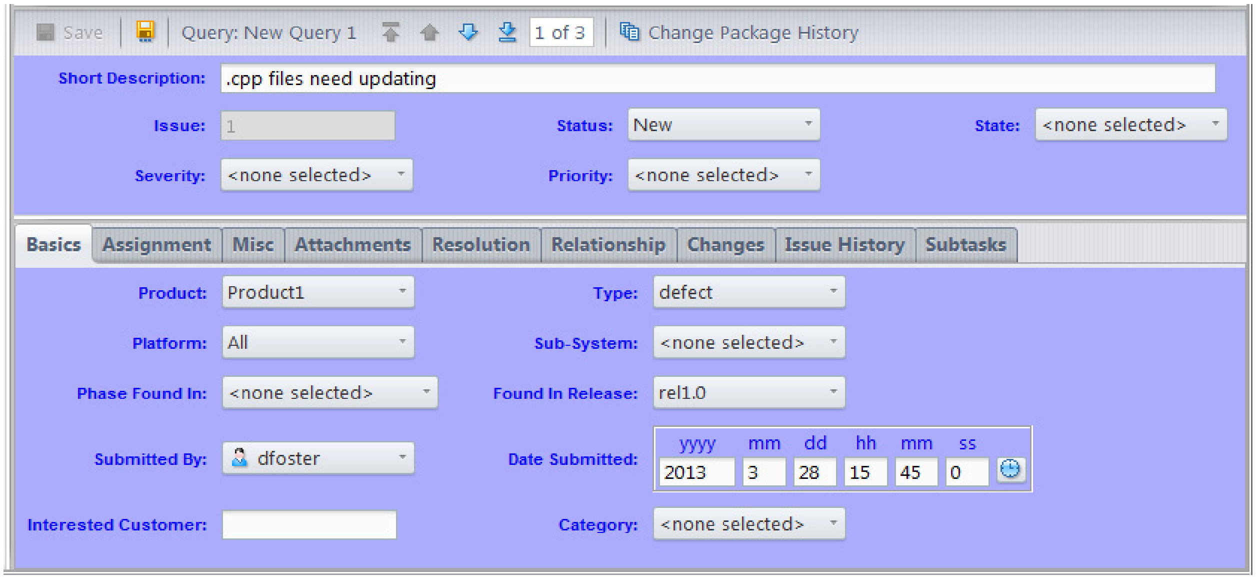
Each query has its own results-table design: a set of columns (fields), in a particular order and with particular column-widths. The results table's design can also include a single-level or multiple-level sort order for the rows (issue records).

When you create a new query, AccuWork automatically starts off the results table with a couple of columns, including Issue (**issueNum** field). You can add more columns at any time. You can also remove the Issue column from the table. **Note:** "Issue" is the default label for the issueNum field, but the edit-form designer can change the label.

See *Working in the Query Results Pane* on page 314.

IssueForm Pane Layout

The Issue form pane is just like the "full size" Issue form, except that it occupies a pane within the Queries tab, instead of having an entire tab to itself. This pane displays the entire contents of the issue record that's currently selected in the Query Results pane. In addition, browse arrows are enabled in the Issue form pane's toolbar, making it easy to view some or all of the records selected by a query. There is also a **Change Package History** button ( ) that displays all the modifications that have been made to the change package for this issue.



Short Description: .cpp files need updating

Issue: 1

Status: New

State: <none selected>

Severity: <none selected>

Priority: <none selected>

Product: Product1

Type: defect

Platform: All

Sub-System: <none selected>

Phase Found In: <none selected>

Found In Release: rel1.0

Submitted By: dfoster

Date Submitted: yyyy 2013 mm 3 dd 28 hh 15 mm 45 ss 0

Interested Customer:

Category: <none selected>

See *Working in the Issue Form Pane* on page 315.

Working with Issues

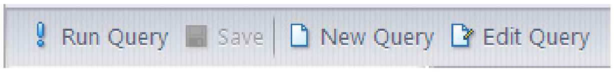
Here's a typical procedure for working with issues in AccuWork:

1. In the Query List pane, create a new query (or select an existing one), then execute the query.
2. In the Query Results pane, view selected fields from the issue records selected by the query.
3. Open the Issue form pane to view the entire contents of one of the selected issue records (and perhaps modify it).

The following sections describe working in each of the Queries tab's panes.

Working in the Query List Pane

You can invoke the following options in the Query List pane, using context menus or the Query List toolbar.



Most options operate on the currently selected query.

Run query

(equivalent: double-click the query name) Executes (or re-executes) the selected query, displaying the results in the Query Results pane. You don't need to run this option when you revise a query in the **Edit Query** window -- AccuWork automatically executes the revised query and updates the results table.

Save

Save all your private queries. (There is no way to save a single query.) Your private queries are stored in the AccuRev repository within the depot directory, in an XML-format file:

```
.../storage/<depot-name>/dispatch/config/user/<username>/query.xml
```

Notes:

Each user's private queries are stored separately

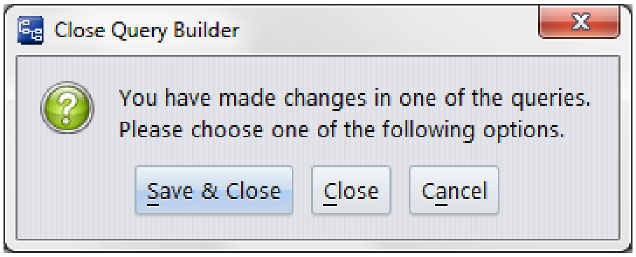
The *<username>* directory in the query storage pathname causes the private queries for each user to be stored separately.

Queries are not version-controlled

Although your queries are stored in the depot, they are not version-controlled in the way AccuRev files are. For example, there is no option that displays or reinstates your queries as you saved them two days ago.

Automatic save-on-close feature

If you try to close a Queries tab in which you've made changes to one or more queries, but haven't run **Save all queries**, AccuWork offers to perform the save.



Close Query Builder

You have made changes in one of the queries.
Please choose one of the following options.

New query

Edit query

Create a new query or revise an existing one, using the **Edit Query** window.

Clone query

(Context menu only) Create a private query that is a copy of the selected private or public query. The new query is initially named "Copy of ...", but you can change the name.

Move query up

Move query down

(Context menu only) Change the position of the selected query in the Query List pane. Note that your own queries (both private and public) are always listed above other users' public queries.

Delete query

(Context menu only) Remove the currently selected query. The deletion does not take effect until you run **Save all queries**. If you close the Queries tab without saving your queries, the "deleted" query will still exist the next time you open the Queries tab. This may or may not be the right thing to do: closing the Queries tab without saving also discards changes you've made to other queries since the most recent save.

Set as Default
Disable as Default

(Context menu only) Designates one of your private queries to be -- or to stop being -- your default query for this depot. **(**This is your default query -- another user can set one of his own private queries to be his default query.)

The query name is redisplayed in ***bold-italic*** to indicate that it is the default query.

Your default query is executed:

- Whenever you open a new Queries tab.
- When you execute **Send to Issue** (see *Send to Issue : Associating Changes with an Issue* on page 354).
- A query loses its status as your default query when you run **Disable as Default** on it, or when you select another query as the default.

Set as Private
Set as Public

Changes a query that you created from public to private, or vice-versa.

Additional Operations in the Query List Pane

- **Renaming a Query --** You can rename any of your private queries in the Query List pane, without having to open an Edit Query window:
  1. Select the query.
  2. Press function key **F2** or click the query name again. (Be careful not to double-click; that runs the query.)
- **Revisiting the Results of a Query --** Whenever you select a particular query in the Query List pane, the Query Results pane displays that query's most recent results. The cache of previous query results is cleared when you close the Queries tab.

Working in the Query Results Pane

In many cases, browsing the results table produced by running a query may be all you need to do. But in other cases, you may want to see a selected issue record in the context of its Issue form. Why?

- The results table is read-only. To modify an issue record, you must access it through its Issue form.
- The data you need may be in a field that is not included in the results table. Instead of adding a column to the results table, you can use the Issue form to display all the fields.

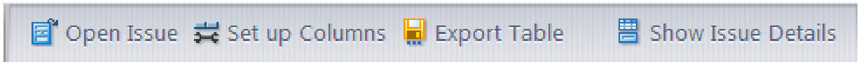
You can use a "full-size" or "reduced-size" Issue form to view a issue record:

- Invoke **Open Issue** to open a separate Issue form for the currently selected issue record.
- Invoke **Show Issue Details** to make an Issue form pane appear within the Queries tab.

With either tool, browse arrows in the Issue form's toolbar make it easy to view some or all of the records in the results table.

Options Available in the Query Results Pane

The following options are available in the Query Results pane:



Open Issue

Open a separate Issue form to view/revise the currently selected issue record.

Setup Columns

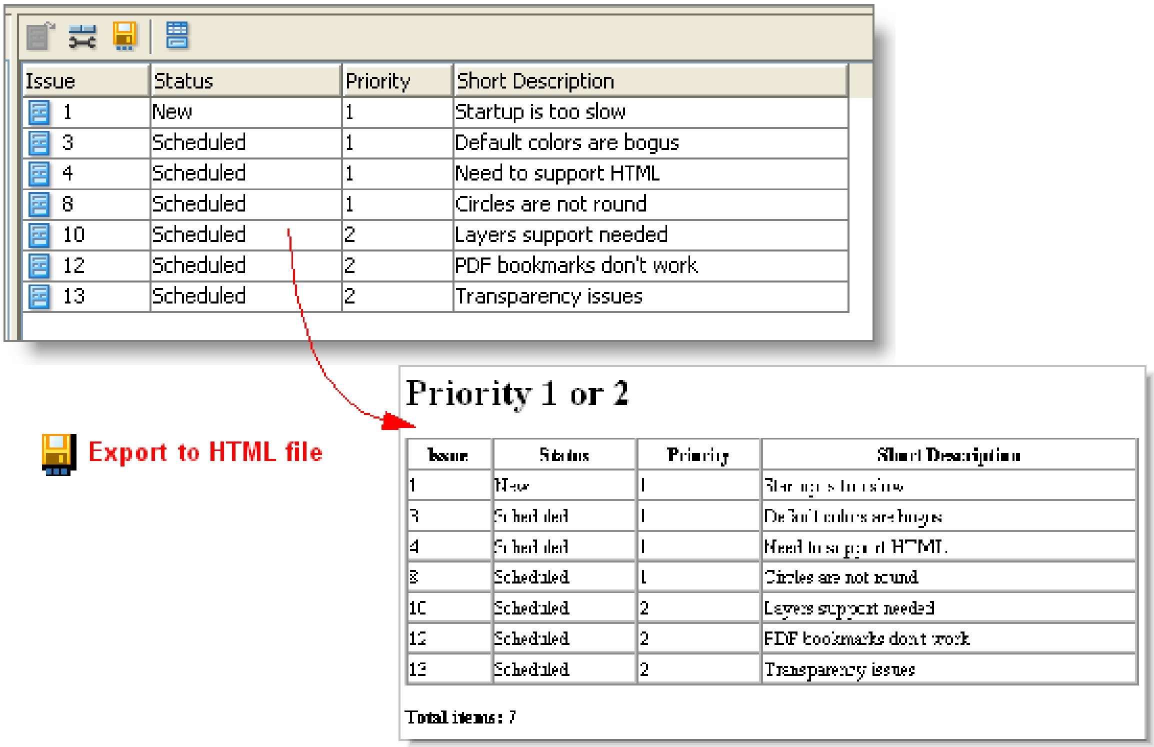
Open a Columns window, in which you can change which columns (fields) appear in the table, and also change their order.

Remove Column

(appears only in the context menu of an individual column) Removes the column you right-clicked from the results table. (This is a shortcut, eliminating the need to invoke **Setup Columns**.)

Export Table

Open a File Chooser window, in which you specify a file to store the entire contents of the results table using an HTML, XML, or CSV format:



Issue	Status	Priority	Short Description
1	New	1	Startup is too slow
3	Scheduled	1	Default colors are bogus
4	Scheduled	1	Need to support HTML
8	Scheduled	1	Circles are not round
10	Scheduled	2	Layers support needed
12	Scheduled	2	PDF bookmarks don't work
13	Scheduled	2	Transparency issues

Export to HTML file

Priority 1 or 2

Issue	Status	Priority	Short Description
1	New	1	Startup is too slow
3	Scheduled	1	Default colors are bogus
4	Scheduled	1	Need to support HTML
8	Scheduled	1	Circles are not round
10	Scheduled	2	Layers support needed
12	Scheduled	2	PDF bookmarks dont work
13	Scheduled	2	Transparency issues

Total items: 7

HTML file displayed in Web browser

Show Issue Details

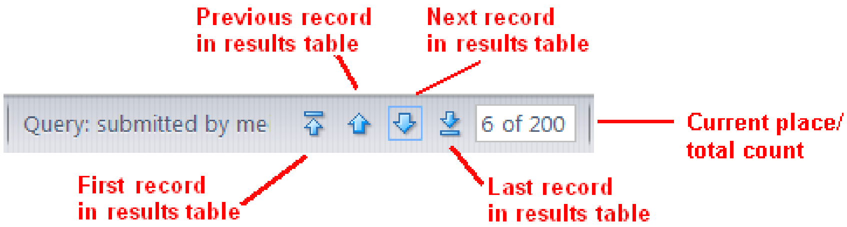
Toggles the Issue form pane, displaying the complete contents of the currently selected record in the Query Results pane.

See also: *Working with Tables* on page 14.

Working in the Issue Form Pane

Using a "reduced-size" Issue form within a pane of the Queries tab is virtually the same as using a "full-size" Issue form in its own tab. Here are some notes on the few differences:

- The browse buttons in the Query Issue form pane's toolbar makes it easy to visit all the records in the Query Results pane table.


Previous record in results table
Next record in results table
Query: submitted by me
6 of 200
Current place/ total count
First record in results table
Last record in results table

- After modifying a record and invoking **Save**, you may want to rerun the query in order to update the table in the Query Results pane. This doesn't occur automatically.

The AccuWork Query Editor

The AccuWork Query Editor is a tool for creating and revising AccuWork queries.

Opening the Query Editor

You can create or revise an AccuWork query in these contexts:

- In a Queries tab, click the **New Query** or **Edit Query** button. This opens a separate Edit Query window.

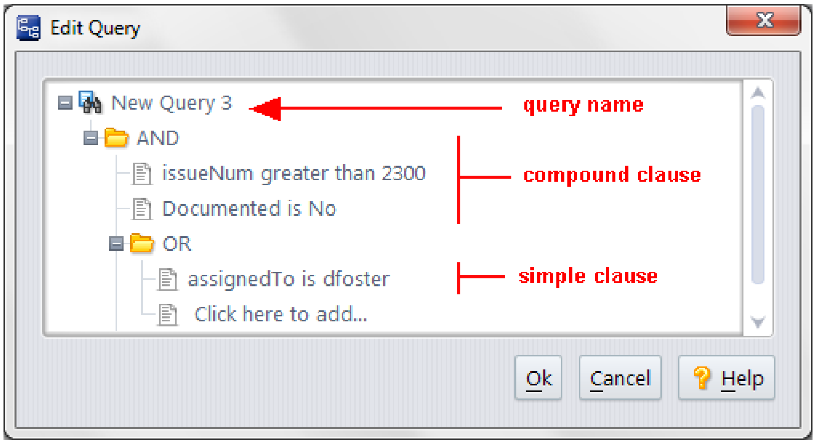
- In the Change Packages subtab of the Schema Editor, click the **New Trigger** or **Edit Trigger** button to open a separate Promote Trigger window. The Query Editor runs in the bottom pane of this window.

Query Editor Layout

The Query Editor displays the contents of one query. The query appears as a hierarchy, which you navigate using familiar expand/collapse controls. The hierarchical organization is a natural fit, because each query is, itself, a hierarchy of simple clauses and compound clauses:

- The first hierarchy level contains the query's name, if the Query Editor was invoked from the Queries Tab.
- A simple clause, such as **value of the *assignedTo* field is *dfoster***, can live at any level.
- A "compound AND" or "compound OR" clause consists of:
    - The operator AND or OR at one hierarchy level
    - Any number of clauses (simple or compound) at the next hierarchy level.

For example, this query can be expressed as:



"Retrieve each issue record that is (1) numbered at or above 2300, (2) is not documented, or (3) is assigned to user *dfoster*."

The context menu of a query's name in the Query Editor includes the option **In English**, which produces a sentence like the one above.

Working in the Query Editor

Video Tip	Click *this link* to view an introductory video on creating queries.

The following sections describe the details of working in the Query Editor.

Naming a New Query / Renaming an Existing Query

When you click the **New Query** button to create an empty query, AccuWork assigns it a placeholder name ("New Query *nnn*"). You can edit the name now: click the name once to select it, then press function key **F2** (or click it a second time) to begin editing it. Don't double-click — that's equivalent to using the expand/collapse control.

You can change a query's name at any time while Query Editor is still active. Or you can change it later, in the Query List pane.

Creating a Simple Clause

Every query consists at least one *simple clause* . A simple clause has three parts:

```
<field-name> <comparison-operator> <value>
```

The point-and-click interface makes creating a simple clause easy and (almost) foolproof. Start by clicking one of the "Click here to add ..." placeholders in the query:

First, you must select the <*field-name*> part of the clause from the list-box containing all the field-names. When you select a field-name, the query editor automatically adjusts the <*comparison-operator*> and <*value*> parts of the clause, based on the selected field. In the example below, the user has selected field-name *productType*, whose value must be one of these names: *ColorStar*, *Mixer*, *Publisher*.

The table below shows all the AccuWork data types, along with the corresponding choices for the <*comparison-operator*> and <*value*> parts of a simple clause. As you "fill in the blanks" to create simple clauses, you'll notice that AccuWork allocates new "Click here to add ..." placeholders, so that one is always available at each level of the query.

Field Type	Comparison Operator	Value
Text	contains matches does not contain does not match equal to not equal to less than less than or equal to greater than greater than or equal to	Any character string. (Do not enclose it in quotes.) The value is always interpreted as a string literal; there is no way to specify the value of some other field here. The comparison is always a case-insensitive string comparison, never a numeric comparison. For example, the value **3** is greater than the value **25**. The ***contains*** and ***does not contain*** operators perform a simple substring search (case-insensitive), with no pattern matching. The ***matches*** and ***does not match*** operators perform pattern matching (case-insensitive), using these wildcards: ***** matches 0 or more characters **?** matches any 1 character [aekz] matches **a**, **e**, **k**, or **z** [a-e] matches **a**, **b**, **c**, **d**, or **e** (**Note:** Don't mix uppercase and lowercase; neither [E–k] nor [e–K] matches any character] {one,two,seven} matches **one**, **two**, or **seven** \ or / Matches any directory-separator character, even if it's the "wrong one" for the client machine. For example, **src/do_*.java** matches **src\do_something.java** on a Windows client. **Examples:** **Rls 4*** matches **Rls 4** and **Rls 4.01**, but not **Rls 4.2** or **Rls 5** **Rls [2-5]*** matches **Rls 2**, **Rls3.04**, and **Rls 5.3**, but not **Rls 1.02** **the -? option** matches **the -X option** and **the -4 option,** but not **the -Op option** **cop{y,ies}** matches **copy** and **copies**, but not **copy(s)**
Timespan	equal to not equal to less than less than or equal to greater than greater than or equal to	A numeric value, representing an amount of time. Note: users specify the value in the Issue form as a number of *hours* (e.g. 7.5); an XML-format dump of the issue record created by *Export* reports the value as a number of *minutes* (e.g. 450).
Choose	is is not	One of the strings specified in the definition of this field in the Schema Editor.
List	is is not	One of the strings specified in the definition of a particular named list in the Schema Editor.
User	is is not is member of is not member of	One of the principal-names in the user registry maintained by the AccuRev server. Alternatively, a user-group defined in the registry.

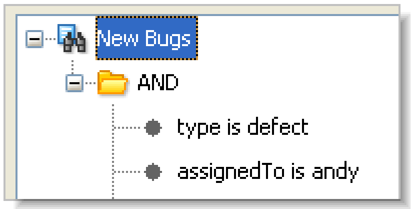
Timestamp	is is not is before is after is before or equal to is after or equal to	An AccuRev timestamp.
Attachments	contains	Any character string. This string is compared to the *Name* of each of an issue record's attachments. See .
internal	equal to not equal to less than less than or equal to greater than greater than or equal to	An integer, identifying a particular AccuWork issue record (*issueNum* field) or a particular AccuRev transaction (*transNum* field).

Creating a Compound Clause

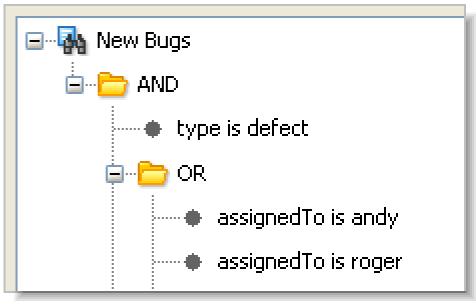
A *compound clause* combines any number of subclauses together, using the same logical operator: AND or OR. (The NOT operator is not supported.) The subclauses to be combined can, themselves, be either simple or compound.

Examples and notes:

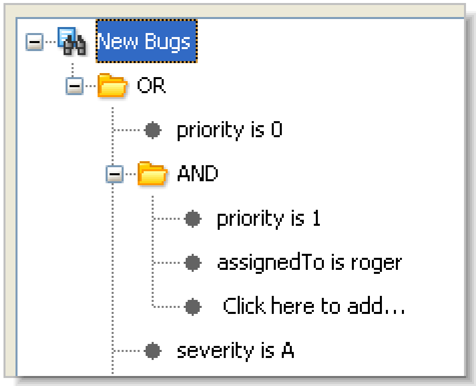
Example 1: simple AND simple



Example 2: simple AND compound

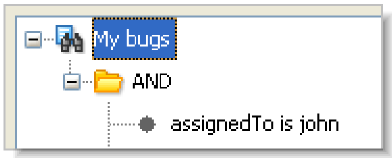


Example 3: simple OR compound OR simple

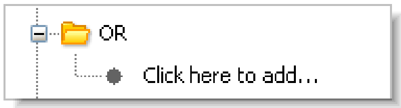


Example 4: Compound clause with a single subclause

A compound clause can contain a single subclause. This is logically equivalent to using the subclause by itself. In fact, the standard "one-line" query contain an AND clause with one simple subclause.

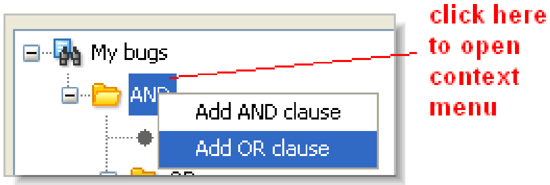


The query editor automatically creates placeholders for simple clauses. But you must explicitly insert a compound-clause placeholder yourself, then fill in the subclauses.



Following are guidelines for adding placeholders to a query:

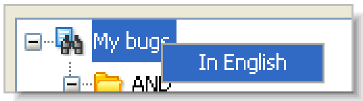
- You cannot add a compound clause at the top level (directly beneath the query name). This level contains a single compound clause -- initially AND, but changeable to OR .
- To create a compound clause below a particular operator, right-click it, and select the type of clause from the context menu.



- To change any AND clause to an OR clause (or vice-versa), select the operator and press function key **F2** (or click it a second time). This opens a listbox with the values AND and OR.

Viewing an English-Language Rendition of the Query

The query name's context menu includes the option **In English**, which renders the query in everyday language, displaying the results in a separate window.



Ending the Query Editor Session

Whether you're composing a new query or revising an existing one, you end by saving your work (**Ok** button) or discarding it (**Cancel** button). If you didn't cancel the Query Editor session, AccuWork automatically executes the query and displays the results in the Query Results pane. **Note:** Closing the window using its title bar controls is equivalent to **Cancel**.

Set Up Columns

Set Up Columns specifies a set of fields from AccuWork, and the order in which they will appear as columns in a particular table.

Note: What about column widths?

> You don't specify column widths with this option. After finishing this operation, you can drag column separators in the tab from which you invoked **Select Columns**. You can also drag-and-drop the column headers to rearrange the columns.

Invoking Set Up Columns

The **Set Up Columns** button appears in the Queries tab, in the Query Results pane toolbar. Use it to specify the format of the table that shows the results of a user-defined AccuWork query.

The **Set Up Columns** button also appears in the following areas of the Schema Editor (see *Appendix A The Schema Editor (Administrators Only)*):

- *Change Packages tab (Change Package Results pane)*

  Specifies the format of the tables produced by ***Show Issues*** and ***Show Diff by Issues***.

- *Change Packages tab (Change Package Triggers pane)*

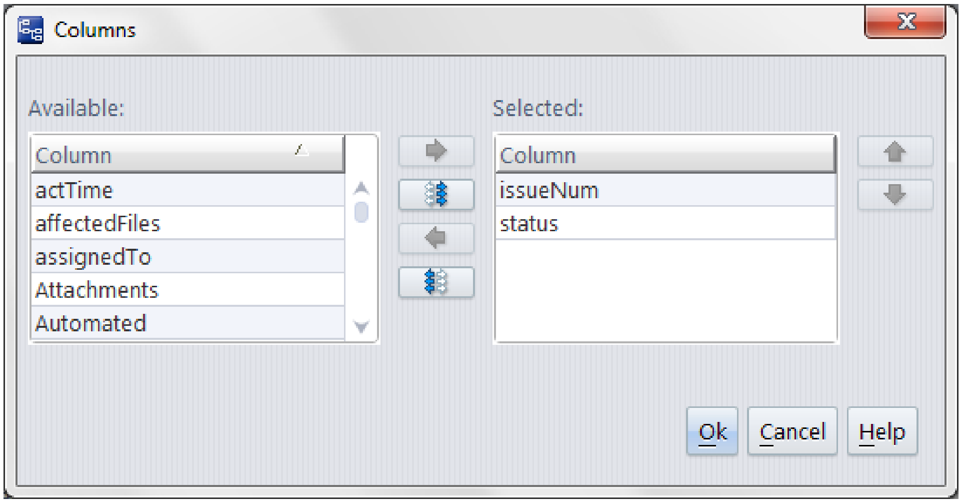
  Specifies the format of the table displayed by the change-package-level integration between AccuRev and AccuWork.

- *Relationship Types tab*

  Specifies the format of the table that shows the set of issue records related to a given issue record.

Using the Columns Dialog

The fields listed under **Selected** when you click **Ok** will be displayed in the appropriate table of issue records.



- To move field names from the Available list to the Selected list (or vice-versa), select one or more names, then click the ⇨ button (or the ⇦ button).
- To move all field names between the lists, click the ⇉ or ⇇ button.
- To rearrange the Selected list, select one or more names, then click the ⇧ or ⇩ button.

Notes:

Field names vs field labels

> The names that you work with in the Columns dialog are the actual **field names** (Name column of the Schema subtab of the AccuWork Schema Editor) In the Query Results pane, the column headers are the corresponding **field labels** (Label column of the Schema subtab).

Further adjustments, including column-width specifications

> You don't specify column widths in the Columns window, just their order. You can adjust column widths and change the order of columns: see *Working with Tables* on page 14.

Multiple-line text fields

> You can include a multiple-line text field in the set of columns to be displayed in the Query Results pane, but it's likely that you won't be able to see the complete contents of such fields, no matter how wide you make its column. But when you display an HTML file created by **Export Table** in a web browser, you'll see the complete contents of every field.

When you click **Ok** to close the Columns dialog, the changes you've made are reflected immediately in the Query Results pane. But the changes to the query's columns setup are not saved in the repository until you invoke 💾 **Save All** in the Query List pane.

10.Change Packages

Change packages are the link between AccuRev's SCM functionality and the AccuWork issue management system.

A change package contains a set of entries, each of which (usually) has the same form as a patch (see Patch and Patch From on page 227). But change package entries can be *modified* (added to) after they are first created. Such modified entries don't fit the "recent changes" description of a patch. Rather, a change package entry can act like an accumulator, recording *all* the changes made to the element to fulfill a particular development task. The change package facility was designed to track exactly what element-by-element changes go into the bug fix or feature that the issue record describes, and to answer questions such as these:

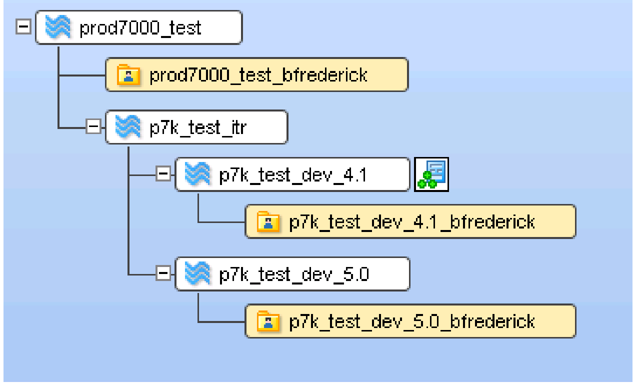
> "What were all the changes made to source files in order to fix bug #457?"
>
> "Have all the changes made to fix bug #457 been promoted to the QA stream?"

Automatic recording of change package entries is enabled through the change-package-level integration between AccuRev configuration management and AccuWork issue management (see the administrator information at *Integrating Configuration Management and Issue Management* on page 383).

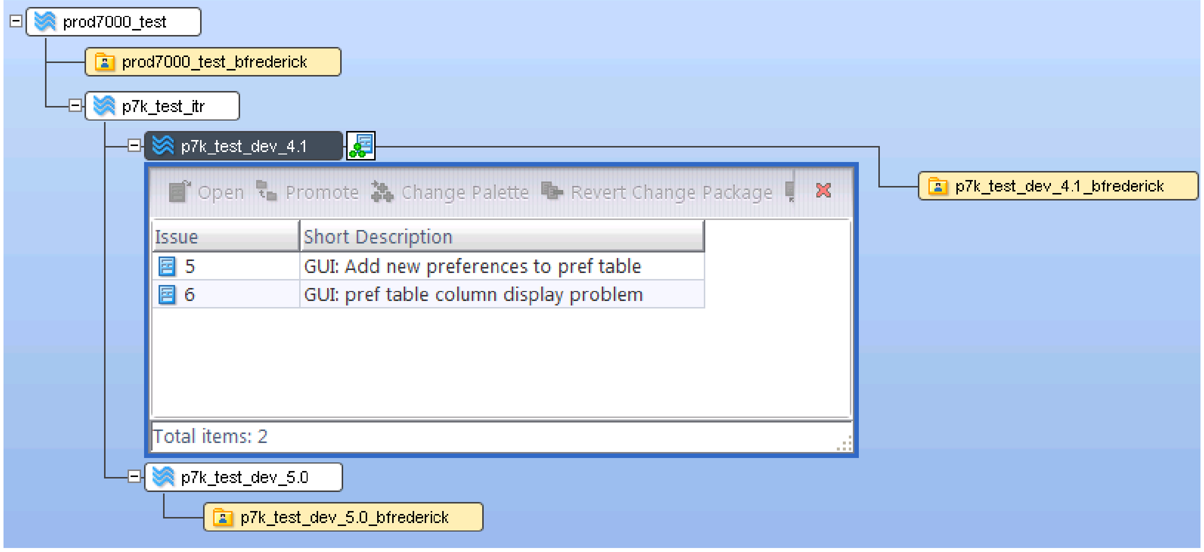
A Change Package Example

The following relatively simple yet realistic example illustrates the benefits of AccuRev change packages in the development environment.

1. Assume a development environment where one stream is dedicated to enhancements and bug fixes for a "point" release ("Release 4.1"), and a parallel stream is dedicated to the next major release ("Release 5.0").



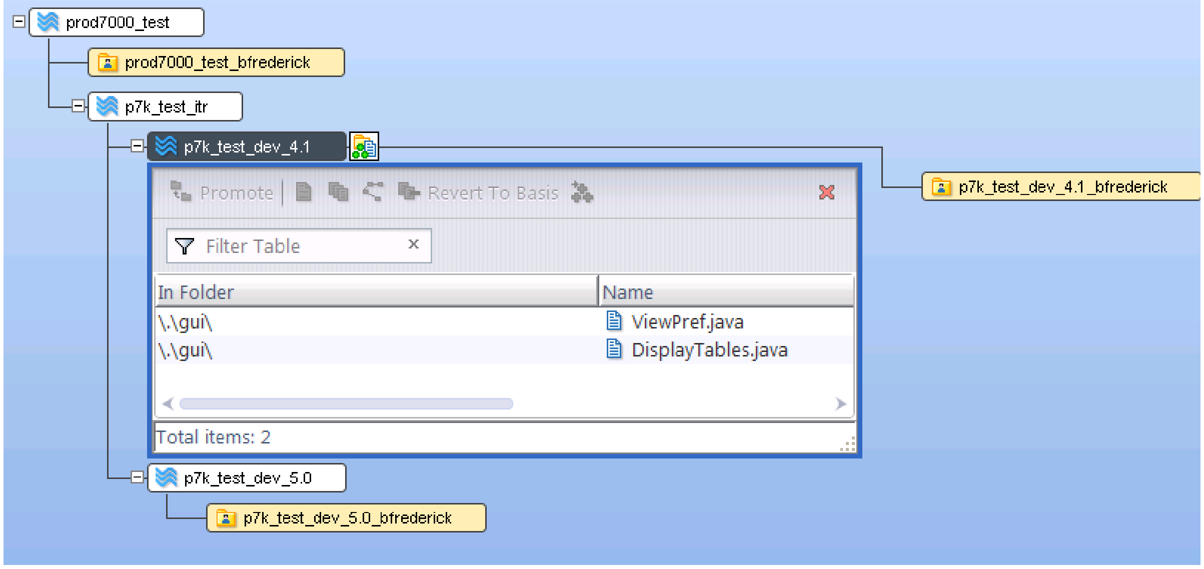
2. In the 4.1 stream, two issues are active: one for an enhancement to a GUI display, and another for a bug fix to the same area of the code:



Issue	Short Description
5	GUI: Add new preferences to pref table
6	GUI: pref table column display problem

Total items: 2

3. These issues affect two files. The enhancement (Issue #5) touches both ViewPref.java and DisplayTables.java. The bug fix (Issue #6, which was promoted before the final Issue #5 enhancement changes were promoted) touches only DisplayTables.java.

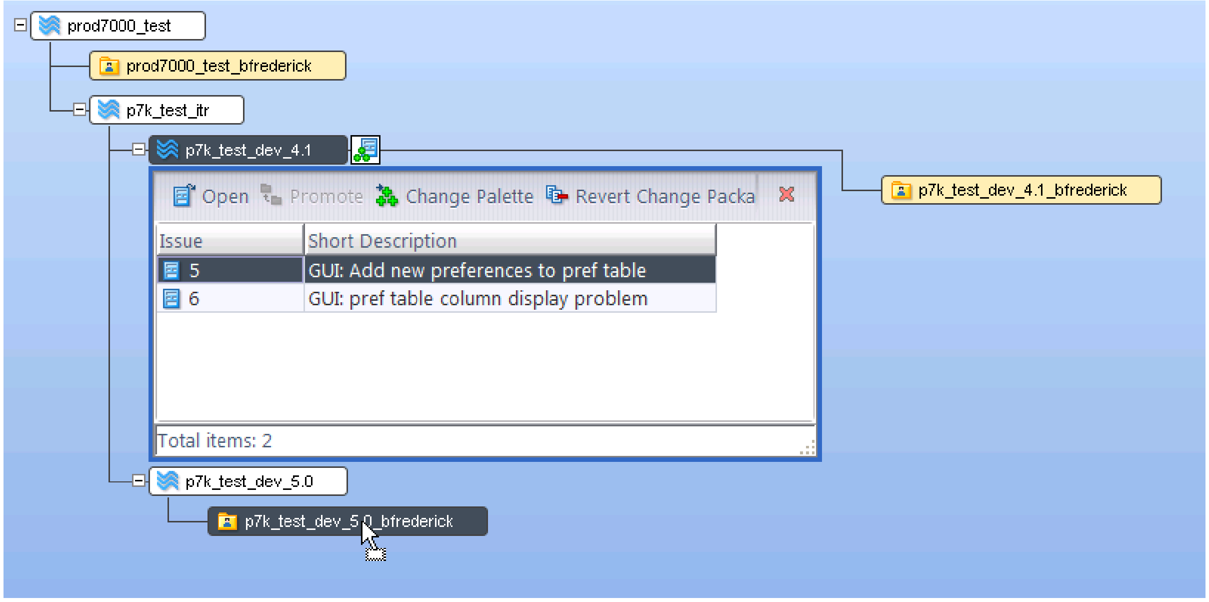


In Folder	Name
\.\gui\	ViewPref.java
\.\gui\	DisplayTables.java

Total items: 2

4. The team working on Release 5.0 wants to incorporate the changes done by the 4.1 team into their own code. If 4.1 was already completed and released, the 4.1 changes would have already been promoted up to the integration stream (the common parent for the development streams), and the changes would be automatically inherited by the 5.0 stream. However, things do not always work in such an orderly manner in the software development world, so the 5.0 team decides to cross-promote Issue #5 from the 4.1 development stream to one of the workspaces under the 5.0 stream. When you cross-promote an issue, you cross-promote the elements of the change package associated with that issue. So this operation brings the 4.1 versions of ViewPref.java and DisplayTables.java associated with Issue 5 into

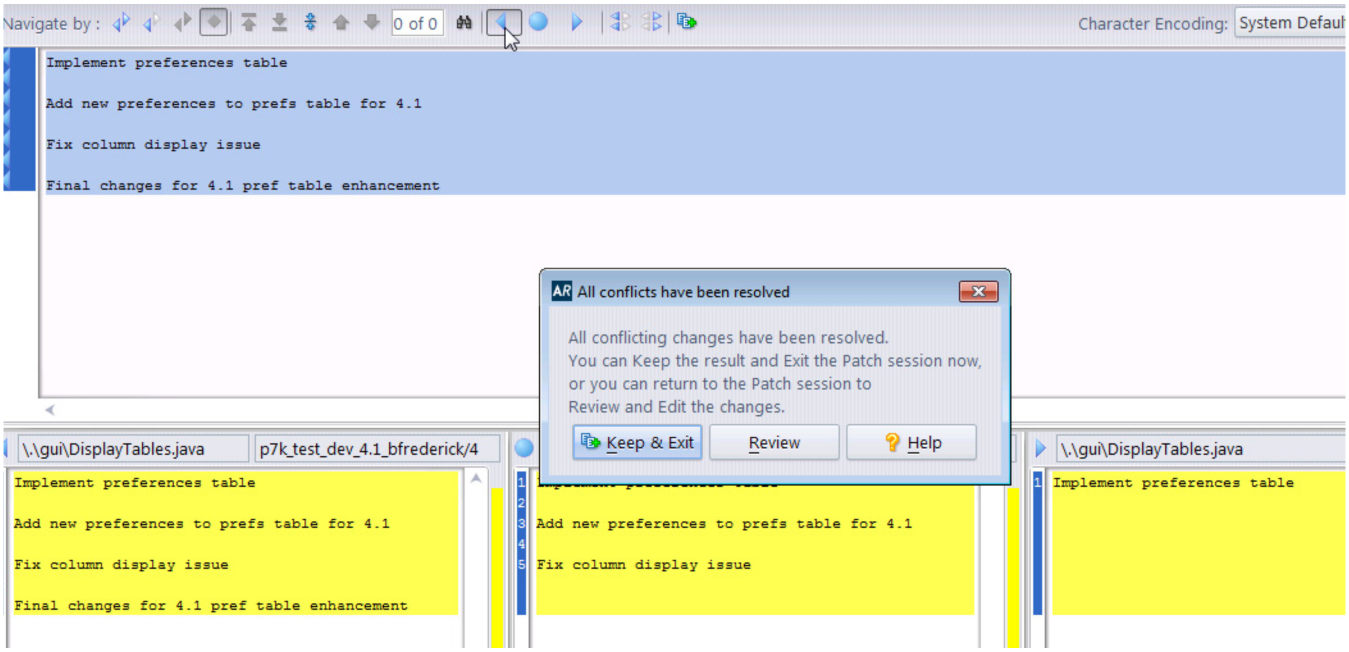
the 5.0 stream structure. Perform the cross-promote with a simple drag & drop of Issue #5 from the 4.1 stream Active Issues view to the 5.0 workspace:



Issue	Short Description
5	GUI: Add new preferences to pref table
6	GUI: pref table column display problem

Total items: 2

5. If this had been a simple cross-promote of changes associated with a single issue, we would be done. But as noted before, software development is rarely simple. Recall that one of the files had also been modified for a bug fix under a different issue. AccuRev detects this complication and automatically displays a diff and merge utility so that you can determine what changes you want to take (see *The AccuRev Merge Tool* on page 216). You decide to take all the changes, including the bug fix, and keep the file. (The example below doesn't show actual code.)


Navigate by :
0 of 0
Character Encoding:
System Defaul
\.\gui\DisplayTables.java
p7k_test_dev_4.1_bfrederick/4
\.\gui\DisplayTables.java
Keep & Exit
Review
Help

```
Implement preferences table

Add new preferences to prefs table for 4.1

Fix column display issue

Final changes for 4.1 pref table enhancement
```

All conflicts have been resolved

All conflicting changes have been resolved.
You can Keep the result and Exit the Patch session now,
or you can return to the Patch session to
Review and Edit the changes.

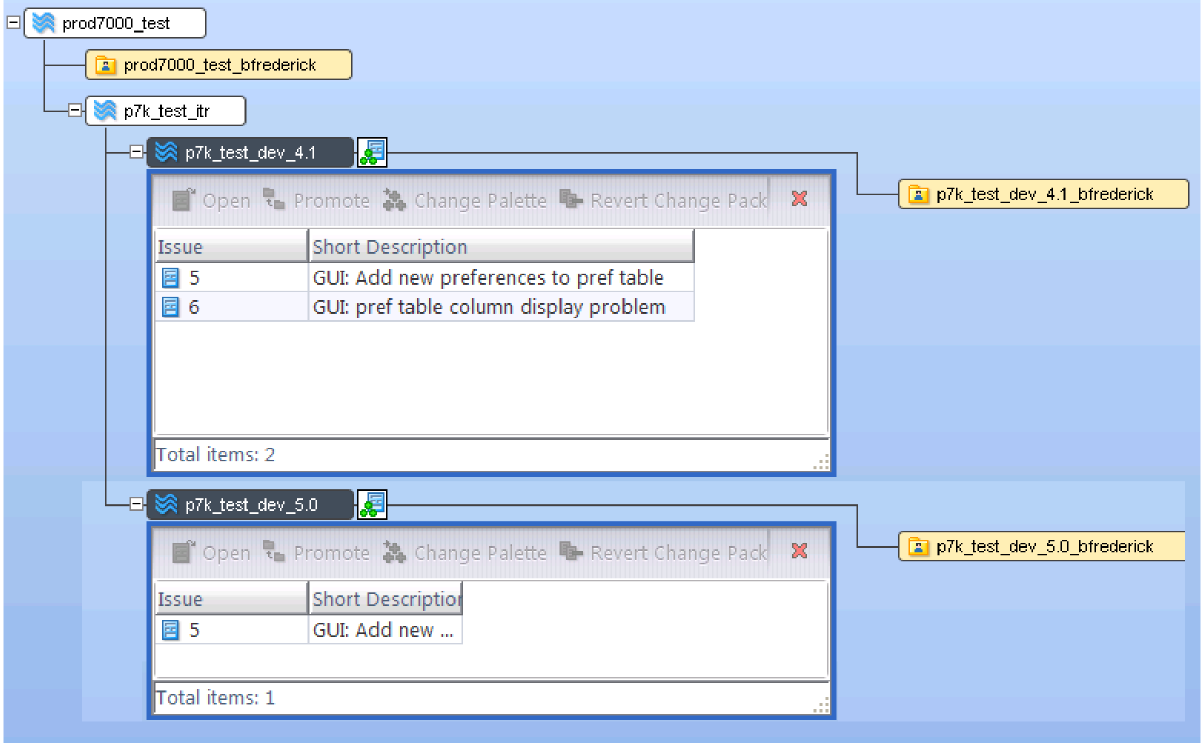
```
Implement preferences table

Add new preferences to prefs table for 4.1

Fix column display issue

Final changes for 4.1 pref table enhancement
```

```
Implement preferences table

Add new preferences to prefs table for 4.1

Fix column display issue
```

```
Implement preferences table
```

6. Finally, the 5.0 team member makes additional changes to one of the files for 5.0, and promotes the files against the original issue (#5). The end result is that Issue #5 is active in both streams, and is not "incomplete" in either stream. (Experienced AccuRev users will note that it is no longer necessary to

create separate issues for this scenario as of AccuRev Release 6.0, and that they encounter “incomplete change packages” less often than in the past.)



Issue	Short Description
5	GUI: Add new preferences to pref table
6	GUI: pref table column display problem

Total items: 2

Issue	Short Descriptio
5	GUI: Add new ...

Total items: 1

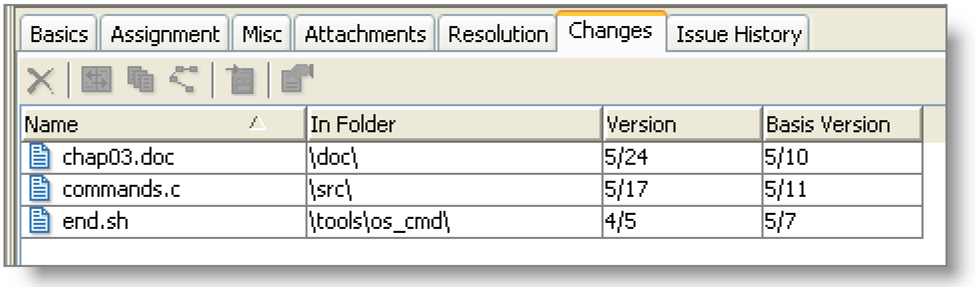
Note: It is still possible to encounter incomplete change packages. For example, if the cross-promote had not automatically invoked the merge tool, the new version in the 5.0 stream would not have been available to the 4.1 stream, and Issue #5 would have become “incomplete”. How do you fix this? Just cross-promote Issue #5 back to the 4.1 stream after you make your changes.

Change Package Details

The key is to go beyond thinking of individual versions to considering change packages, which are collections of versions. In AccuRev, change packages are implemented by AccuWork issue records. An issue record records the details of a bug or feature: its description, how important it is, who originated it, who's working on it, and so on. An issue record can also keep track of the changes that have been made to elements, in order to implement that particular bug fix or new feature.

For example, issue record #9 might contain a bug report, "Circles are not round". The bug fix involves changes to three elements:

When viewing the issue record through its form, go to the Changes tab to view the change package. In this example, the change package contains three entries:



Name	In Folder	Version	Basis Version
chap03.doc	\doc\	5/24	5/10
commands.c	\src\	5/17	5/11
end.sh	\tools\os_cmd\	4/5	5/7

change package for issue record #9 contains three changes

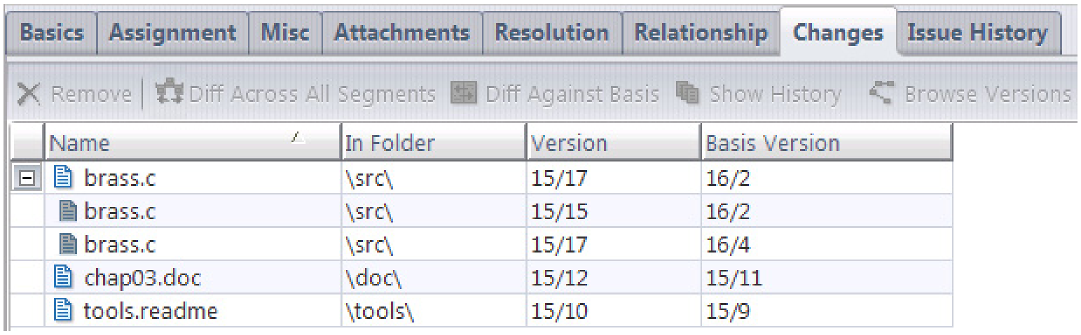
- The set of changes that were made between version 5/10 and version 5/24 of element *chap03.doc*
- The set of changes that were made between version 5/11 and version 5/17 of element *commands.c*
- The set of changes that were made between version 5/7 and version 4/5 of element *end.sh*

(See *Patch and Patch From* on page 227 for the precise definition of "between".)

In the simplest terms, you create a change package when you promote one or more elements out of your workspace, and specify an issue for those elements to be associated with. Thereafter, you and other AccuRev users can (and should) promote by issue to promote the changes further up the stream hierarchy. (In general, you should avoid promoting by file once you adopt change packages.) You can also cross-promote and use “Send to Issue” to modify change packages outside of the strict, hierarchical promote structure.

Change Package Display

In the **Changes** subtab of an issue record's form (and equivalently, in a *Stream Issues* tab), a change package is displayed in a table, each row showing one change package entry. Example:



Name	In Folder	Version	Basis Version
brass.c	\src\	15/17	16/2
brass.c	\src\	15/15	16/2
brass.c	\src\	15/17	16/4
chap03.doc	\doc\	15/12	15/11
tools.readme	\tools\	15/10	15/9

An element that has been promoted against different issues intermittently is displayed with a plus (+) sign, which expands to display all the *segments* that are included in this change package.

As with a patch, each change package entry has a head version (shown in the “Version” column) and a basis version.

Structure of a Change Package

A change package is a collection of element versions; for example:

version **kestrel_dvt_jjp/13** of element **/./src/brass.c**
version **kestrel_dvt_jjp/14** of element **/./src/brass.h**
version **kestrel_dvt_jjp/16** of element **/./src/commands.c**

A change package can contain entries for any number of elements.

Elements can have multiple entries showing *segments* -- versions between a head and a basis representing promotion(s) against different issues while this issue has been active. (Elements with such multi-segments are shown collapsed by default, and can be expanded by clicking the plus (+) icon.)

The elements must belong to the same depot as the issue database. Each change package entry is similar to a patch -- it consists of all the element's versions between a designated head version and basis version.

The basic idea is that this set (or "package") of versions contains all the changes required to implement a certain development project. But we need to refine this idea. Consider that version 14 of **brass.h** probably contains *more* than just the changes for that development project. For example:

- Versions 1-7 might have been created years ago, when the product was first developed
- Versions 8 and 9 might have been minor tweaks, performed last month
- Versions 10-14 are the only versions with changes for the development project in question

So we need a way to express the idea that only the "recent changes" to **brass.h**, those in versions 10-14, are to be included in the change package. AccuRev accomplishes this by defining each change package entry using two versions: a user-specified head version and an older, automatically-determined basis version.

> **Note:** As of AccuRev 6.0 and the introduction of "multi-segments", users can also specify basis versions under certain circumstances.

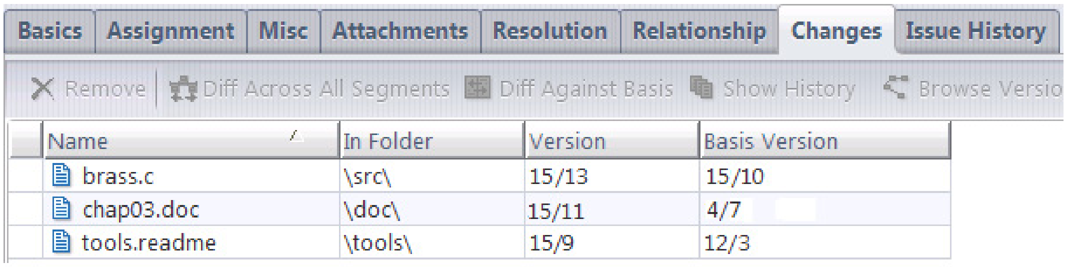
The "recent changes" to be included in the change package were made by starting with the basis version (version 9 in this example) and performing a **Keep** on one or more new versions (versions 10, 11, 12, 13, and 14 in this example).

In the AccuRev GUI, the head version of a change package entry is usually identified simply as the "Version".

> **Note: Patch** uses the same "recent changes" analysis to determine which changes in the "from" version are to be incorporated into the "to" version.

The AccuWork issue-management facility provides a logical place for keeping track of development activities for **brass.h**. Each task -- fixing a bug, creating a new feature, etc. -- is tracked by a particular AccuWork issue record. So AccuRev uses using issue records to implement change packages.

Each issue record includes a **Changes** section that acts as an "accumulator" for versions' changes. Here's how the above example of a change package would appear in an issue record's form:

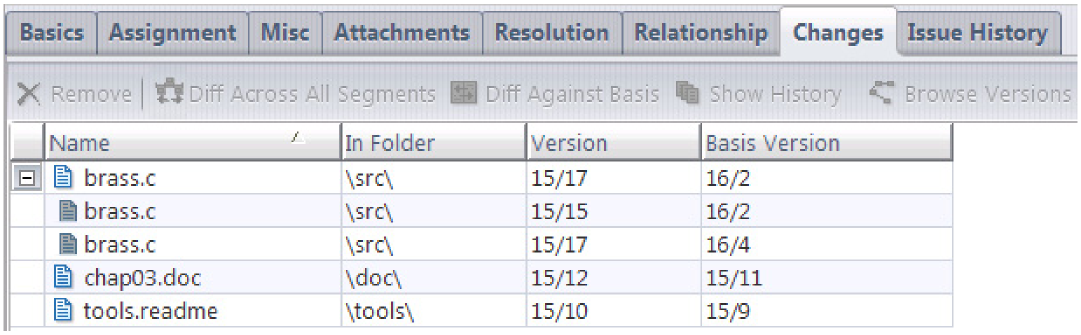


Name	In Folder	Version	Basis Version
brass.c	\src\	15/13	15/10
chap03.doc	\doc\	15/11	4/7
tools.readme	\tools\	15/9	12/3

This change package has entries for three elements:

- **brass.c**: The basis version, 15/10 was created in the user's own workspace. This indicates that the user promoted 15/10 to the backing stream. AccuRev assumes that this change was for another task, not the one covered by this issue record. Then, the user turned his attention to the current task, creating additional versions up to and including 15/13, the head version.
- **chap03.doc**: This change began when the user updated his workspace, bringing in version 4/7 of the element (which had originally been created in another workspace, then was promoted to the backing stream). Then, the user created one or more versions in his own workspace, up to and including version 15/11, the head version.
- **tools.readme**: Similarly, this change began when the user updated his workspace, bringing in version 12/3, originally created in another workspace. The user created one or more versions in his workspace, ending with version 15/9, the head version.

But what happens if these files get changed again for a new release, and one of the files also gets changed in the meantime by another developer for a support issue? Fast-forward a year or so:



Name	In Folder	Version	Basis Version
⊟ brass.c	\src\	15/17	16/2
brass.c	\src\	15/15	16/2
brass.c	\src\	15/17	16/4
chap03.doc	\doc\	15/12	15/11
tools.readme	\tools\	15/10	15/9

The change package for this new issue now includes multiple entries for one of the files, due to the introduction of *multi-segments* -- the ability of AccuRev change packages to track and correctly handle versions that are not necessarily related to the current changes.

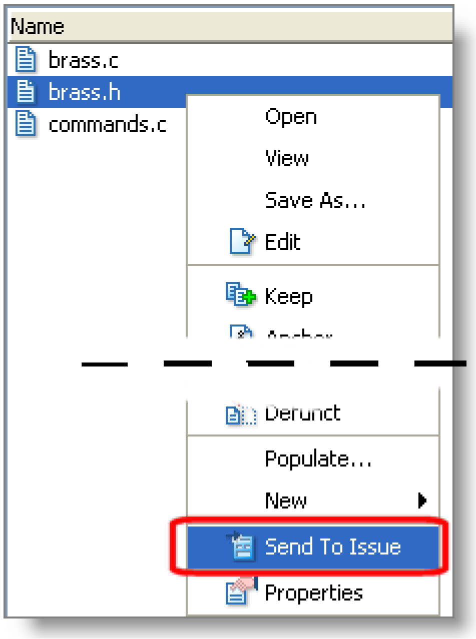
- **chap03.doc**: This change is basic -- the previously promoted version was updated and then promoted.
- **tools.readme**: Similarly, this file was updated from its backed version and promoted.
- **brass.c**: This file starts to get interesting -- it was updated and promoted against a different issue prior to being promoted for the current issue. Each set of versions between the basis and head version are recognized and treated as a separate *segment*.

  Note: If you used AccuRev and AccuWork prior to Release 6.0, you needed to follow processes that prevented this situation, such as promoting a file only against a single issue at a time. AccuRev now keeps track of such promotions automatically through the use of multi-segments. You and your colleagues can work on the same files for different issues at the same time (merging as necessary), and AccuRev keeps track of the change package records for you.

Creating a New Change Package Entry

AccuRev records change package entries automatically, whenever **Promote** is invoked in a workspace. For example, suppose issue record #3 represents a particular bug (and its fix). Whenever a developer promotes one or more versions whose changes address that bug, he specifies issue #3 at a prompt. AccuRev automatically creates a change package entry in issue #3 for each promoted version.

Also, you can manually add entries to a change package: right-click a version in the File Browser, Version Browser, or History Browser, and then select **Send to Issue** from the context menu. The selected version becomes the head version of the change package entry. AccuRev automatically determines the corresponding basis version. As the examples above suggest, AccuRev uses an algorithm that determines the set of “recent changes” to the element, made in a single workspace.



In the Version Browser, a variation of this operation, **Send to Issue (specifying basis)**, enables you to pick the basis version, rather than allowing AccuRev to determine it automatically.

You can also invoke **Send to Issue** on the Changes tab of an issue record. This copies an existing change package entry to a different change package (issue record).

Also, with the introduction of “multi-segments” in AccuRev 6.0, there are certain other circumstances in which users can specify basis versions.

Modifying Existing Change Package Entries

When you want to create a change package entry for a particular element, but an entry for that element already exists in the change package, AccuRev attempts to combine the new entry with the existing one. This produces an updated entry that includes all the changes. AccuRev always attempts to coelesce versions that are contiguous. For example, if version 3, 4, and 5 are combined with versions 6 and 7, they will be coalesced to versions 3 through 7. However, if version 3, 4, 5 are combined with versions 8 and 9, while versions 6 and 7 belong to a different change package, 3, 4, and 5 will be recorded as one segment in the change package, while versions 8 and 9 will be added as a separate segment.

Change Packages "In" Streams

As the versions in a change package are promoted up the stream hierarchy, the change package itself implicitly moves up the hierarchy, also. Roughly speaking, a change package has risen to a certain level if all its entries have risen to that level. More precisely, the change package entry for an element is said to be "in" a particular stream if the element's version in the stream is -- or is a descendent of -- the head version.

The reason that the descendant of the head version qualifies is that AccuRev assumes that changes made to an element *remain* in that element as subsequent versions are created. Suppose a change package entry consists of versions 13/4, 13/5, and 13/6 of some element. Another user then brings version 13/6 into her workspace with an *Update*, and creates descendent version 49/2. It's fair to say that the change package entry is "in" her workspace, and in any stream to which version 49/2 is promoted.

And the reason that the head (but not the basis) version is considered is that the basis version of a change package entry is always an ancestor of the head version. AccuRev assumes that changes that were present in the ancestor version remain in the later version.

In summary:

- A change package is said to be "completely in" a particular stream if all of its entries are "in".
- A change package is said to be "partially in" a particular stream if some -- but not all -- of its entries are "in". See *Incomplete Change Packages* on page 344.
- A change package is said to be "active" in a particular stream if at least one of its head versions is in the stream's default group.

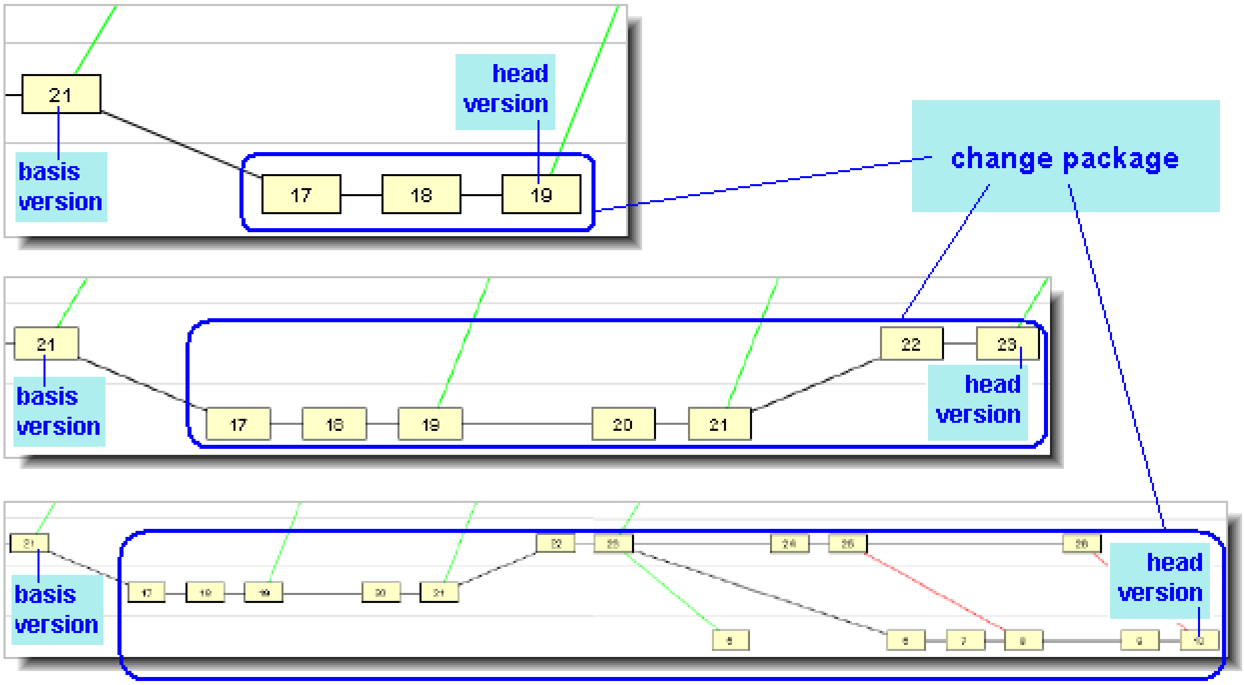
Complex Change Package Entries

All change package entries are recorded in terms of real versions (those created in users' workspaces), even though there may be corresponding virtual versions (created by promoting the real versions from workspaces to dynamic streams). In all the examples shown above, each change package entry is a series of consecutive real versions created in the same workspace — that is, each change package entry records a particular patch to the element.

But the change package facility can also track *ongoing* changes to elements — changes made at different times, and in different workspaces. To support this capability, AccuRev defines a change package entry in a more general way than a patch:

> A change package entry for an element consists of all the real versions in the element's version graph between a specified basis version and a specified head version. Between-ness is determined both by direct predecessor-successor connections (created, for example, by **Keep**) and by merge connections (created by **Merge**). Patch connections are *not* considered in this determination; the basis version itself is not part of the change package entry.

The following Version Browser excerpts show the range of complexity that a change package entry can have. In fact, these excerpts show how the same change package entry can change over time, becoming more complex.



These illustrations suggest the following definition for a change package entry, which is equivalent to the definition above:

> A change package entry for an element consists of the element's entire version graph up to the specified head version, *minus* the entire version graph up to the specified basis version. For these purposes, the version graph includes direct predecessor-successor connections and merge connections, but not patch connections.

Notation

To help explain how AccuRev performs "change package arithmetic" to combine and update entries, we'll use a simple notation. Suppose a change package entry contains the set of an element's versions defined by these specifications:

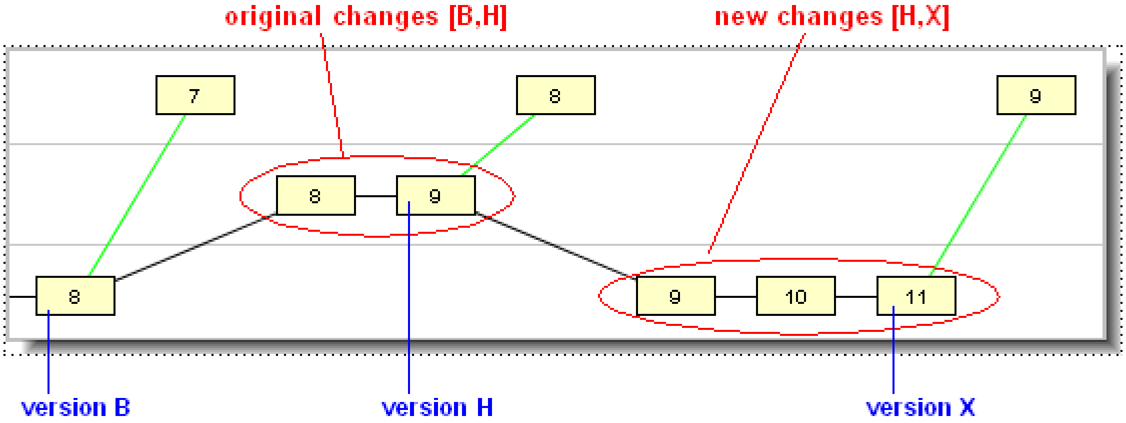
> the head version is **H**
> the basis version is **B**

We'll use the ordered pair **[B,H]** to indicate this change package entry.

Combining Two Change Package Entries

Now, suppose a new change is to be combined with the existing change package entry **[B,H]**. There are several cases, each handled differently by AccuRev:

- **Case 1: [B,H] + [H,X]** — This simple case typically arises when you think you're done with a task and record your work as change package entry **[B,H]** — but it turns out that more work on the same element is required. So if you (or a colleague) start where you left off, with version **H**, and make changes up to version **X**. Then, you want to incorporate the new set of changes **[H,X]** into the same change package.



In this case, it's clear that the two series of changes can be viewed a single, uninterrupted series — starting at version B and ending with version X. That is:

[B,H] + [H,X] = **[B,X]**

Accordingly, AccuRev updates the change package entry automatically — keeping **B** as the "Basis Version" and changing the "Version" from **H** to **X**.

- **Case 2: [B,H] + [J,X]** (where H is an ancestor of J: "change package gap") — This case typically arises when you do work on a task at two different times, and someone else has worked on the same element in between.

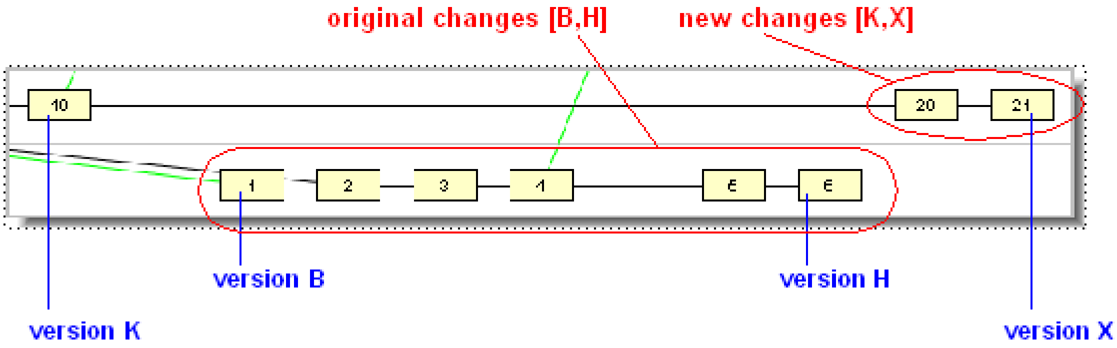
  In this example, a colleague updated her workspace to bring in your original changes, created versions 9 and 10 in her workspace, and promoted her changes. You then updated your workspace to bring in her changes, and made a new set of changes.

  When AccuRev tries to combine the change **[B,H]** and the change [**J,X]** into a single change package entry, it detects that version **H** and version **J** are not the same, but that **H** is a direct ancestor of **J**. Thus, there is a simple "gap" in the potential combined change package entry (in this example, consisting of your colleague's versions 9 and 10).

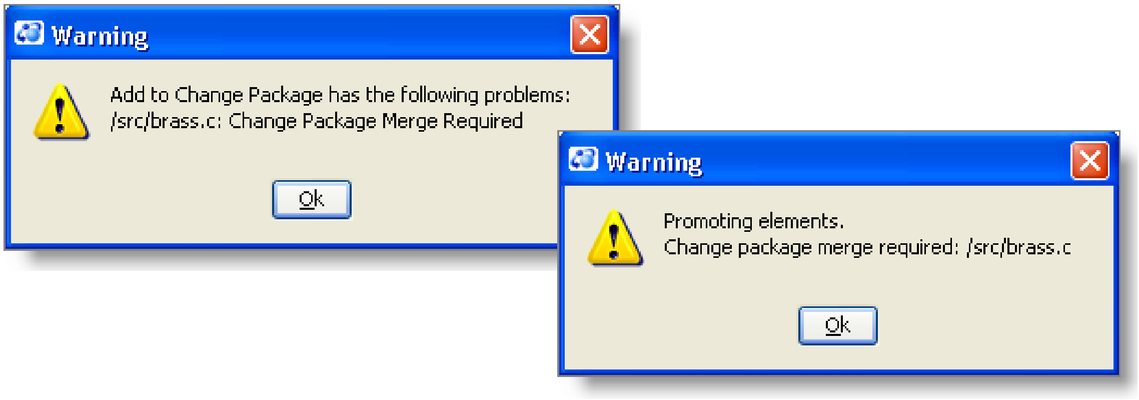
  Probably, your colleague was not working on the same task when she made her changes. (If she had been, she would have added her changes to the same change package, as in Case 1.)

  Accordingly, AccuRev treats the two change set entries as separate segments, and the in-between versions that are associated with other work can be safely ignored.

- **Case 3: [B,H]** + **[K,X]** (where H is *not* an ancestor of K: "change package merge required") — This case typically arises when developers in workspaces that do not share the same backing stream try to use the same change package. There is no simple "gap" between the existing change package entry and the new one — which means there is no way to combine them into a single change package entry, according to definitions in *Complex Change Package Entries* on page 331.



AccuRev signals this situation with a “change package merge required” message, and cancels the current operation.



Add to Change Package has the following problems:
/src/brass.c: Change Package Merge Required

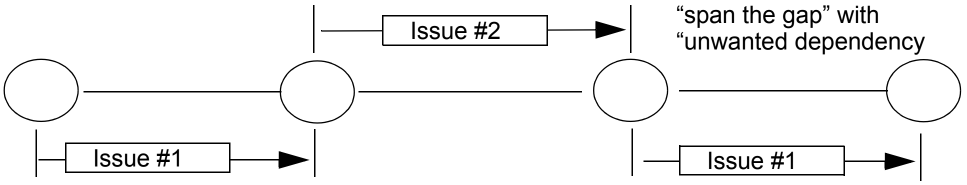
Promoting elements.
Change package merge required: /src/brass.c

You can remedy this situation by performing a merge at the element level. (There is no merge operation defined at the change package level.) In the example above, merging version H and version X would create a new version; a change package entry with the new version as its head can be combined with the existing entry.

Change Packages, Segments, and Dependencies

An issue dependency occurs when one issue has some kind of requirement on the contents of another issue. For example, consider a file that is being edited by two different people for two different issues. One person edits the file and promotes it under Issue #1. The second person takes this updated version, makes further edits and promotes it under Issue #2.

A problem can occur when the first person then wants to make additional changes to the same file under Issue #1. The changes from Issue #2 are now an *unwanted dependency* for Issue #1: They might not have anything to do with Issue #1, but they must be included by “*spanning the gap*” in any subsequent promote operation.



Not only does Issue #2 have nothing to do with Issue #1, but prior to AccuRev 6.0, it became a part of Issue #1 after “spanning the gap”. If Issue #2 has, say, three other files, then those would become dependencies of Issue #1 that are not desired.

AccuRev 6.0 can now handle this situation for you through the introduction of *multiple segments*.

AccuRev treats contiguous versions of a file as a *segment*. Each segment can be tracked and handled independently. That way, multiple promotes for Issue #1 can be tracked, and having a “gap” caused by an intervening promote of Issue #2 does not cause a problem. You can think of each segment almost as a separate patch (not the AccuRev patch command, but a generic, development patch).

AccuRev handles all the record-keeping automatically, and you no longer need to deal with the changes introduced by Issue #2.

> **Note:** Issue #1 still has dependencies on Issue #2, but the dependency can now be addressed by a patch operation.

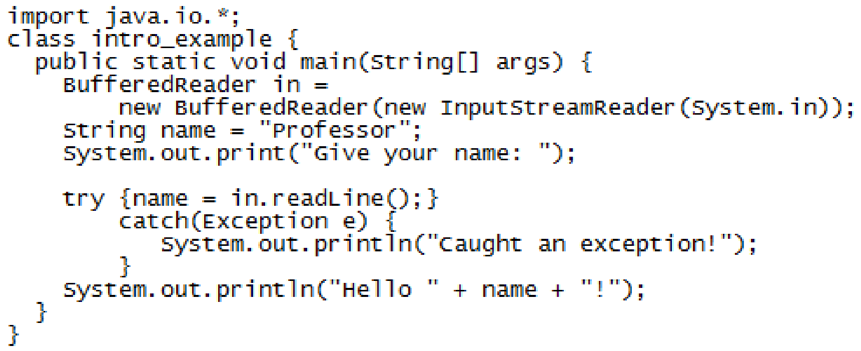
Comparing Versions Across Segments

If a text file has multiple segments because it has been promoted against different issues and you want to compare only the versions of a file that were promoted against the *same* issue, you can accomplish this by using the **Diff Across All Segments** option. When you use this feature, the diff shows all changes to the file that were promoted against a single issue while ignoring any changes to the file that were promoted against a different issue.

To understand the purpose of the **Diff Across All Segments** option, imagine the following scenario:

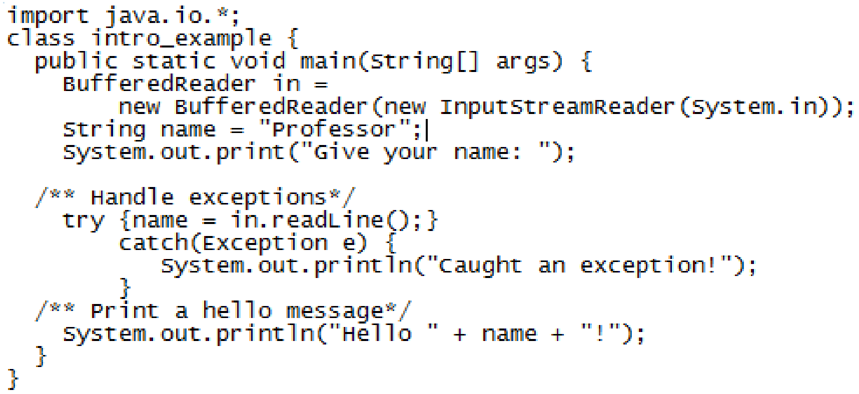
1. You create a file and promote it against issue 5.
2. A co-worker makes changes to the file for unrelated reasons and promotes the file against a *different* issue, say issue 7.
3. You make further edits to the file and promote it against issue 5 once more.
4. You can view all the edits that you have made to the file by opening issue 5, selecting the **Changes** tab, selecting the file, and choosing **Diff Across All Segments**. In this case, AccuRev displays a diff that shows only those changes made to the file that were promoted against issue 5.

For example, suppose you create the following file called **intro_example.java**:



```
import java.io.*;
class intro_example {
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Professor";
    System.out.print("Give your name: ");

    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!");
        }
    System.out.println("Hello " + name + "!");
  }
}
```

You then promote **intro_example.java** to issue 5. At this point, a co-worker, who has been assigned to add comments to source code, edits **intro_example.java** and adds comments so that the edited file now appears as follows:

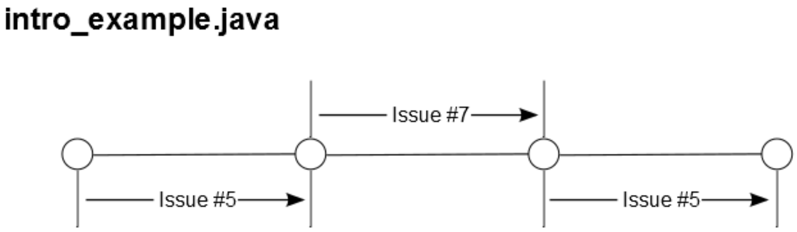


```
import java.io.*;
class intro_example {
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Professor";|
    System.out.print("Give your name: ");

  /** Handle exceptions*/
    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!");
        }
  /** Print a hello message*/
    System.out.println("Hello " + name + "!");
  }
}
```

Your co-worker now promotes **intro_example.java** against issue 7.

At this point, you make more edits to **intro_example.java** in order to create a Spanish version of the file and you once again promote the file against issue 5.

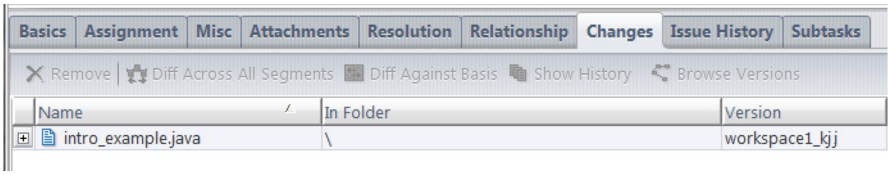
In this example, **intro_example.java** is now composed of the following segments:



To view all the edits you have made to **intro_example.java**, and *only* those edits, you would take the following steps:

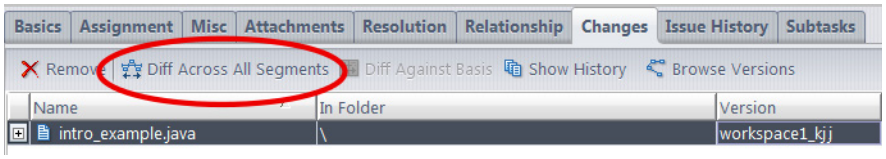
1. Select **Issues > Open Issue**.
2. In the Look Up Issue dialog box, enter **5** in the **Enter Issue Number** field and click **Ok**.
3. Select the **Changes** tab.

   The file **intro_example.java** appears with a plus sign (+) aside it, indicating that the file contains multiple segments:



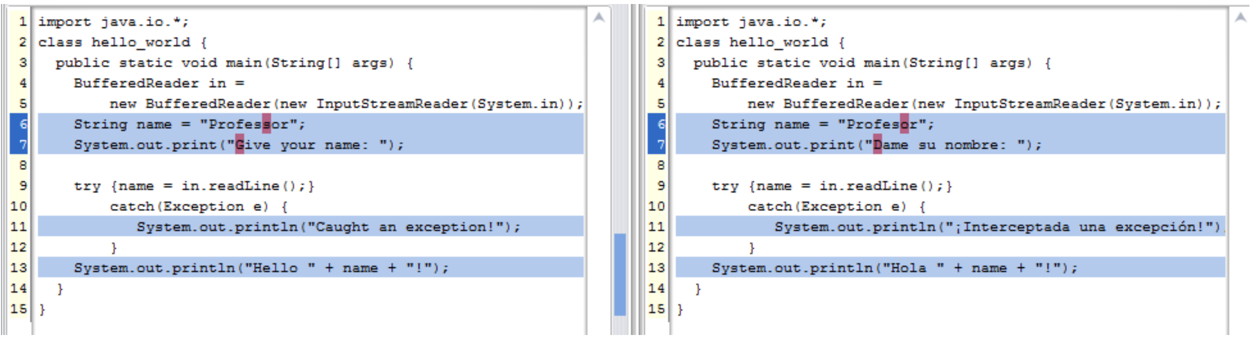
Name	In Folder	Version
⊞ intro_example.java	\	workspace1_kjj

4. Select the **intro_example.java** file and choose **Diff Across All Segments**:



Name	In Folder	Version
intro_example.java	\	workspace1_kjj

The resulting diff shows all the changes to the file that have been promoted against issue 5:



```
import java.io.*;
class hello_world {
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Professor";
    System.out.print("Give your name: ");

    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println("Caught an exception!");
        }
    System.out.println("Hello " + name + "!");
  }
}
```

```
import java.io.*;
class hello_world {
  public static void main(String[] args) {
    BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));
    String name = "Profesor";
    System.out.print("Dame su nombre: ");

    try {name = in.readLine();}
        catch(Exception e) {
           System.out.println(";Interceptada una excepción!")
        }
    System.out.println("Hola " + name + "!");
  }
}
```

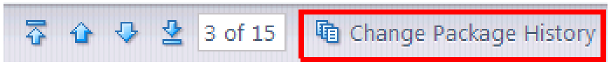
The changes shown do not include the comments that were added to the file and promoted against issue 7.

> **Note:** Under some conditions, changes made to a file that were promoted against a *different* issue may appear when you perform a **Diff Across All Segments** operation. This can occur when the changes that were promoted against a different issue were contiguous to changes promoted against the same issue.

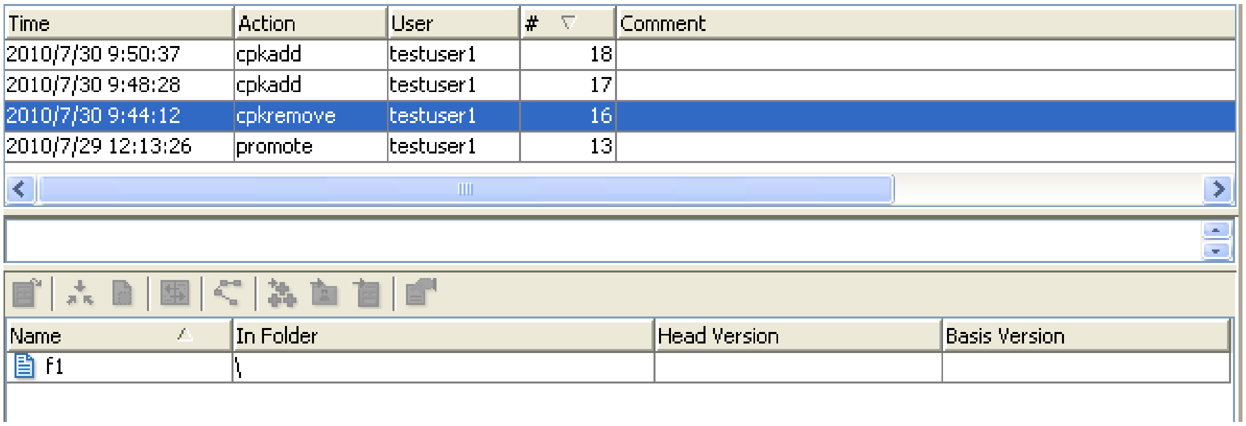
The **Diff Across All Segments** option is inactive for files that contain only a single segment. For more information about other types of diff operations in AccuRev, see *Chapter 7 Diff, Merge, Patch, and the Change Palette*.

Change Package History

How do you display the changes that have been made to a change package? Use the **Change Package History** icon on the Issues toolbar:



This brings up a list of all changes that have been made to the change package, including promotions, and elements that have been added (**cpkadd**) or removed (**cpkremove**).



Time	Action	User	#	Comment
2010/7/30 9:50:37	cpkadd	testuser1	18	
2010/7/30 9:48:28	cpkadd	testuser1	17	
2010/7/30 9:44:12	cpkremove	testuser1	16	
2010/7/29 12:13:26	promote	testuser1	13	

Name	In Folder	Head Version	Basis Version
f1	\		

For information about specific changes, select the transaction in the upper pane and view the elements associated with that change in the lower pane.

Association with Issue Record

An issue record's change package differs from its user-defined fields (Status, Severity, Description, etc.).

When you change the value of a user-defined field, the **Save** button is enabled in the edit-form toolbar. You can discard the change by closing the edit-form tab without performing a **Save**. But the commands that create, modify, and delete change package entries take effect immediately. There is no way to discard such changes, and you do not need to explicitly save them. (You can **Remove** a change package entry altogether, but you can't undo the adding of a new change to an existing entry.)

Change packages do not participate in AccuWork's issue-history capability. Modifications to a change package do not appear in the **Issue History** subtab of an Issue form, although you can use the **Change Package History** feature described above.

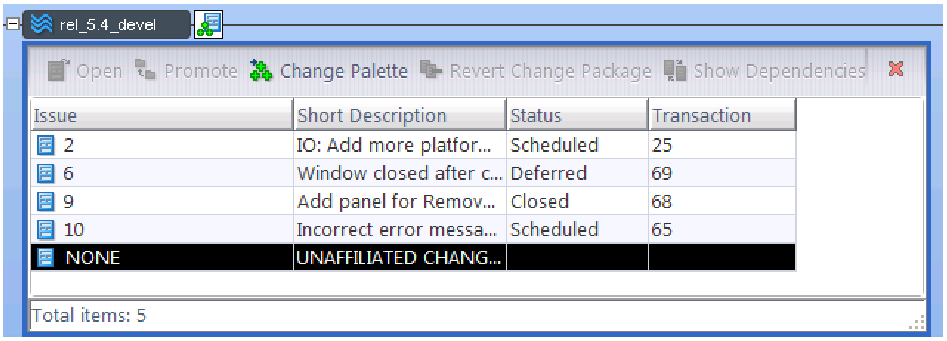
Unaffiliated Changes

It is possible to end up with changes that are not associated with any issue. For example, somebody might decide that a file does not belong to an issue and remove it from the change package.

> **Note:** Whenever you remove an element version from a change package, either from the GUI or from the CLI **cpkremove** command, you should immediately follow the operation with a **Send to Issue** on the removed element, to prevent it showing up as an Unaffiliated Change.

Whenever element versions are not associated with any issue, they are considered *unaffiliated*. Such element versions can lead to confusion, so AccuRev provides a way to view and deal with unaffiliated changes in the GUI.

In the Stream Browser, select **Show Active Issue** mode:



Issue	Short Description	Status	Transaction
2	IO: Add more platfor...	Scheduled	25
6	Window closed after c...	Deferred	69
9	Add panel for Remov...	Closed	68
10	Incorrect error messa...	Scheduled	65
NONE	UNAFFILIATED CHANG...		

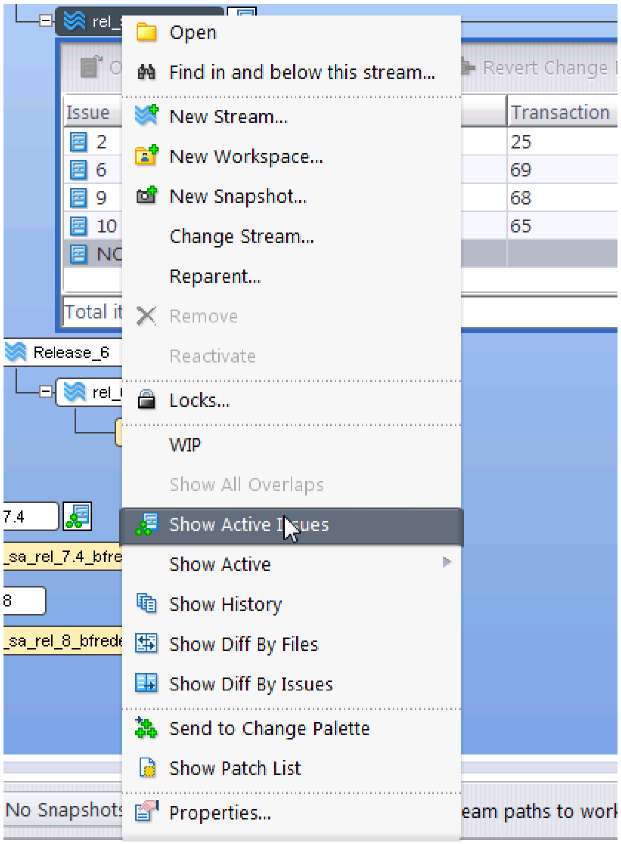
Total items: 5

If any unaffiliated changes exist in the stream, the issue table gains an extra row labeled “NONE”. If your schema is configured to show the Short Description field, “UNAFFILIATED CHANGES” is displayed here. Double-clicking this row displays an error message informing you that, "This issue cannot be opened because it doesn't really exist".

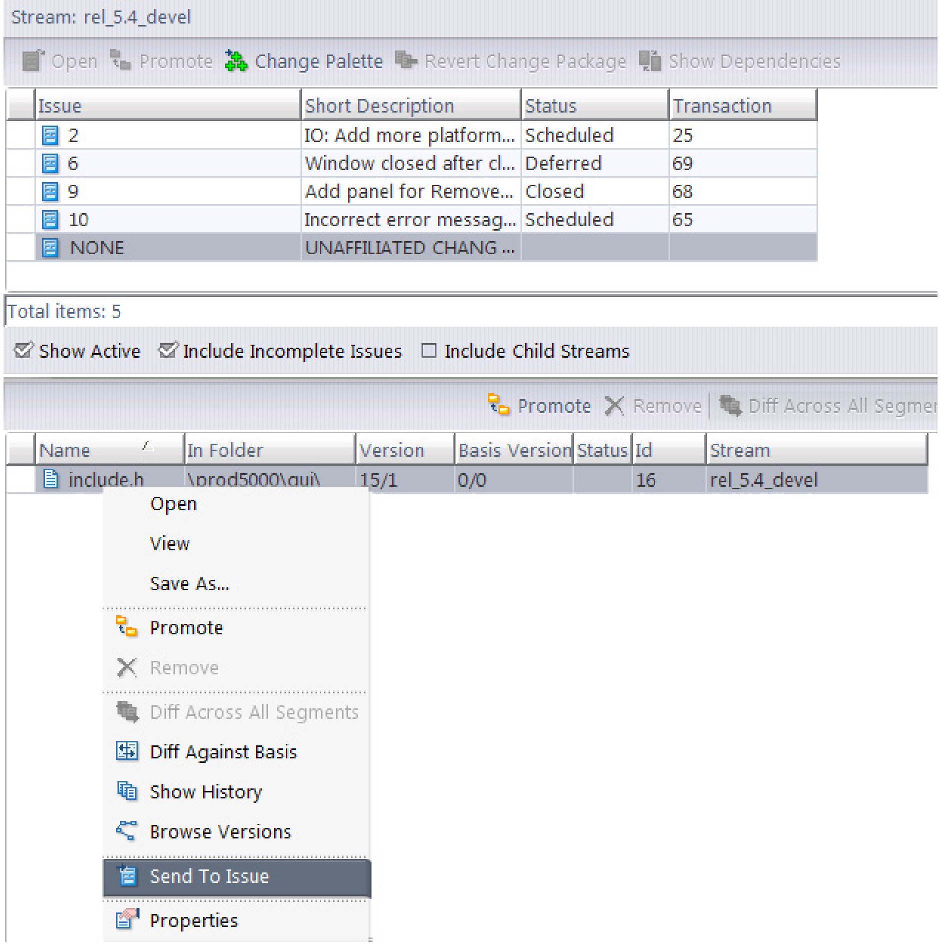
There are two ways to deal with unaffiliated changes: short-term and long-term. If you are working under a deadline and just need a way to deal with these files so that they don't keep you from promoting everything in a stream, you can simply right-click this row and select **Send to Issue**. You then can either assign these unaffiliated changes to a new issue that you create on the fly, or send them to an existing issue if an appropriate one already exists. (An “appropriate” issue might be a catch-all issue that you always use to hold miscellaneous unaffiliated changes. Or, it might be an issue that you know makes logical sense for these particular changes.)

The long-term, more correct solution is to determine why these changes are unaffiliated, and depending on how they got into this state, attempt to correct them. In this case, it would be better to examine the individual element version that is unaffiliated, and determine if different elements require different issues for resolution. For example, if a specific unaffiliated change was caused by removing an element from a change package, the correct fix could be to assign that element to a different more appropriate issue, or to revert that element's changes altogether. (In any case, this brings up a best practice: whenever you remove an element from a change package, you should always immediately follow-up by reassigning that element to another issue, or reverting its changes.)

To examine the unaffiliated versions in detail and deal with them, you should right-click the stream and select **Show Active Issues**.



From the resulting display, you can right-click the unaffiliated elements, explore their history, and then take appropriate actions to correct their unaffiliated status. For example, you can use **Send to Issue** and then promote, or promote the files to an issue while promoting from this stream to its parent.



Stream: rel_5.4_devel

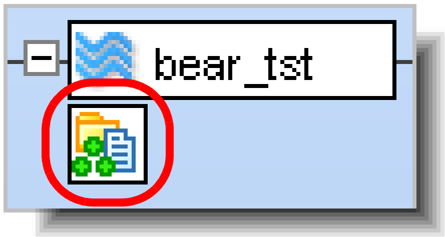
Issue	Short Description	Status	Transaction
2	IO: Add more platform...	Scheduled	25
6	Window closed after cl...	Deferred	69
9	Add panel for Remove...	Closed	68
10	Incorrect error messag...	Scheduled	65
NONE	UNAFFILIATED CHANG ...		

Total items: 5

☑ Show Active ☑ Include Incomplete Issues ☐ Include Child Streams

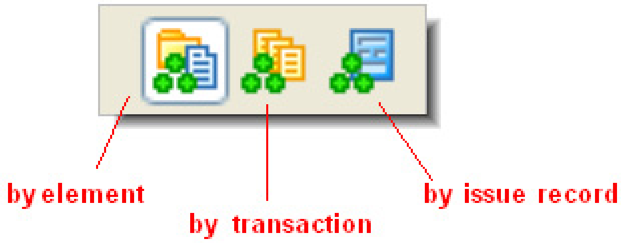
Name	In Folder	Version	Basis Version	Status	Id	Stream
include.h	\prod5000\gui\	15/1	0/0		16	rel_5.4_devel

Promote-by-Issue Basics

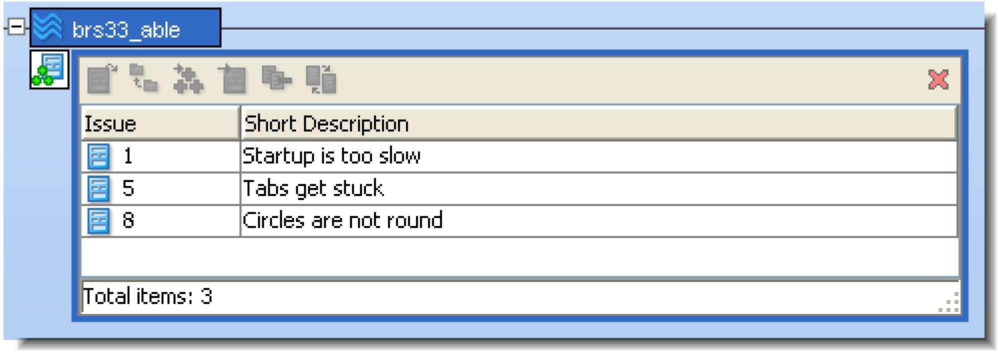


The Stream Browser can show the development activity currently taking place in each stream or workspace. A control below or to the right of the stream or workspace opens or closes a subwindow that displays the details of the development activity. The activity details can appear in several ways — by element, by transaction, or by issue record.

Use the development-activity mode controls at the right side of the Stream Browser toolbar to determine how the activity details will be displayed. (The icons on the controls below the streams and workspace change accordingly.) You can change modes either before or after opening a development-activity subwindow.



When displaying a stream's activity by issue record, the subwindow displays the issue records that are in a particular stream:



brs33_able

Issue	Short Description
1	Startup is too slow
5	Tabs get stuck
8	Circles are not round

Total items: 3

(This display is the same as the results of **Show Active Issues** or **Stream Diff by Issues**. In this note, all the examples use the Stream Browser's development activity subwindow, but you can perform promote-by-issue operations in these other contexts, also.)

Promote-by-issue involves selecting one or more issue records and performing any of the following operations:

- Invoking **Promote** from the subwindow's toolbar or the context menu of the issue(s).
- Drag-and-dropping the issue record(s) onto another stream or workspace in the Stream Browser display. The destination can be the parent (backing) stream, another dynamic stream, or one of your workspaces. This invokes a **Promote** or **Send to Workspace** operation on all the elements in the selected issue record(s).

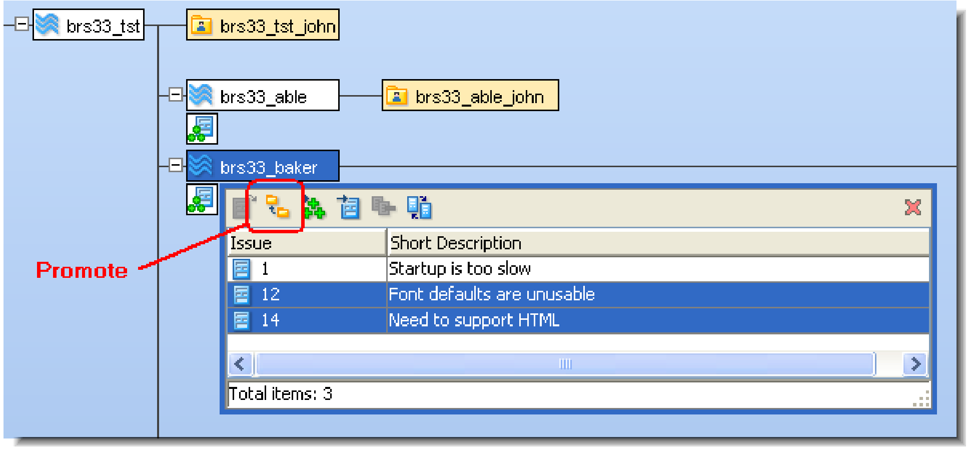
You can also drag-and-drop the subwindow control to another stream. This performs the **Promote** or **Send to Workspace** operation on all the issue records in the subwindow (even if only some of them are currently selected).

The following sections explore several promote-by-issue scenarios.

> **Note:** in the remainder of this technical note, we shorten the term "issue record" to "issue".

Promoting Issues to the Parent Stream

In many cases, promote-by-issue is simple and easy. You promote one or more issues to the parent stream:

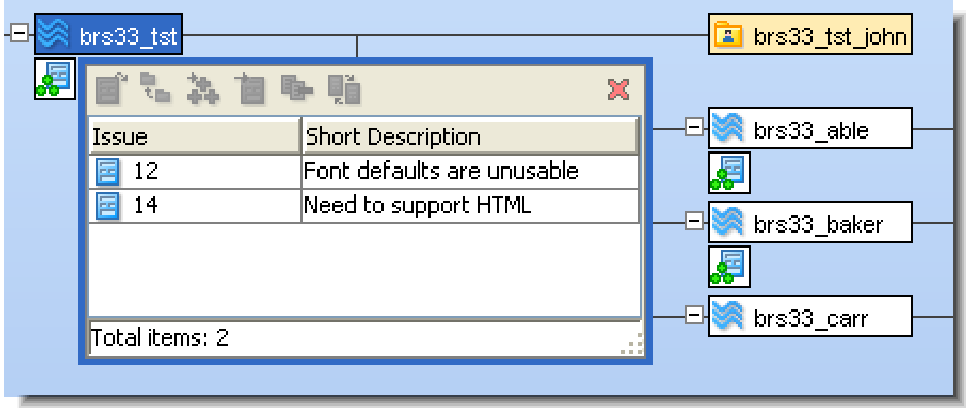


Promote

Issue	Short Description
1	Startup is too slow
12	Font defaults are unusable
14	Need to support HTML

Total items: 3

And the issues simply move into the parent stream:



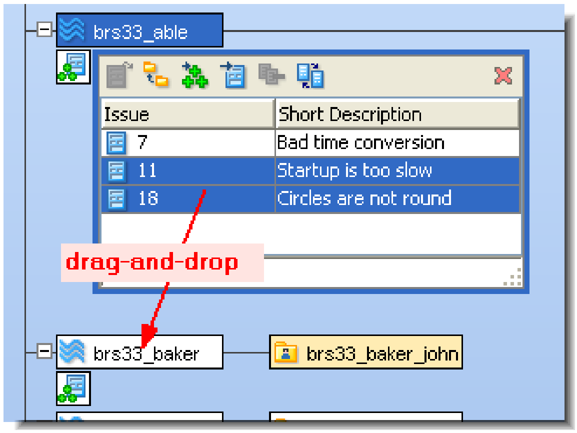
Issue	Short Description
12	Font defaults are unusable
14	Need to support HTML

Total items: 2

Note: if **Promote** fails with a “merge required” error, you must perform a merge for one or more elements in the source workspace (or a workspace below the source stream). When promoting the merged version(s), assign them to the same issues as the original version(s).

Cross-Promoting Issues to a Non-Parent Stream — Simple Case

Similarly, in many cases cross-promoting from one dynamic stream to another dynamic stream proceeds without complication. Perform a drag-and-drop operation:



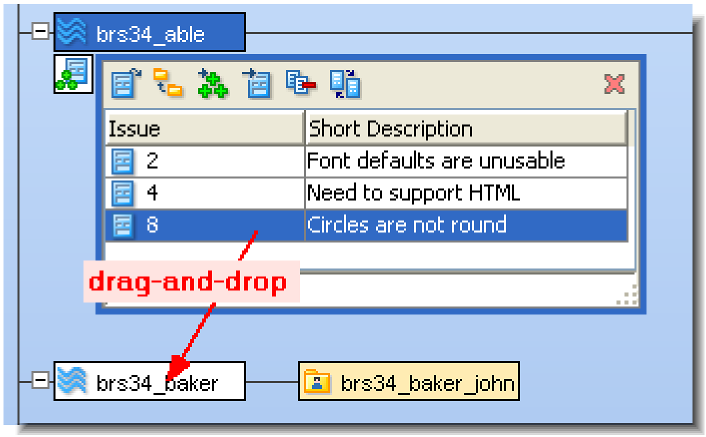
Issue	Short Description
7	Bad time conversion
11	Startup is too slow
18	Circles are not round

Assuming that all the elements in the issues’ change packages can be propagated to the destination stream without merging or patching, the operation completes automatically.

Cross-Promoting Issues to a Non-Parent Stream — Patch Required

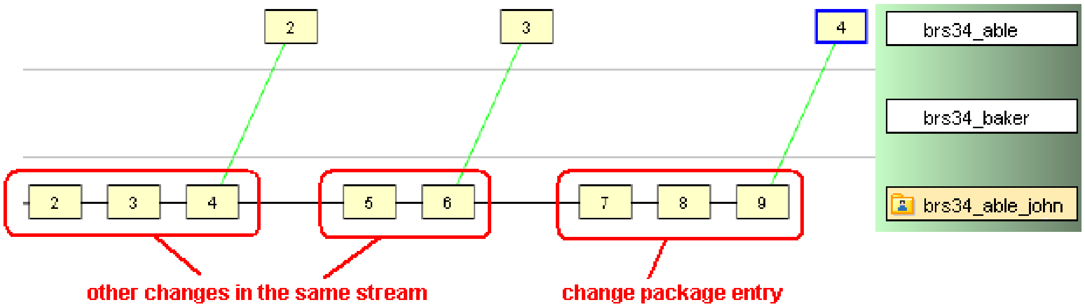
In this scenario, your intention is the same as in the preceding one — to cross-promote the versions in one or more issues from one dynamic stream to another. Before proceeding, please see the definition of the following terms in the *AccuRev Glossary* chapter of the *AccuRev Concepts Manual*: *direct*, *indirect*, *incomplete*, *cyclical*, and *coalesce*.

You start the same way, with a drag-and-drop operation from stream **brs34_able** to stream **brs34_baker**:



Issue	Short Description
2	Font defaults are unusable
4	Need to support HTML
8	Circles are not round

But in this scenario, one or more of the elements requires a patch operation before it can be promoted. In this case, issue(s) are loaded into the Change Palette. This occurs when the element’s change-package entry does not contain the complete set of changes to the element that (1) are in the source stream and (2) have not yet been promoted to the destination stream:



This scenario is covered in *Working with the Issue Change Palette*, *Issue Mode* on page 262.

Incomplete Change Packages

AccuRev change packages provide a powerful tool for dealing with a set of element versions as a group. However, if you do not institute and follow some best practice policies, you can encounter error messages warning of missing versions when you promote by issue. This section describes the three most common causes of these warnings, how to correct them, and how to avoid them in the future.

Overview

The formal definition of an incomplete change package is "Some of the versions of elements associated with the change package are not part of the current stream."

But what does this really mean? How did this happen? And how do you fix it? Better yet, how do you avoid getting into this situation in the first place?

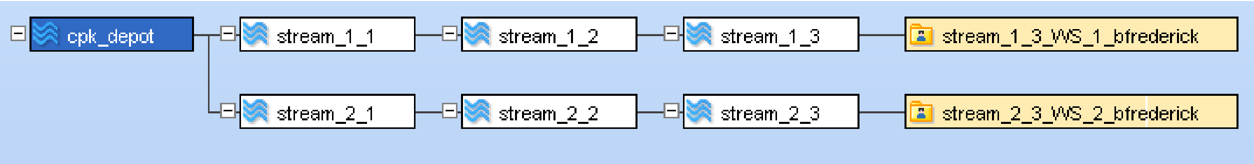
There are three typical scenarios that result in incomplete change packages:

- **Purge** ("**Revert to Basis**" in the GUI) operations on individual files.
- Reuse of issues across multiple streams.
- **Promote by File** (instead of **Promote by Issue**) operations.

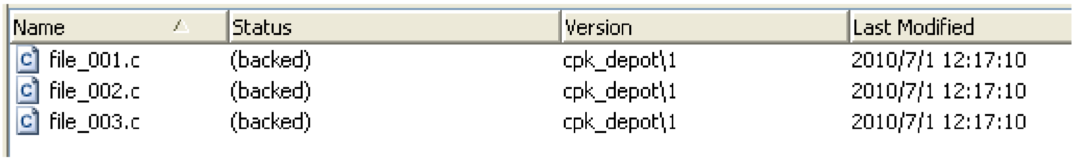
All three of these result in the same situation: one or more of the files in your change package will not have the correct version in the current stream.

An Example Scenario

Here is a very simple stream environment that we will use to illustrate a typical incomplete change package situation:

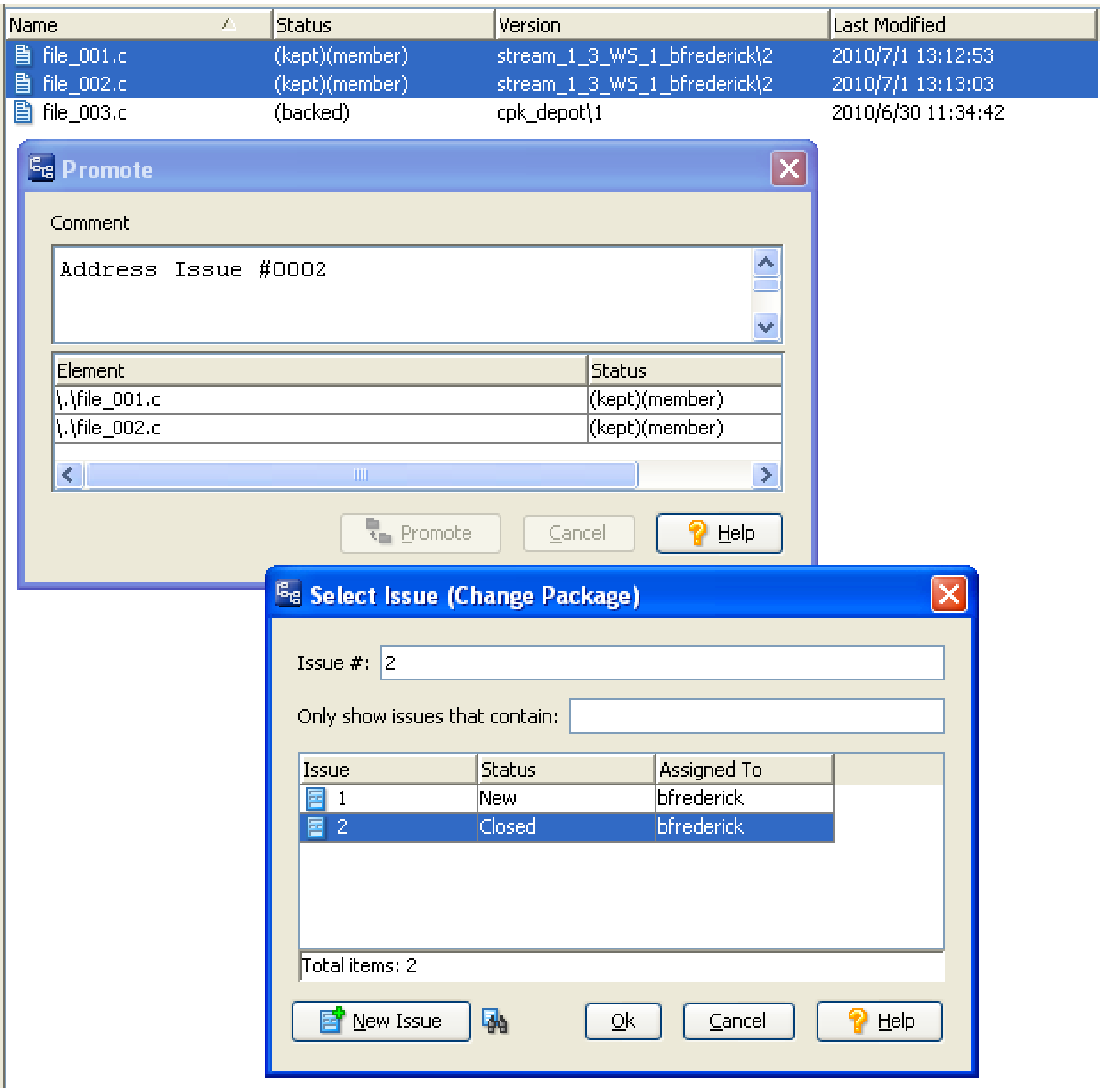


Assume that workspace stream_1_3_WS_1 contains three source code files:



Name	Status	Version	Last Modified
file_001.c	(backed)	cpk_depot\1	2010/7/1 12:17:10
file_002.c	(backed)	cpk_depot\1	2010/7/1 12:17:10
file_003.c	(backed)	cpk_depot\1	2010/7/1 12:17:10

You modify file_001.c and file_002.c to address issue #0002, and promote them from your workspace to the parent stream (stream_1_3).



Name	Status	Version	Last Modified
file_001.c	(kept)(member)	stream_1_3_WS_1_bfrederick\2	2010/7/1 13:12:53
file_002.c	(kept)(member)	stream_1_3_WS_1_bfrederick\2	2010/7/1 13:13:03
file_003.c	(backed)	cpk_depot\1	2010/6/30 11:34:42

Promote

Comment

Address Issue #0002

Element	Status
\.\file_001.c	(kept)(member)
\.\file_002.c	(kept)(member)

Select Issue (Change Package)

Issue #: 2

Only show issues that contain:

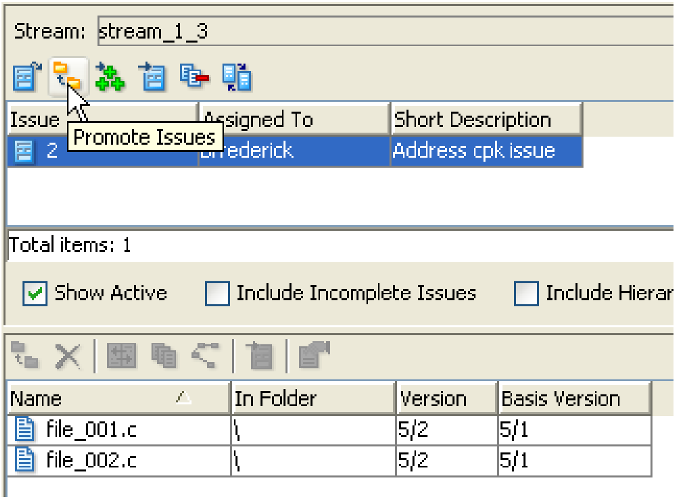
Issue	Status	Assigned To
1	New	bfrederick
2	Closed	bfrederick

Total items: 2

Note: You must have the change package integration enabled in the Schema Editor to be prompted for an issue when you promote from a workspace. If you don't see the Select Issue (Change Package) dialog when you promote, you need to enable the integration. See "Change-Package-Level Integration" in the *AccuRev Administrator's Guide* for details.

In addition, if you have implemented support for a third-party issue tracking system (ITS), the Select Issue (Change Package) dialog displays controls that let you choose whether to submit changes against the AccuWork issue number or the third-party ITS key. See *Using Third-Party Keys in the AccuRev GUI* on page 408 for more information.

At this point, if you go to the Stream Browser and select **Show Active Issues** by right-clicking over stream 1_3, you will see issue #2 and the files that you just promoted. Make sure that **Include Incomplete Issues** is not checked to confirm that this is a complete change package. If you were ready to continue promoting this package up the stream hierarchy, you could do so by selecting issue #2 and clicking the **Promote** icon ( ).



Stream: stream_1_3

Issue	Assigned To	Short Description
2	...ederick	Address cpk issue

Total items: 1

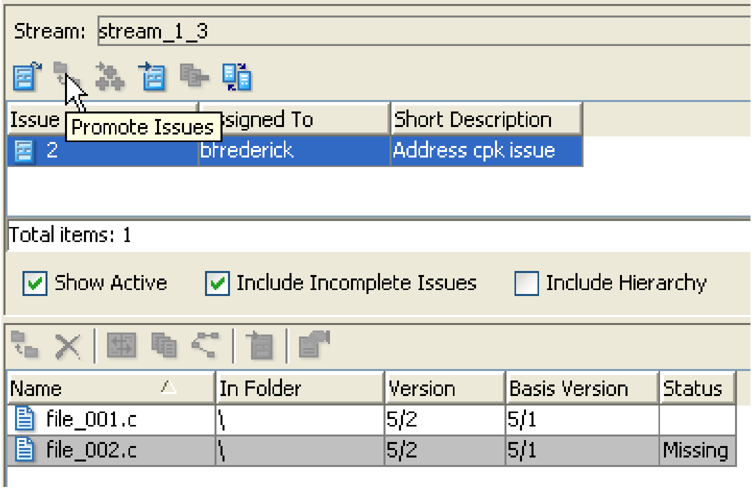
☑ Show Active ☐ Include Incomplete Issues ☐ Include Hierar

Name	In Folder	Version	Basis Version
file_001.c	\	5/2	5/1
file_002.c	\	5/2	5/1

This is the correct way to promote change package files higher up the stream hierarchy.

But what if somebody had performed one of the three operations listed above, which caused, for example, only the backed version of file_002.c to appear in stream_1_3?

This first thing you might notice is that the issue does not appear unless the **Include Incomplete Issues** checkbox is enabled. And when it is, the **Promote** icon is grayed out when you select issue #2, and file_002.c appears with a gray band.



Stream: stream_1_3

Issue	Assigned To	Short Description
2	bfrederick	Address cpk issue

Total items: 1

☑ Show Active ☑ Include Incomplete Issues ☐ Include Hierarchy

Name	In Folder	Version	Basis Version	Status
file_001.c	\	5/2	5/1	
file_002.c	\	5/2	5/1	Missing

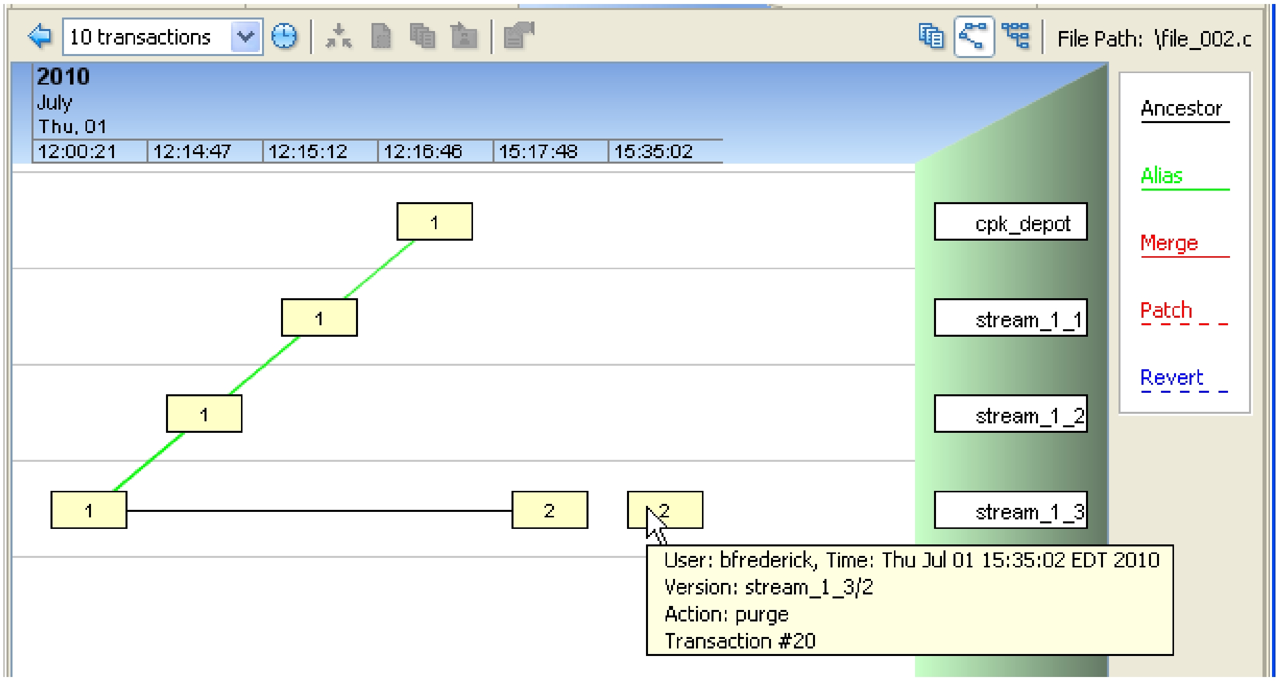
For steps on how to troubleshoot and identify the cause of your incomplete change package, see *How to Troubleshoot Incomplete Change Packages* at the end of this section. But first, here are details about each of the three main scenarios.

File Purge (Revert to Basis)

Scenario:

You fix an issue and then promote the affected files from your workspace to an integration stream. For some reason, someone decides to perform a purge (**Revert to Basis)** on one of the files. When it's time to promote the change package from the integration stream to its parent stream, you discover that the change package is incomplete and the **Promote** icon is grayed out. This is, in fact, exactly what happened in *An Example Scenario* above.

You can tell that this is the case by right-clicking on the “missing” file (file_002.c) and selecting **Browse Versions**.



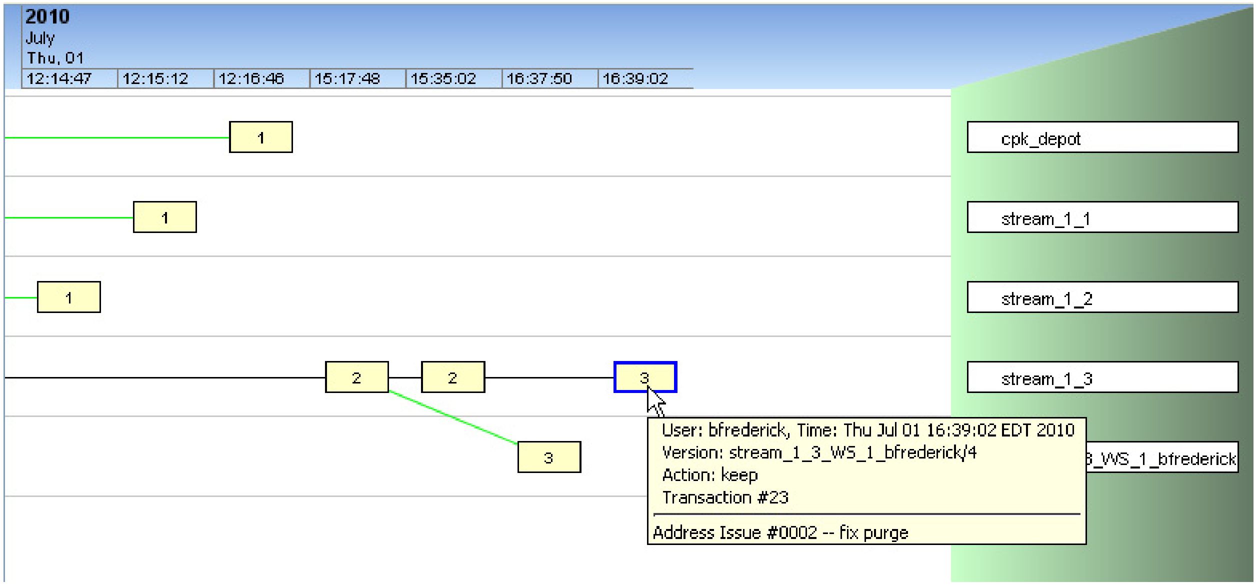
User: bfrederick, Time: Thu Jul 01 15:35:02 EDT 2010
Version: stream_1_3/2
Action: purge
Transaction #20

The second, disconnected box for version 2 is the tip-off that the file was purged, and the tooltip message confirms the action.

How do you fix it?:

If you determine that the purge was a good idea, and that the change package doesn't require the purged file, your simplest fix is to remove it: go to the Changes tab for the issue, select the file and then click **Remove**. Once you have done this, the issue will no longer appear when **Include Incomplete Issues** is checked, and you will be able to promote the change package further up the stream hierarchy.

However, if it turns out that the purge was a mistake, you'll need to reverse it. Because AccuRev is TimeSafe, you can’t just make the purge disappear like it never happened. But what you can do is send that version of the file to your workspace (right-click the version in the Version Browser and select **Send To Workspace** and re-promote it. When the file becomes active in your workspace, it will not have the same version number—in this example it will now be version “3” instead of the original version “2”, and will become version “4” when you perform a **Keep** on it. After you promote it, the change package in the parent stream will again be complete, and the version browser will no longer show the purged version as a disconnected box.

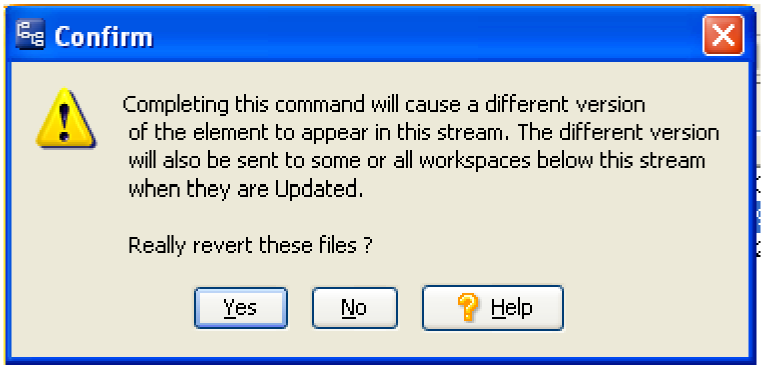


User: bfrederick, Time: Thu Jul 01 16:39:02 EDT 2010
Version: stream_1_3_WS_1_bfrederick/4
Action: keep
Transaction #23

Address Issue #0002 -- fix purge

How do you prevent this from happening in the future?:

If you need to purge a file, first make sure that it is not part of a change package. Pay attention to any error message that tries to warn you against making a mistake.



Completing this command will cause a different version of the element to appear in this stream. The different version will also be sent to some or all workspaces below this stream when they are Updated.

Really revert these files ?

You typically want to click **No** in response to such a warning.

If the file is a part of a change package, and you still want to purge the file, then perform a **Revert by Change Package** instead: in the Stream Browser, right-click the stream from which you wish to revert the change package and click **Show History**. Then select the promotion operation you wish to reverse, and click the Revert icon ( ). Make the adjustments you need in your workspace, and then re-promote the modified change package.

> **Note:** As of AccuRev Release 5.2, you now have two ways to **Revert by Change Package**:
>
> - **Revert Change Package Using Workspace**
> - **Revert Change Package Directly in Stream**
>
> **Revert Change Package Using Workspace** gives you the opportunity to test your changes before promoting them into a stream. **Revert Change Package Directly in Stream** puts the results of the

revert into the stream immediately, so should be used for simpler reverts, or in streams that do not propagate untested changes to other users.

You can also prevent non-administrators from purging files in streams by customizing the server_preop_trig trigger. See *How to Troubleshoot Incomplete Change Packages* below for more information.

Reuse of Issues Across Streams

Scenario:

In the previous sections, you worked on issue #0002 in workspace stream_1_3_WS_1 and promoted the following files against it:

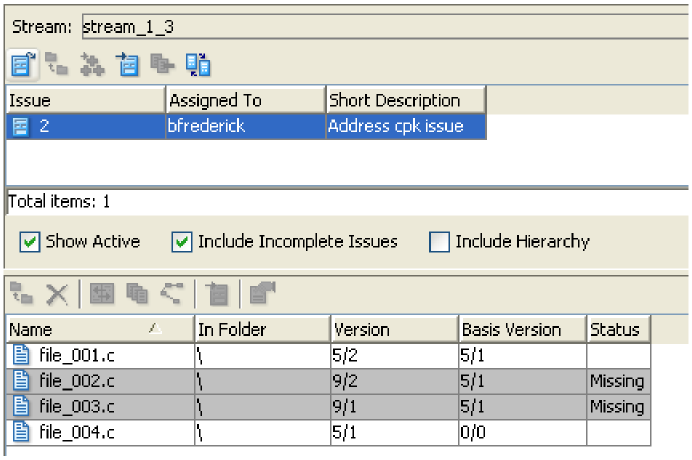
- file_001.c
- file_002.c

But what if this issue also applied to a different release being handled in a different stream (in this case, the hierarchy under stream_2_1)? In that stream, the fix needs to be made in a different set of files:

- file_002.c
- file_003.c

You perform the changes to these files in workspace stream_2_3_WS_2. Even worse, somebody else continues to work on issue #2 in the original stream and adds a new file named file_004.c which doesn't even exist in Stream 2. Both developers promote their changes against the same issue, one from workspace stream_1_3_WS_1 and the other from stream_2_3_WS_2.

Now the change package is incomplete in both streams: the modified versions of file_002.c and file_003.c are not available under Stream 1, nor does the new file_004.c exist under Stream 2. If you do a Show Active Issues in both of the parent streams, you see two different incomplete change packages, each missing different files.



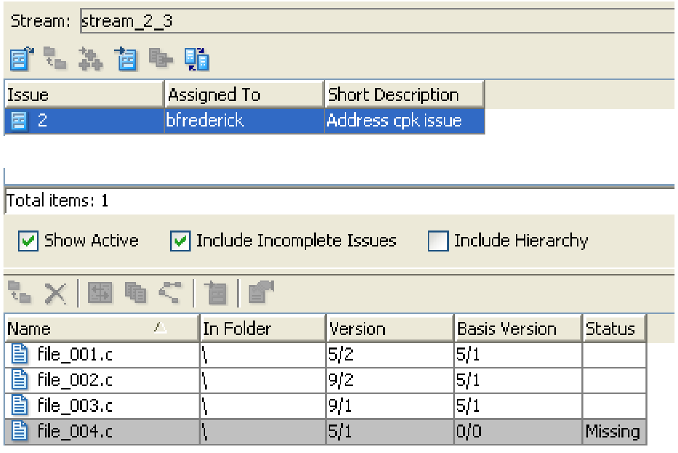
Stream: stream_1_3

Issue	Assigned To	Short Description
2	bfrederick	Address cpk issue

Total items: 1

☑ Show Active ☑ Include Incomplete Issues □ Include Hierarchy

Name	In Folder	Version	Basis Version	Status
file_001.c	\	5/2	5/1	
file_002.c	\	9/2	5/1	Missing
file_003.c	\	9/1	5/1	Missing
file_004.c	\	5/1	0/0	



Stream: stream_2_3

Issue	Assigned To	Short Description
2	bfrederick	Address cpk issue

Total items: 1

☑ Show Active ☑ Include Incomplete Issues □ Include Hierarchy

Name	In Folder	Version	Basis Version	Status
file_001.c	\	5/2	5/1	
file_002.c	\	9/2	5/1	
file_003.c	\	9/1	5/1	
file_004.c	\	5/1	0/0	Missing

How do you fix it?:

Don't try to cross-promote or merge the missing files. Instead:

1. Create a new issue for Stream 2.
2. Open the original issue and select the files which were added when promoted into Stream 2.
3. Right click and select **Send to Issue**.

4. Select the new issue created in **Step 1** to send the selected file(s) to.

You will now see the original issue as complete in Stream 1, and the new issue as complete in Stream 2.

How do you prevent this from happening in the future?:

Make it a policy to not use the same issue to address problems in multiple streams. If you need to fix the same issue in another stream, create a new issue for that stream and promote files against that issue.

Promoting by File Instead of by Issue

Scenario:

Similar to the previous examples, you work on issue #0002 in workspace stream_1_3_WS_1 and promote the following files against it:

- file_001.c
- file_002.c

Later, in the backing stream(stream_1_3), somebody promotes file_002.c (but not file_001.c) to its parent stream (stream_1_2). This leaves the change package for issue 2 complete in stream_1_3, but now it appears as an incomplete change package in stream_1_2.

How do you fix it?

This one is easy: simply promote the entire change package from stream_1_3 to its parent stream_1_2. Since file_002.c has already been promoted, it is ignored, but file_001.c gets promoted, making the change package complete.

How do you prevent this from happening in the future?:

Make it a policy to always promote by issue. Avoid promoting by file. You can enforce this policy by customizing the server_preop_trig trigger to disable promotion by file and only allow promotions to occur via issue. See *Sample server_preop_trig rules for Change Packages* on page 92 of the AccuRev *Administrator's Guide* for more information.

How to Troubleshoot Incomplete Change Packages

Use the following steps to identify incomplete change packages and their causes. Then you can use the information above to fix the situation and avoid it in the future.

1. To see if an incomplete change package exists in a stream, go to the Stream Browser, right-click over the stream, and select **Show Active Issues.**
2. Make sure that **Include Incomplete Issues** is checked and look for any grayed-out issues.
3. If you see a grayed-out issue, select it. This displays the change package files in the lower pane. At least one of these will be grayed-out and have a status of "missing".
4. Make note of the values in the "Version" and "Basis Version" columns. The first number (before the slash) identifies the stream and the second number (after the slash) identifies the file version that should exist in that stream.
5. To identify the stream names for these numbers, go to the Stream Browser and display the tabular mode. Look for the number under the column labeled "#", and identify its name under the "Name" column.
6. To figure out why the file is missing, use the Version Browser:

- If the version is simply missing from this stream, but exists in the child stream, then it is likely that one or more of the files from the change package were promoted individually (rather than by issue) and this file was not. (See *Promoting by File Instead of by Issue* above.)
- If the version shows a disconnected box in the Version Browser, the problem is due to a purged file. (See *File Purge (Revert to Basis)* above.)
- If the version is missing from this stream, and doesn't exist in the child stream but appears to exist in a different stream hierarchy, it is likely that the same issue was used to fix a problem in two different streams. (See *Reuse of Issues Across Streams* above.)

Note: There is one additional incomplete change package scenario that you may encounter: if the issue contains only one file, the issue may not appear as incomplete when it should. You need to go to the **Show Active Issues** display and ensure that the **Include Hierarchy** checkbox is enabled for this situation to be displayed.

Demote by Issue

The concept of demoting changes was discussed in *Demote: Moving Elements Out of a Stream* on page 99 where it was noted that AccuRev supports the ability to demote by file, by transaction, and by issue. If you promote by issue, you have the option to demote by issue if you want to back out specific sets of changes.

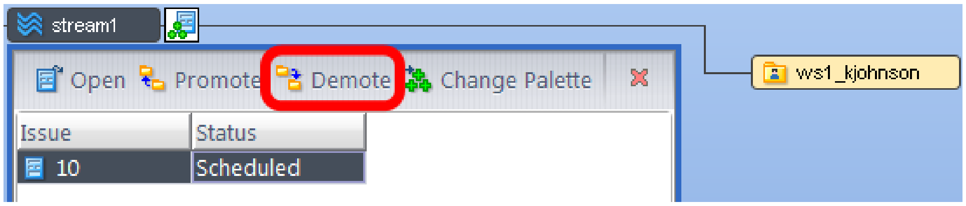
When you demote by issue, you move the associated changes from a parent stream to a child stream or workspace. In this way, you can, in effect, 'undo' the changes that were promoted to the parent while retaining these changes in a child stream or workspace.

Say, for example, that changes associated with issue 10 introduced a defect that broke your build and you would like to move these changes out of your build stream. You can simply demote issue 10 to a child stream or workspace, fix the defect, and then promote issue 10 back into the parent stream once more.

Demoting Issues to a Child Stream or Workspace

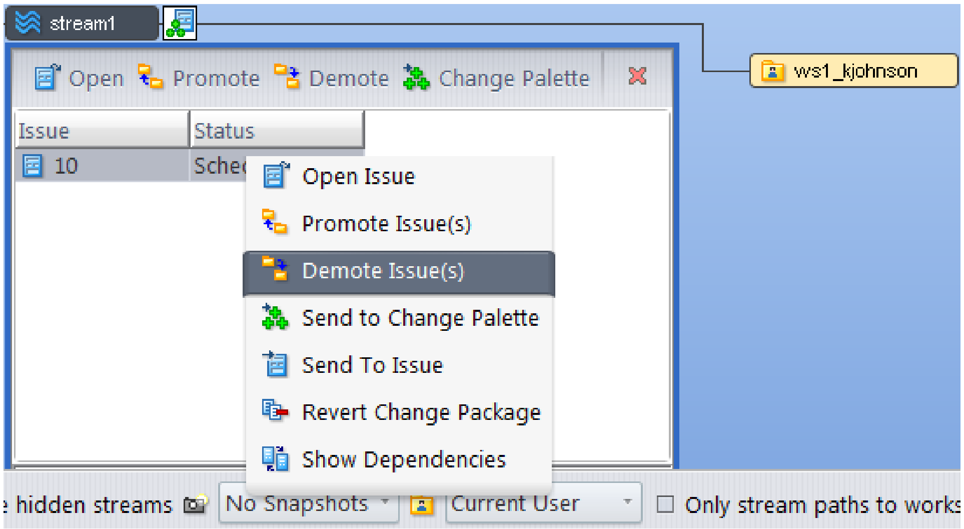
This following section describes the process of demoting by issue through the StreamBrowser. You can also demote by issue through the Command Line Interface. See the description of *demote* for more information.

In the StreamBrowser, select the issue(s) you want to demote. You can either click the **Demote** button:

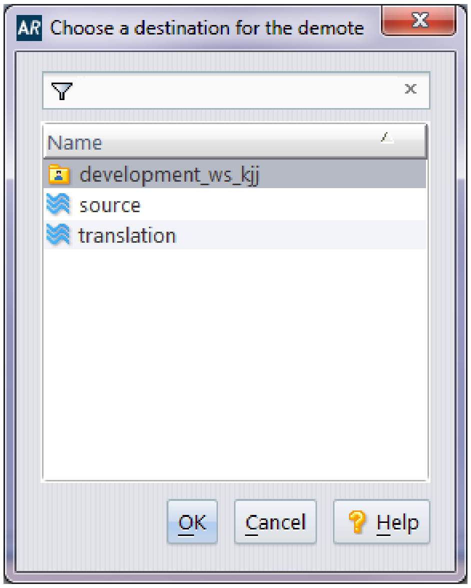


Issue	Status
10	Scheduled

or right-click the issue(s) and select **Demote Issue(s)** from the context menu:

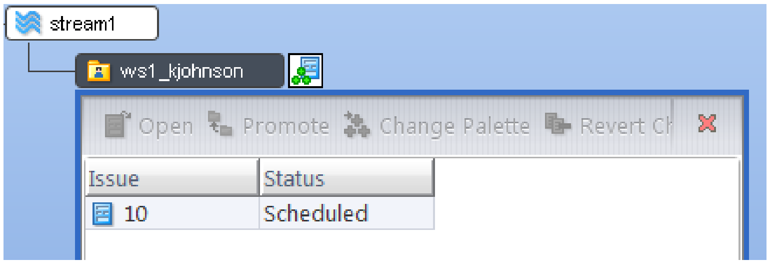


If more than one child stream or workspace exists for the current stream, the Choose a destination for the demote dialog box appears:



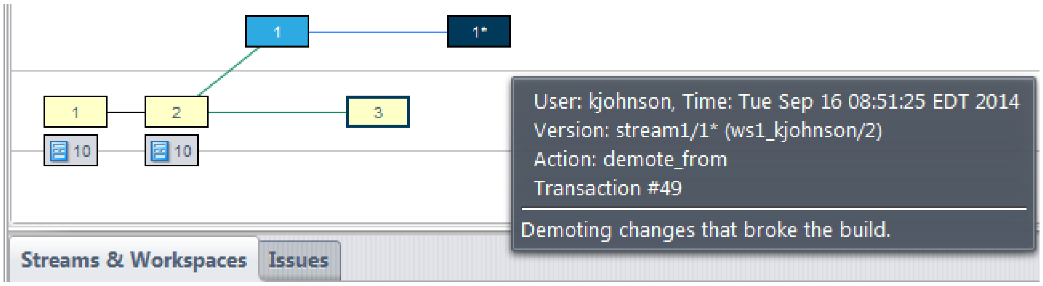
After you select a destination stream or workspace and click **OK**, the changes associated with the issue(s) are no longer present in the parent stream but are now active in the default group of the destination stream or workspace you specified.

The StreamBrowser shows the fact that issue 10 now exists in the child workspace:



Issue	Status
10	Scheduled

If you right-click a file in issue 10, select **History** > **Browse Versions** from the context menu, and hover over the demoted version, the tool tip confirms the demotion:



User: kjohnson, Time: Tue Sep 16 08:51:25 EDT 2014
Version: stream1/1* (ws1_kjohnson/2)
Action: demote_from
Transaction #49

Demoting changes that broke the build.

Note that you cannot "cross demote" an issue and neither can you demote issues by using drag and drop.

A demote by issue operation may fail under certain circumstances, including the following conditions:

- An overlap status exists for one of the files selected for demote in the destination stream or workspace
- The issues to be demoted contain dependencies that are not selected for demote. All dependencies must be included in the demote operation or it will fail.
- A file in the issue contains non-contiguous segments, which is a condition that arises when the file has been promoted against, say, issue 1, then issue 3, then issue 1 again.

Common CPK Messages and Displays

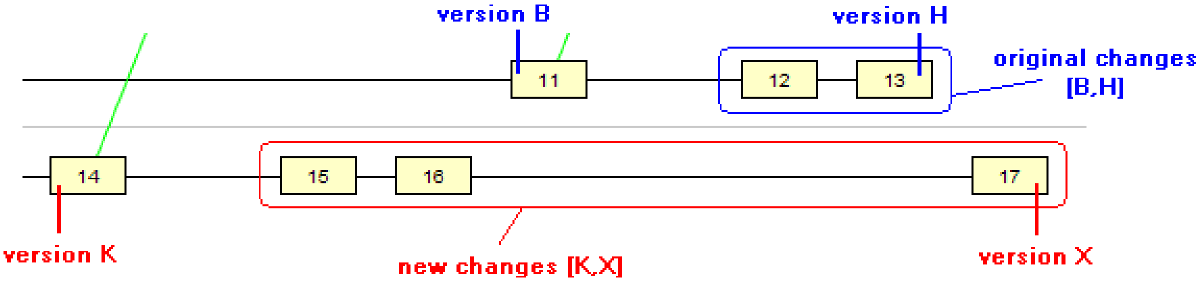
When you promote issues with change packages, you sometimes encounter displays and messages that alert you to common situations, but it is not always obvious what is causing the alert or how to address it. This section describes some common scenarios and how to deal with them.

"Promote failed: Some element versions in specified issues cannot be coalesced."

You will sometimes see this when promoting an issue from one stream to its parent. All this means is that some versions of an element in this issue's change package were promoted against a different issue. This creates a *dependency* on the other issue.

The Issues Conflict Message Box

This error box appears when there is no way to produce a valid change package entry for an element by combining your new set of changes with the existing change package entry. The requirement that the basis version must be an ancestor of the head version cannot be fulfilled.



You can remedy this situation by performing a *merge* at the element level. (There is no merge operation defined for change package entries.) In the example above, merging version H and version X would create a new version; a change package entry with the new version as its head can be combined with the existing entry.

Promote Coalesce Error

If a promote-by-issue operation (a standard child-to-parent promote, not a cross-promote) involves multiple issues whose change packages include the same element, AccuRev attempts to combine or *coalesce* those entries into a single, valid change package entry. If the element's change package entries cannot be coalesced, the promote operation fails.

If this occurs, please follow the steps described in *Cross-Promoting Issues to a Non-Parent Stream — Patch Required* on page 343.

Send to Issue : Associating Changes with an Issue

The ***Send to Issue*** command creates or modifies change package entries for one or more elements in one or more issue records. When first created, a change package entry has the same structure as a patch: the set of 'recent changes' to an element. After subsequent modifications, a change package entry can include a much broader set of changes.

The command has several variations, providing different ways to specify the data to be sent to the change package(s):

- You specify a version, which AccuRev interprets as the head version of a patch. AccuRev automatically determines the corresponding basis version.
- You specify both the head version and basis version of a patch.
- You specify an existing change package entry, to be sent to one or more additional change packages.

Invoking the Send to Issue Command

You can invoke ***Send to Issue*** in the following contexts. Note that each context provides access to some, but not all, variations of the command.

- In the ***Version Browser***, choose ***Send to Issue*** or ***Send to Issue (specifying basis)*** from the context menu of any version.
- In the Versions pane of the ***History Browser***, select one or more versions and choose ***Send to Issue*** from the selection's context menu.
- In the ***Stream Version Browser***, choose ***Send to Issue*** from the context menu of any version.
- In the ***Changes*** subtab of an issue record or the Change Package Contents pane of a ***Stream Issues*** tab, select one or more change package entries and choose ***Send to Issue*** from the selection's context menu.

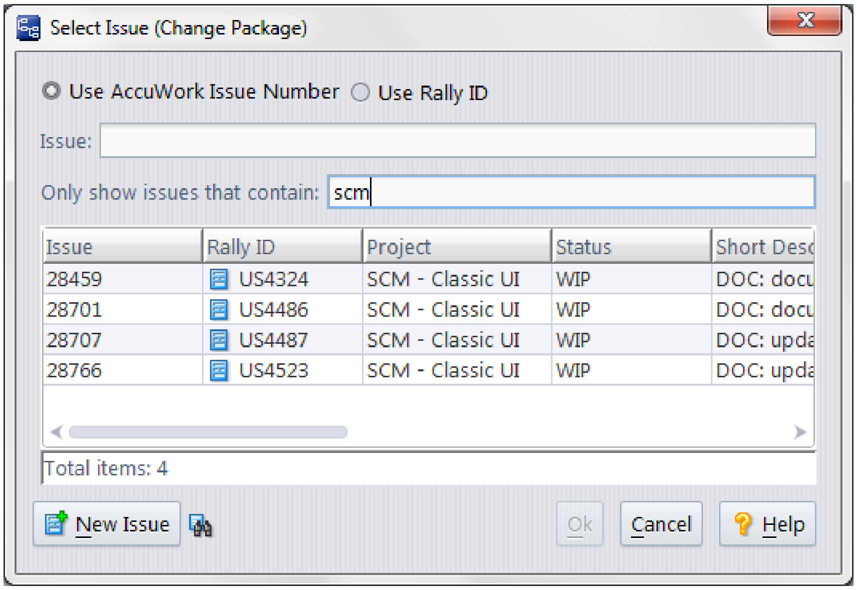
Specifying the Basis Version of the Change

In the ***Send to Issue (specifying basis)*** variant of the command, you specify the basis version, defining the extent of the change to be sent to the change package(s). After you invoke the command, the mouse pointer changes to . Click the version you want to be the basis version for the change. (You can press the **Esc** key at this point to cancel the command.)

The basis version you specify must be an *ancestor* of the head version. If it isn't, an "Invalid Change Package" error occurs.

Using the Select Issue (Change Package) Dialog

AccuRev prompts you to specify one or more issue records, in which change package entries can be created or modified.



Select Issue (Change Package)

● Use AccuWork Issue Number ○ Use Rally ID

Issue:

Only show issues that contain: scm

Issue	Rally ID	Project	Status	Short Desc
28459	US4324	SCM - Classic UI	WIP	DOC: docu
28701	US4486	SCM - Classic UI	WIP	DOC: docu
28707	US4487	SCM - Classic UI	WIP	DOC: upda
28766	US4523	SCM - Classic UI	WIP	DOC: upda

Total items: 4

(This is the same dialog box you see when you use **Promote**.)

If you've specified one of your AccuWork queries as the default query, AccuRev executes the query and displays the results.

> **Note:** If you have enabled the use of third-party issue tracking system (ITS) keys in the AccuWork schema, the ***Select Issue (Change Package)*** dialog displays additional controls that let you specify whether you want to use AccuWork issue numbers or third-party keys when selecting issues to promote against. See *Using Third-Party Keys in the AccuRev GUI* on page 408 for more information.

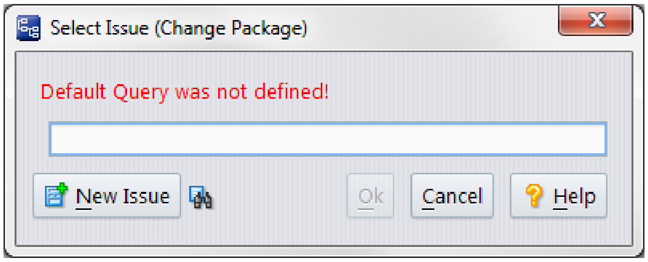
You can select one or more of the issue records:

- By clicking: left-click to select an issue record. Hold down the **Ctrl** or **Shift** key to select multiple records.
- By typing: enter one or more numbers, separated by SPACEs.
- By filtering: enter text into the **Only show issues that contain** field, then click to choose an issue from the filtered list. Delete the text in that field to remove the filter and display all issues.

You also have two other options for specifying issue records other than the initial set shown by the dialog:

- Choose an issue record associated with another user. To do so, choose a user or group name from the drop-down list to the right of the ***Issue #*** field. The set of names shown here is defined when the change-package configuration is set (see *Change Packages Tab* on page 398).
- Create a new issue using the ***New Issue*** button at the bottom of the dialog. Note that if you have chosen to display issue records associated with another user, the new issue must be associated with that user when you create it, or the issue will not display in the Send to Issue dialog.

If you don't have a default query, AccuRev prompts you to type in an issue number (or multiple numbers, separated by SPACEs).



Default Query was not defined!

(You can set a default query by clicking **View Issue Queries**, right-clicking an appropriate entry in the resulting display, and selecting **Select as Default** from the context menu.)

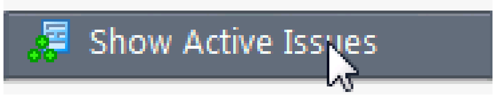
The Active Issues Tab (Show Active Issues)

The **Active Issues** tab (also referred to as "**Stream Issues**") displays a table in the upper pane showing the issue records that are in a particular stream. The lower pane displays the entries in a selected issue record's change package.

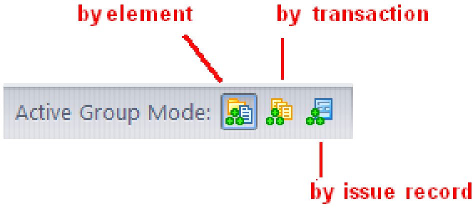
> **Note:** Strictly speaking, issues are not "in" a stream -- the only objects that are really in a stream are element versions. And since you can describe a set of versions as being in a stream, it makes sense to describe the set of versions in a change package as being in a stream. From there, it is a short leap to describe the AccuWork issue record containing the change package as being "in" a stream.

Opening the Tab

To open an **Active Issues** tab, showing the contents of a stream in terms of change packages: choose **Show Active Issues** from the context menu of any stream, or snapshot, or workspace in the Stream Browser, or from the **Actions** menu in the Stream Browser toolbar.

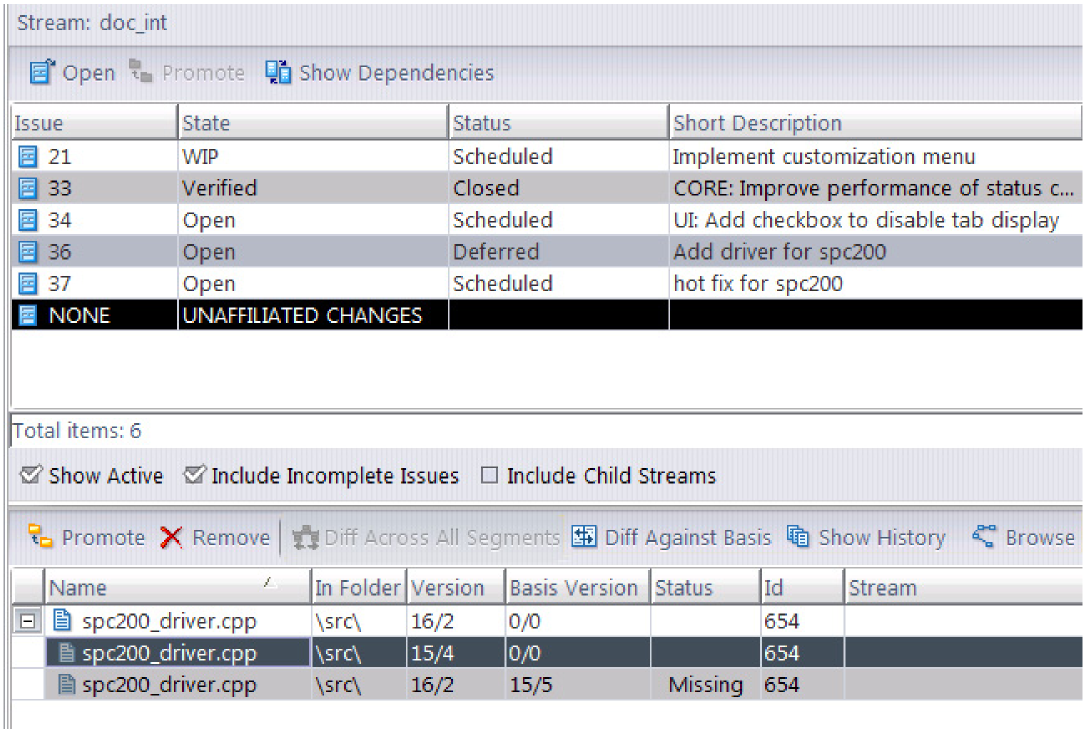


Note: You can display a subset of the **Show Active Issues** tab by clicking the icon under or next to the stream box. You display this icon by clicking the issue record icon on the Stream Browser toolbar..



Stream Issues Tab Layout

The Stream Issues tab includes two panes, each with its own toolbar:



Stream: doc_int

Issue	State	Status	Short Description
21	WIP	Scheduled	Implement customization menu
33	Verified	Closed	CORE: Improve performance of status c...
34	Open	Scheduled	UI: Add checkbox to disable tab display
36	Open	Deferred	Add driver for spc200
37	Open	Scheduled	hot fix for spc200
NONE	UNAFFILIATED CHANGES		

Total items: 6

☑ Show Active ☑ Include Incomplete Issues □ Include Child Streams

Name	In Folder	Version	Basis Version	Status	Id	Stream
⊟ spc200_driver.cpp	\src\	16/2	0/0		654	
spc200_driver.cpp	\src\	15/4	0/0		654	
spc200_driver.cpp	\src\	16/2	15/5	Missing	654	

- The upper Issues pane displays selected fields from the issue records whose change packages are "in the stream"— partially or completely. (The visible fields are determined by an administrator using the Schema Editor. See *Change Packages Tab* on page 398 of *Chapter A The Schema Editor (Administrators Only)* for more information.)
- The lower Change Package Contents pane displays the entries in the change package of the currently selected issue record.

Working in the Issues Pane

The highlighting of rows in the Issues pane conveys the following information:

- Dark black ("Unaffiliated Changes") -- This row connects you to any file versions that are not currently associated with an issue. When you choose to work in a Change Package environment, all files, in theory, should be associated with an issue. If they are not, it may flag a problem with your process. See *Unaffiliated Changes* on page 338 for more information.
- Light black highlighting indicates rows that are currently selected.
- Light brown highlighting indicates issues that are missing versions in this stream. (This is only visible if you have the **Include Incomplete Issues** checkbox enabled.) See *Incomplete Change Packages* on page 344 for more information.
- Light gray highlighting indicates that you have this issue’s contents selected in the lower pane.

You can control which issues are displayed and you can invoke commands on one or more issues' change packages.

Controlling Which Issues are Displayed

The **Show Active, Include Incomplete Issues,** and **Include Child Streams** check boxes appear at the bottom of the Issues pane and control which issues are displayed.

Show Active

If checked, an issue is included only if one or more of its versions was promoted *to* the stream, but has not yet been promoted *from* the stream to its parent. This helps you to concentrate on current programming efforts, rather than on those that were completed long ago.

Note: For the purposes of the Show Active filter, a **Revert to Basis** operation has the same effect as a **Promote** operation. For a workspace, substitute the phrase "created in" for "promoted to".

Include Incomplete Issues

If cleared, the listing includes only issues that are completely in the stream, where “completely in” means that for every change package entry, the head version (Version column) itself is in the stream, or one of its descendants is in the stream.

If checked, the listing includes all issues, including ones that are partially in the stream, where “partially in” means that for some (but not all) change package entries, the head version (Version column) itself is in the stream, or one of its descendants is in the stream.

On the Change Package Contents pane, the Status column and background shading indicate which change package entries are "in" the stream and which are not in ("missing" from) the stream.

Include Child Streams

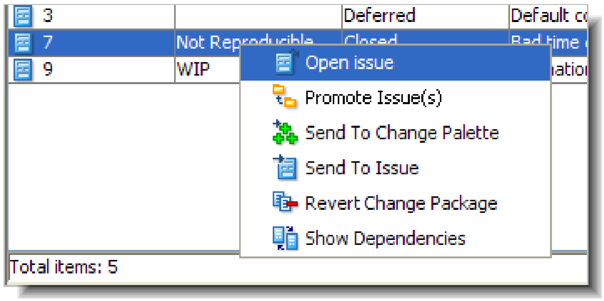
(Available only if **Show Active** is checked) If checked, displays issues that are in the streams below the selected stream. This can be helpful when tracking down streams or workspaces that may be responsible for incomplete change packages or unaffiliated changes.

Working in the Issues Pane

You can control which issues are displayed, and you can invoke commands on one or more issues' change packages.

Commands Available in the Issues Pane

Select one or more issues to invoke commands on the issues, or on their change packages.



These commands are available through context menus and in the toolbar of the issues pane.

Open Issue

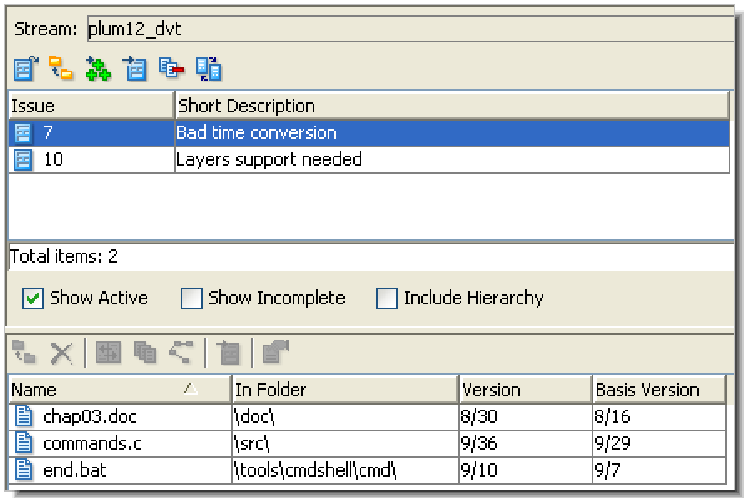
Open a form on the selected issue, and display the Changes tab.

Promote Issue(s)

Stream Issues tab (enabled only if *Show Active* is set and *Show Incomplete* is cleared): promotes each currently-active head version in the issue's change package to the parent stream. That is, if a version listed in the Version column is currently active in the stream on which you invoked the *Show Issues* command, that version is promoted to the parent stream.

Example:

The following illustration shows the Stream Issues tab for dynamic stream *plum12_dvt*.



Stream: plum12_dvt

Issue	Short Description
7	Bad time conversion
10	Layers support needed

Total items: 2

☑ Show Active ☐ Show Incomplete ☐ Include Hierarchy

Name	In Folder	Version	Basis Version
chap03.doc	\doc\	8/30	8/16
commands.c	\src\	9/36	9/29
end.bat	\tools\cmdshell\cmd\	9/10	9/7

Invoking the *Promote* command on issue #7 causes some or all these versions to be promoted. The versions promoted are the ones that are currently in the workspace's default group:

- version 8/30 of *chap03.doc*
- version 9/36 of *commands.c*
- version 9/10 of *end.bat*

Stream Diff (Issues) tab (works only for issues in the source stream, not for issues in the destination stream): promotes each currently-active head version in the issue's change package from the source stream to the destination stream. If there is an overlap between the two streams for any element, the promote operation fails with a "merge required" error box.

Send to Change Palette

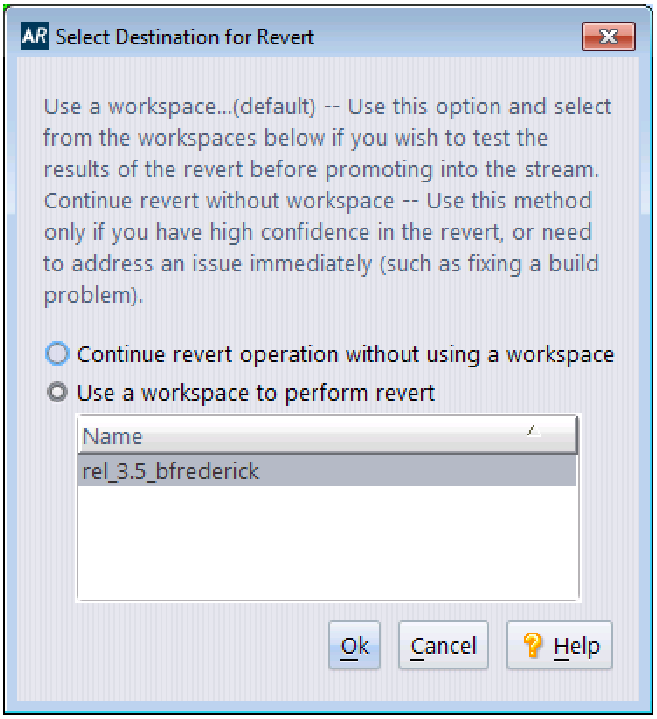
(enabled only if *Show Active* is set and *Show Incomplete* is cleared) Opens a Change Palette tab, containing each head version in the issue's change package.

Send to Issue

Adds (or combines) the selected issue record's change package entries with those of one or more other issue records. A popup window prompts you to specify the destination issue record(s).

Revert Change Package

(Formerly named *Revert by Change Package*.) Removes the changes in the selected issue's change package from the stream. A dialog appears offering you two options for performing this operation:



Select Destination for Revert

Use a workspace...(default) -- Use this option and select from the workspaces below if you wish to test the results of the revert before promoting into the stream. Continue revert without workspace -- Use this method only if you have high confidence in the revert, or need to address an issue immediately (such as fixing a build problem).

○ Continue revert operation without using a workspace

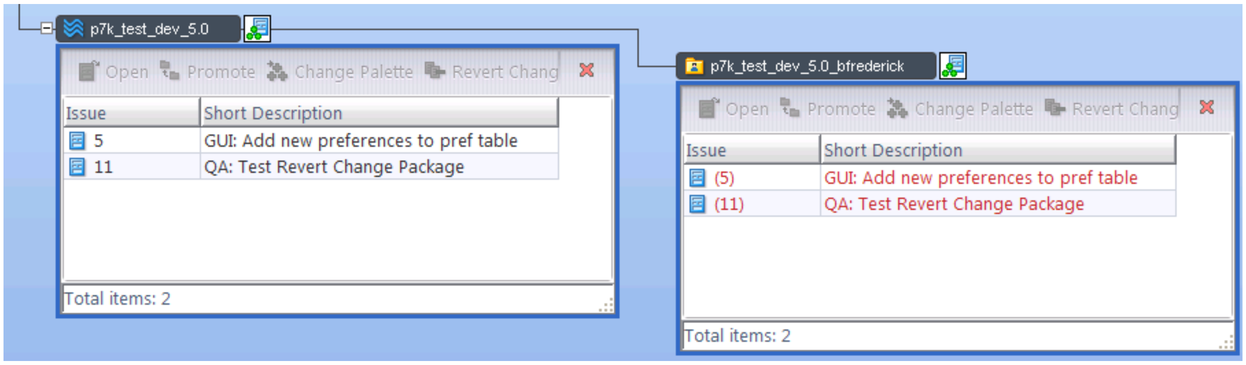
● Use a workspace to perform revert

Name
rel_3.5_bfrederick

In reverse order, these options are:

- **Use a workspace to perform revert** — Choose this option if you wish to test the results of the revert before promoting them back into the stream. You must choose a workspace to which you have access to perform the revert. (All workspace appear in the selection list. If you select a workspace to which you do not have access, AccuRev will display an error). The revert operation typically involves one or more reverse patch operations on file elements' contents, performed with the *Merge tool*.

  After you have reverted a change package, *the issue remains active in the stream until you promote the changes you performed in the workspace*. A special variant of the issue appears in red in the workspace Active Issue display, as shown in the following example. (The image has been modified to show both Active Issue displays simultaneously.)



Issue	Short Description
5	GUI: Add new preferences to pref table
11	QA: Test Revert Change Package

Total items: 2

Issue	Short Description
(5)	GUI: Add new preferences to pref table
(11)	QA: Test Revert Change Package

Total items: 2

When you promote the red reverted issue, the original issue disappears frim the backing stream and you can continue to promote the red reverted issue up the stream hierarchy.

Important: If no valid workspaces exist, this dialog is bypassed and you are prompted whether or not to proceed without a workspace (see below). A "valid" workspace is one of your own workspaces which has local (or accessible mounted) storage on the machine that the **revert** is being performed from. If none of your workspaces are attached to the stream the issue is being **revert**ed from, or for some reason are not accessible from the machine that the client is running on, you can still proceed with the revert, keeping the new file versions directly in the stream. However, AccuRev warns you that this is what happening if you choose to proceed with the **revert** operation.

Note: This feature is not intended to be used *within* a workspace; it is meant to be executed within a stream, but with the aid of a workspace to perform any necessary *Merge* operations. If you try to execute this operation directly within a workspace, AccuRev returns the following error message: "You need to have at least one workspace associated with stream..." Try the following workaround:

1. Create a new stream between the workspace and its backing stream.
2. Promote the issue to the new stream.
3. Invoke the *Revert Change Package Using Workspace* command on that issue in the new stream.

- **Continue revert operation without using a workspace —** Similar to *Use a workspace to peform revert*, except that the results are placed directly in the stream, without the requirement of a workspace. This means that the results of the revert will be immediately propagated downstream in the hierarchy.

  AccuRev prompts you for an issue number against which to the record the *Revert* changes. Either create a new issue, or select an existing issue that you created specifically for this *Revert*. (You should rarely—if ever—select an existing issue that has been used for previous operations.) Then peform whatever merge changes are required by the revert.

  Important: If no workspaces to which you have access exist, the dialog described above is bypassed and you are simply prompted whether or not to proceed without a workspace.

  Note: *Continue revert operation without using a workspace* saves the step of merging and testing the revert in a workspace and then promoting the changes, but at the risk of propagating the changes as soon as the revert has been completed. This feature should not be used casually by developers, as you could easily propagate complex, untested changes to other users. This feature is useful for reverting relatively simple changes, or for backing out a change in an isolated build stream. AccuRev recommends that you build and test these changes in an updated workspace to verify the results, before promoting or cross-promoting the changes to other users.

Show Dependencies

Opens an *Issue Dependencies* tab, displaying the change package dependencies of the selected issue record(s).

Working in the Change Package Contents Pane

Select any change package entry and invoke any of the commands described below. In most cases, the operation is performed on the entry's head version (**Version** column). You can invoke the *Remove* and *Send to Issue* commands on a selection consisting of two or more change package entries.

Open (equivalent to double-click)

Windows: Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

UNIX: Open a text editor on the file.

View

Open a text editor on a temporary copy of the currently selected version (text files only).

Save As

Copy the currently selected version to another filename.

Promote

Promote the head version(s) of the selected entry(s) to the parent stream.

Remove

Remove the selected entry(s) from the change package.

Diff Across All Segments

Compare the most recent version of a multi-segment, text file that was promoted against an AccuWork issue with the earliest version of the file that was promoted against the same issue. This diff procedure ignores changes to the file that were promoted against a different AccuWork issue.

Diff Against Basis

Compare the selected version with the corresponding basis version.

Show History

Open a *History Browser* tab, containing the transactions involving the selected file or directory.

Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected file or directory, and their interrelationships (ancestry).

Send to Issue

Copies the selected entry(s) to another issue record's change package. If there is an existing entry for an element in a destination change package, an attempt is made to *combine* the existing entry with the entry you specified.

The default query is executed, and the results are displayed in a *dialog*. You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

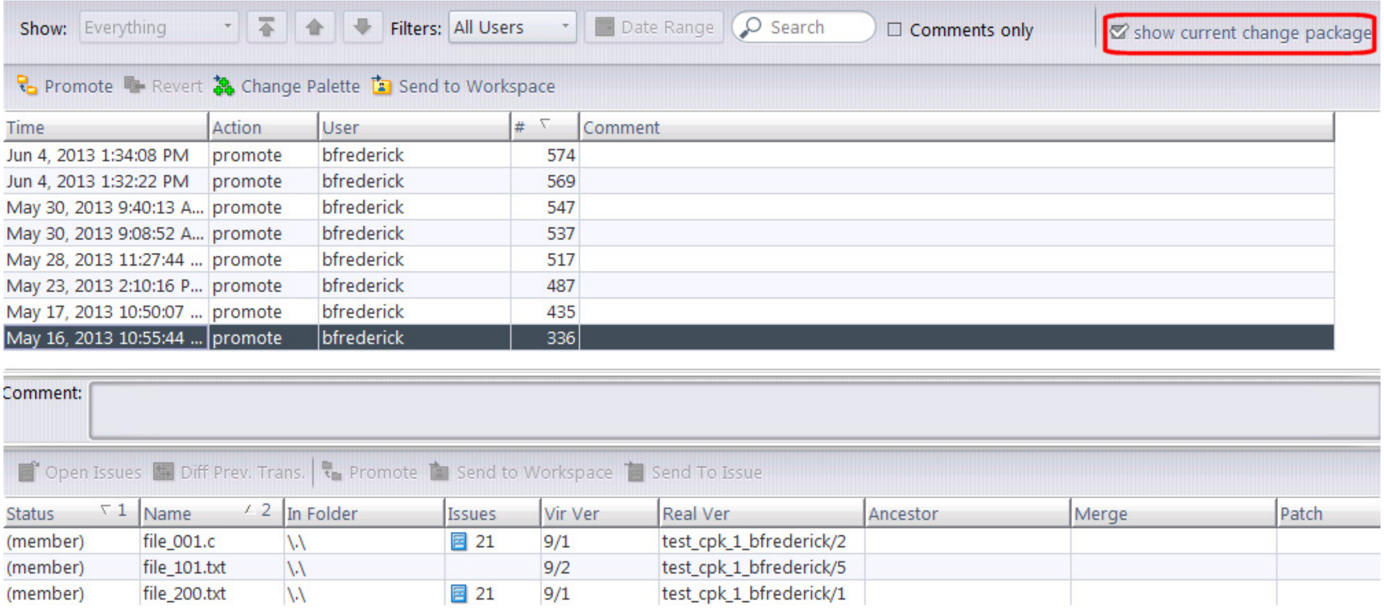
Properties

Displays a message box with *information* about the selected element.

A Note about the Active Transactions Tab

Although most change package-related functionality is concentrated in the **Active Issues** tab, you should be aware of the **show current change package** control in the **Active Transactions** tab.

When the **show current change package** checkbox is selected (the default setting), the lower pane, which shows the change package details for the selected transaction, reflects the change package contents at the current time. In the following illustration, for example, `file_001.c` and `file_200.txt` are both associated with Issue #21.

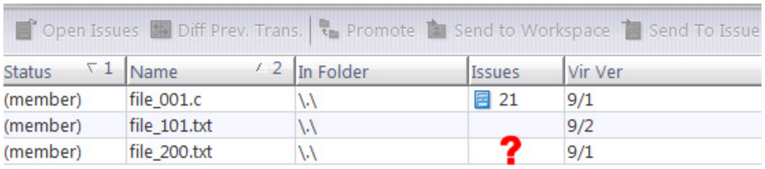


Time	Action	User	#	Comment
Jun 4, 2013 1:34:08 PM	promote	bfrederick	574	
Jun 4, 2013 1:32:22 PM	promote	bfrederick	569	
May 30, 2013 9:40:13 A...	promote	bfrederick	547	
May 30, 2013 9:08:52 A...	promote	bfrederick	537	
May 28, 2013 11:27:44 ...	promote	bfrederick	517	
May 23, 2013 2:10:16 P...	promote	bfrederick	487	
May 17, 2013 10:50:07 ...	promote	bfrederick	435	
May 16, 2013 10:55:44 ...	promote	bfrederick	336	

Comment:

Status 1	Name 2	In Folder	Issues	Vir Ver	Real Ver	Ancestor	Merge	Patch
(member)	file_001.c	\.\	21	9/1	test_cpk_1_bfrederick/2			
(member)	file_101.txt	\.\		9/2	test_cpk_1_bfrederick/5			
(member)	file_200.txt	\.\	21	9/1	test_cpk_1_bfrederick/1			

However, what if `file_200.txt` was removed from Issue #21's change package at some point *after* the selected transaction? With the **show current change package** checkbox selected, Issue #21 disappears from the **Issues** column of `file_200.txt` in the lower pane.



Status 1	Name 2	In Folder	Issues	Vir Ver
(member)	file_001.c	\.\	21	9/1
(member)	file_101.txt	\.\		9/2
(member)	file_200.txt	\.\	?	9/1

This is because, *at the current time*, `file_200.txt` is no longer associated with Issue #21's change package. If you wanted to see the file's change package association at the time of the selected transaction, you would clear the **show current change package** checkbox.

For general information about the **Active Transactions** tab, see *The Active Transactions Tab (Show Active Transactions)* on page 172.

The Stream Diff (Issues) Tab

The Stream Diff (Issues) tab is similar to the Show Active Issues tab (*The Active Issues Tab* on page 171), but allows you to compare the issues in two different streams, and provides access to operations that are suitable for such a context.

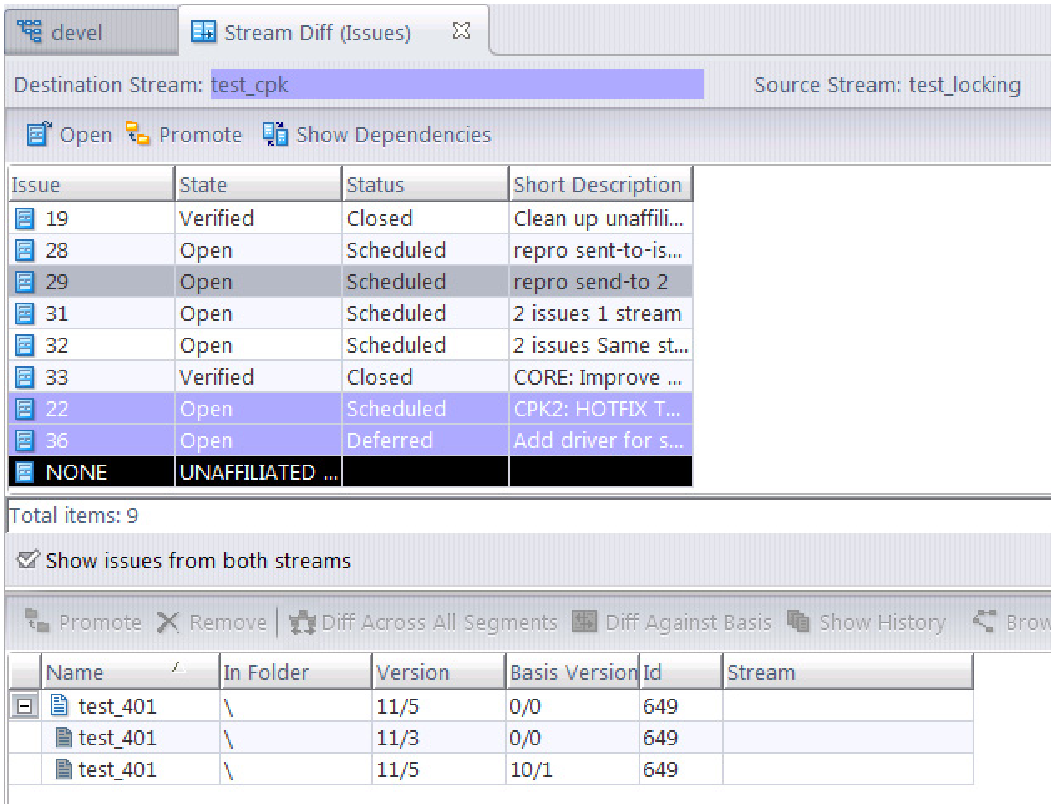
Opening the Tab

To open a Stream Diff (Issues) tab:

1. In the Stream Browser, select a stream, snapshot, or workspace.
2. Choose the **Show Diff by Issues** command from the selection's context menu or the **Actions** menu.
3. The mouse pointer changes to . Left-click another stream, snapshot, or workspace.

Stream Diff (Issues) Tab Layout

The Stream Diff (Issues) tab is very similar to the Show Active Issues tab, but with the exceptions noted below:



Destination Stream: test_cpk

Source Stream: test_locking

Issue	State	Status	Short Description
19	Verified	Closed	Clean up unaffili...
28	Open	Scheduled	repro sent-to-is...
29	Open	Scheduled	repro send-to 2
31	Open	Scheduled	2 issues 1 stream
32	Open	Scheduled	2 issues Same st...
33	Verified	Closed	CORE: Improve ...
22	Open	Scheduled	CPK2: HOTFIX T...
36	Open	Deferred	Add driver for s...
NONE	UNAFFILIATED ...		

Total items: 9

☑ Show issues from both streams

Name	In Folder	Version	Basis Version	Id	Stream
⊟ test_401	\	11/5	0/0	649	
test_401	\	11/3	0/0	649	
test_401	\	11/5	10/1	649	

- When you're comparing the contents of two streams, this pane displays the issues that appear in one stream, but not the other.
- When enabled, the **Show issues from both streams** checkbox will display in a highlighted color any issues from the target stream that do not appear n the source stream.

The Issue Dependencies Tab

The Issue Dependencies tab contains a hierarchical display of the **dependencies** of a selected set of issue records -- that is, the set of issue records on which the selected records depend.

> **Note:** In this description, "issue record N" is shorthand for "change package of issue record N".

From this tab, you can propagate issues' change packages to other streams, using simple *Promote* commands and/or the *Change Palette*. When propagating an issue's change package to another stream, you can choose to propagate the changes in the issue's dependencies, as well.

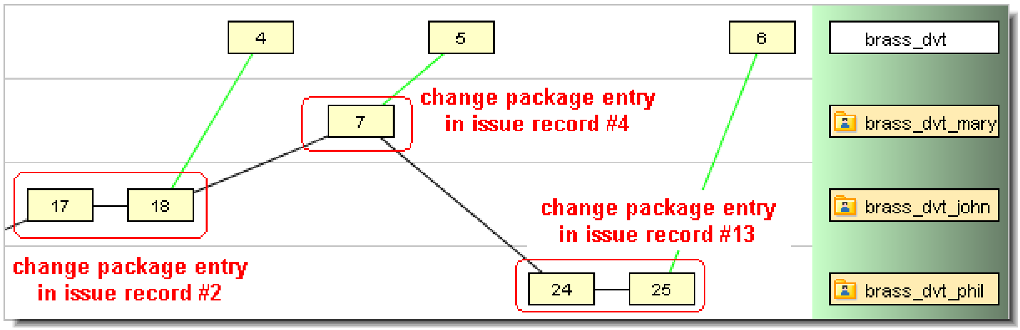
Opening an Issue Dependencies Tab

In a *Stream Issues or Stream Diff (Issues)* tab, select one or more issue records in the Issues pane, then click the *Show Dependencies* toolbar button or invoke the *Show Dependencies* command from the selection's context menu.

Change Package Dependencies

The concept of a change package depending on another change package (or an issue record depending on another issue record) is complex, utilizing several aspects of AccuRev's product architecture. But the basic idea is familiar: when you ask for something, you sometimes get more than you asked for.

This Version Browser example shows three sets of changes to an element, each assigned to a different issue record: #2, #4, and #13.



change package entry in issue record #4

change package entry in issue record #13

change package entry in issue record #2

At this point, issue record #13 is active in stream **brass_dvt**. Promoting that issue record — for example, using the *Stream Issues* tab — promotes version **brass_dvt/6** (real version: **brass_dvt_phil/25**).

Promoting issue record #13 would give you "more than you asked for": it would automatically promote issue records #2 and #4, as well. We describe this situation by saying that issue record #13 depends on issue records #2 and #4.

In practice, most issue records' change packages contain entries for multiple elements, not a single element. In many cases, the promotion causes issue records #2 and #4 to be *"partially in" the destination stream, not "completely in"*.

Dependency Defined

The discussion above is conceptual and informal. Here's a rigorous definition of change package dependency:

> Issue A depends on issue B, in the context of a promotion from stream S1 to stream S2, if issue A is active in stream S1 and at least one element meets all the following conditions:

- The element has a change package entry in both issue A and issue B.
- The head version of the element's entry in issue A is a descendant of the head version in issue B. (That is, the A version contains the changes made in the B version.)
- Stream S2's version of the element is not either the head version of the element's entry in issue B or a descendant of it. (That is, the changes made in the B version have not yet been promoted to stream S2.)

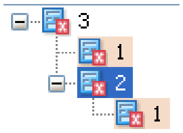
The Dependency Context

Note that the definition of change package dependency cites a pair of streams as its context. The Issue Dependencies tab shows dependencies in the context of a promotion ...

- *from* the stream or workspace on which you opened a Stream Issues tab ...
- *to* its parent stream.

The Dependency Hierarchy

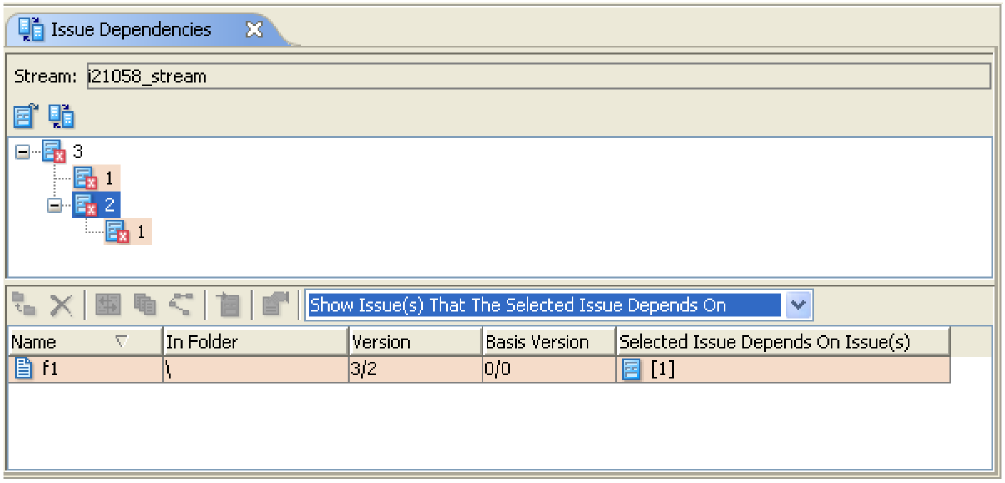
For each issue record, there is a dependency hierarchy: the issue directly depends on several issues; each of those issues directly depends on several issues; and so on.



The Issue Dependencies tab uses a tree control to represent the dependency hierarchy.

Issue Dependencies Tab Layout

The Issue Dependencies tab layout is similar to that of the *Stream Issues tab*: two panes, each with its own toolbar. The upper pane is the Issues Pane, and the lower one is the Change Package Contents Pane:



Stream: i21058_stream

Name	In Folder	Version	Basis Version	Selected Issue Depends On Issue(s)
f1	\	3/2	0/0	[1]

Issues Pane

The Issues pane displays the issue records that were selected for the *Show Dependencies* command.

> **Note:** For each issue record, it displays the fields specified in the *Change Package Results* section of the Schema Editor's Change Packages tab.

If an issue record has dependencies, a tree control appears, initially in the collapsed state. Color-coding and icon differences indicate dependency-related information about the individual issue records:

As you expand issue records' dependency hierarchies, you may discover that some issue records that you initially selected for the *Show Dependencies* command also appear as dependencies at lower levels. To help you keep track, color-coding is used at the lower levels:

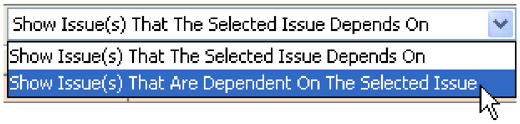
- Issue records that you initially selected for the *Show Dependencies* command appear without a tinted background.
- Issue records that you *did not* initially select appear with a tinted background. These are the "new records" added to the listing by the dependency analysis.

Variations in an issue record's icon provide more information about that issue's dependencies:

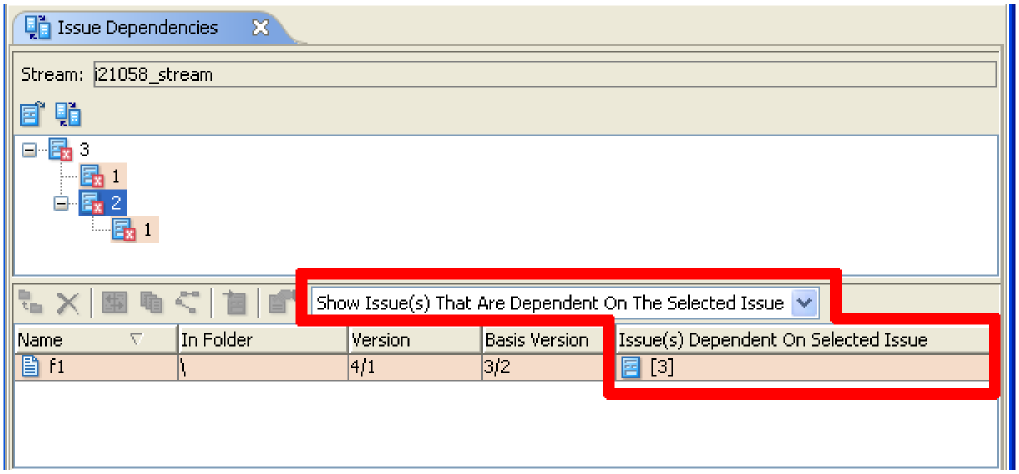
- (no sub-icon): the issue has no dependencies in this context.
- (yellow warning sub-icon): all of the issue's dependencies are included in the set you initially selected for the *Show Dependencies* command (the issues at the top hierarchy level).
- (red warning sub-icon): some of the issue's dependencies are *not* included in the set you initially selected for the *Show Dependencies* command.

Change Package Contents Pane

The information displayed in the Change Package Contents pane varies depending on the setting of the dependencies drop-down list.


Show Issue(s) That The Selected Issue Depends On
Show Issue(s) That The Selected Issue Depends On
Show Issue(s) That Are Dependent On The Selected Issue

Toogle between the two settings to display dependencies to and from the issue selected in the upper pane. This changes the information displayed in the bottom pane.



Stream: 21058_stream

Show Issue(s) That Are Dependent On The Selected Issue

Name	In Folder	Version	Basis Version	Issue(s) Dependent On Selected Issue
f1	\	4/1	3/2	[3]

The far right column toggles to display the "Depends On" or "Are Dependent On" issues.

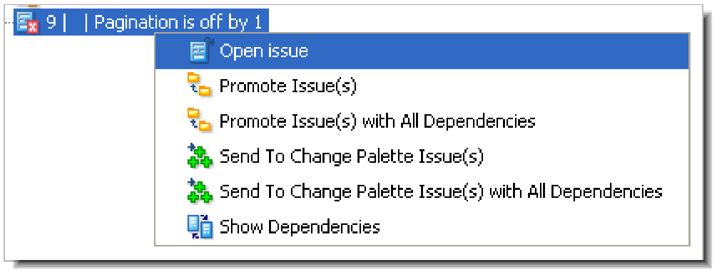
The versions are displayed as follows:

Show Issue(s) That The Selected Issue Depends On -- Displays the versions in those issues that the selected issue depends on.

Show Issue(s) That Are Dependent On The Selected Issue -- Displays the versions in the selected issue.

Working in the Issues Pane

Use the tree control to navigate the dependency hierarchy of an issue record.


9 | | Pagination is off by 1
Open issue
Promote Issue(s)
Promote Issue(s) with All Dependencies
Send To Change Palette Issue(s)
Send To Change Palette Issue(s) with All Dependencies
Show Dependencies

Select one or more issue records, and invoke any of these commands from the selection's context menu:

Open Issue

Open a form on the selected issue record, and display the Changes tab.

Promote Issue(s)

Promote each currently-active head version in the issue(s)' change package(s) to the parent stream. That is, if a version listed in the Version column is currently active in the stream from which you invoked the *Show Dependencies* command, that version is promoted to the parent stream.

Promote Issue(s) with All Dependencies

Similar to *Promote Issue(s)*, but also include the issue records on which the selected issue(s) depend.

Send to Change Palette Issue(s)

Similar to *Promote Issue(s)*, but instead of promoting active versions, invoke the *Send to Change Palette* command on them.

Send to Change Palette Issue(s) with All Dependencies

Similar to *Promote Issue(s) with All Dependencies*, but instead of promoting active versions, invoke the *Send to Change Palette* command on them.

Show Dependencies

Open another Issue Dependencies tab, with the selected issue records at the top level.

Working in the Change Package Contents Pane

In this pane, you can perform the same operations as in the *corresponding pane* of a Stream Issues or Stream Diff (Issues) tab.

Variants

Issue *variants* are automatically-generated, internal variations of issues and their change packages that AccuRev creates to keep track of changes caused by patch, merge, and revert operations. In previous AccuRev releases, users were prompted to create tracking issues (or they might create totally different issues) and these issues could be difficult to relate to the original issue(s).

With variants, AccuRev now handles this bookkeeping automatically, and you generally do not need to deal with variants in your day-to-day work. But they are visible in a limited number of areas of the GUI, and this discussion explains why they are there and what they do for you. This section is directed at administrators and power users who need to understand the ramifications and history of patch, merge, and revert operations, or who want to understand how these operations are being recorded by AccuRev.

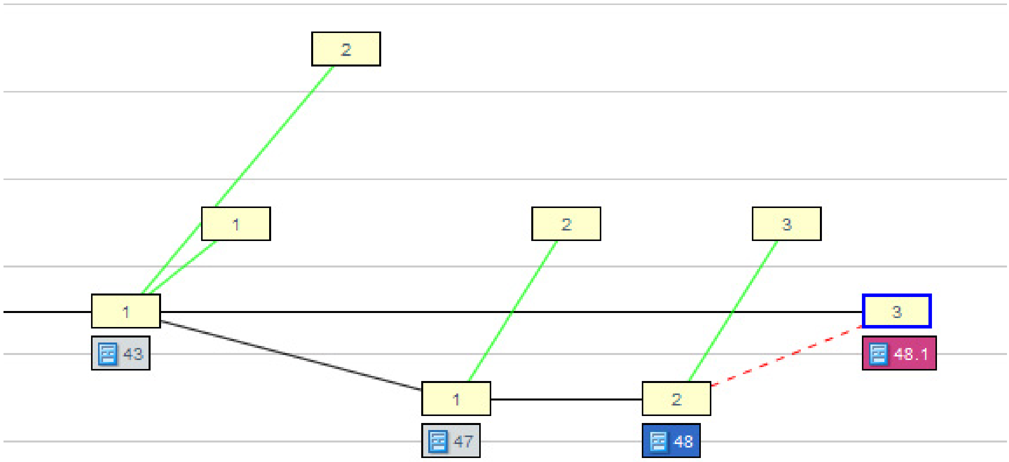
When AccuRev users prepare to make a change, they typically make the following determination:

- Is the change entirely new, and generally unrelated to previous changes (such as a new feature, or a unique bug fix)? If so, create a new issue.
- Is the change a continuation of a previous change (a tweak to the previously promoted enhancement, or a more comprehensive fix to a known bug)? If so, simply reuse the existing issue.
- Is the change similar to -- but somewhat different than -- an existing change (for example, making an enhancement work on a new platform, so that much of the code applies, but with some fundamental differences). In this case, you may want to patch an existing change package into your current work, and this is where AccuRev will handle the bookkeeping automatically with a variant of the existing issue.

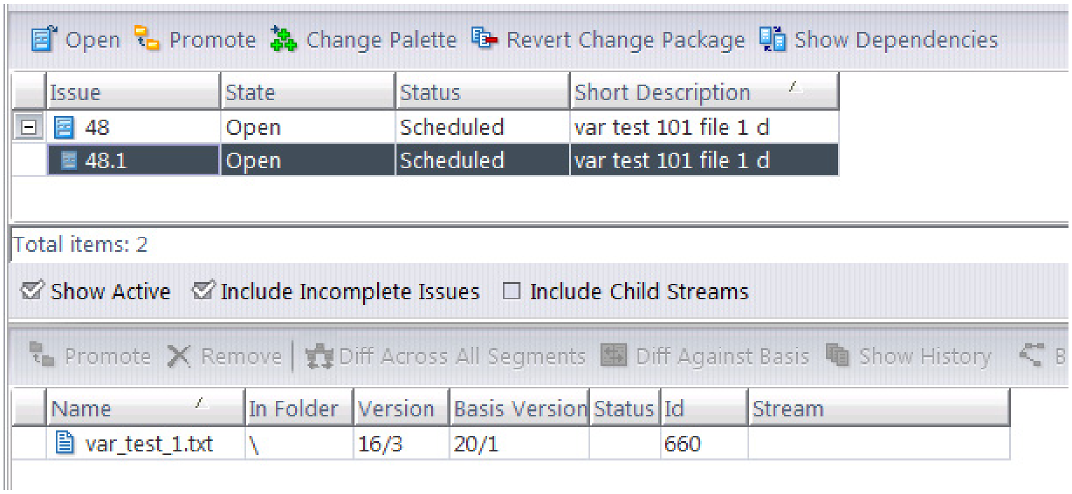
A variant appears as an issue number with a period and number appended to it. For example, if you patch issue #48 and AccuRev needs to create a variant to track that change, the variant will appear in a limited number of places in the UI as "48.1".

Here are the places in the UI where you might see a variant:

- the Version Browser



- the **Show Active Issues** view:



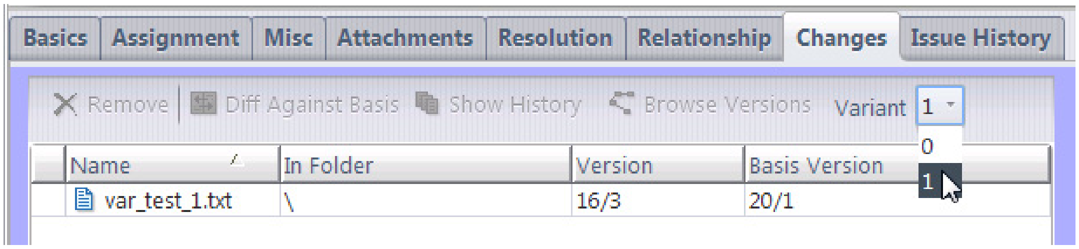
Issue	State	Status	Short Description
48	Open	Scheduled	var test 101 file 1 d
48.1	Open	Scheduled	var test 101 file 1 d

Total items: 2

☑ Show Active ☑ Include Incomplete Issues ☐ Include Child Streams

Name	In Folder	Version	Basis Version	Status	Id	Stream
var_test_1.txt	\	16/3	20/1		660	

- the **Changes** tab of an Issue Form:



Variant 1

Name	In Folder	Version	Basis Version
var_test_1.txt	\	16/3	20/1

- the CLI output from a **cpkdescribe** or **cpkdepend** command:

```
C:> accurev cpkdescribe -I 48 -p devel

Issue: 48
/var_test_1.txt      21/2 21/1


Issue: 48.1
/var_test_1.txt      16/3 20/1
```

- the XML output from a **cpkdescribe** or **cpkdepend** command:

```
C:> accurev cpkdescribe -I 48 -p devel -fx

<?xml version="1.0" encoding="utf-8"?>
<acResponse>
  <issues>
      <issue>
          <issueNum>48</issueNum>
          <elements>
              <element
                 id="660"
                 real_version="21/2"
                 basis_version="21/1"
                 location="/var_test_1.txt"
                 dir="no"
                 elemType="text"/>
          </elements>
      </issue>
      <issue>
          <issueNum
             variant_num="1">48</issueNum>
          <elements>
              <element
                 id="660"
                 real_version="16/3"
                 basis_version="20/1"
                 location="/var_test_1.txt"
```

```
                    dir="no"
                    elemType="text"/>
             </elements>
         </issue>
     </issues>
   </acResponse>
```

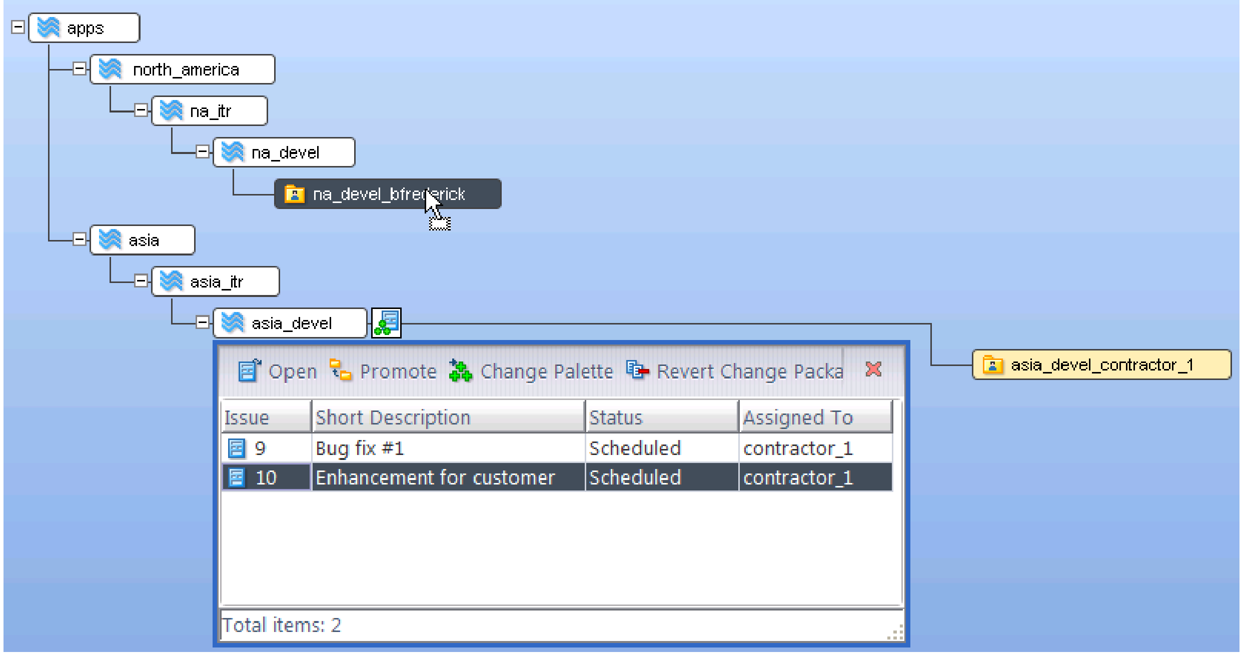
Again, it is important to emphasize that in your day-to-day work, you do not need to concern yourself with variants. AccuRev is handling this bookkeeping for you automatically. But you can view the existence of variants to help understand the history of changes packages.

Also note that although tracking issues are no longer required to record these kinds of changes, tracking issues are still available in certain scenarios if you prefer to use them. See *Tracking Issues* on page 375 for more information.

Example

The following very basic example demonstrates a scenario in which AccuRev creates a variant:

1. A new source file (`app_src_rel_1a.txt`) for a new application is created by the development team in North America, in workspace `na_devel`.
2. This new file is promoted up the stream hierarchy against Issue #8 as the product is developed and released.
3. A maintenance team in Asia makes a bug fix to this file for a specific customer, and promotes it part way up the stream hierarchy against Issue #9.
4. A short time later, the maintenance team makes another small change to this file for a customer-requested enhancement, and promotes it part way up the stream hierarchy against Issue #10.
5. Back in North America, the development team decides that they need to incorporate the bulk of the Issue #10 enhancement into their own work for the next release. So they choose to patch the Issue #10 enhancement into their development stream. They do this by dragging and dropping Issue #10 from the Active Issues view of the `asia_devel` stream, directly into the workspace of the developer in North America.

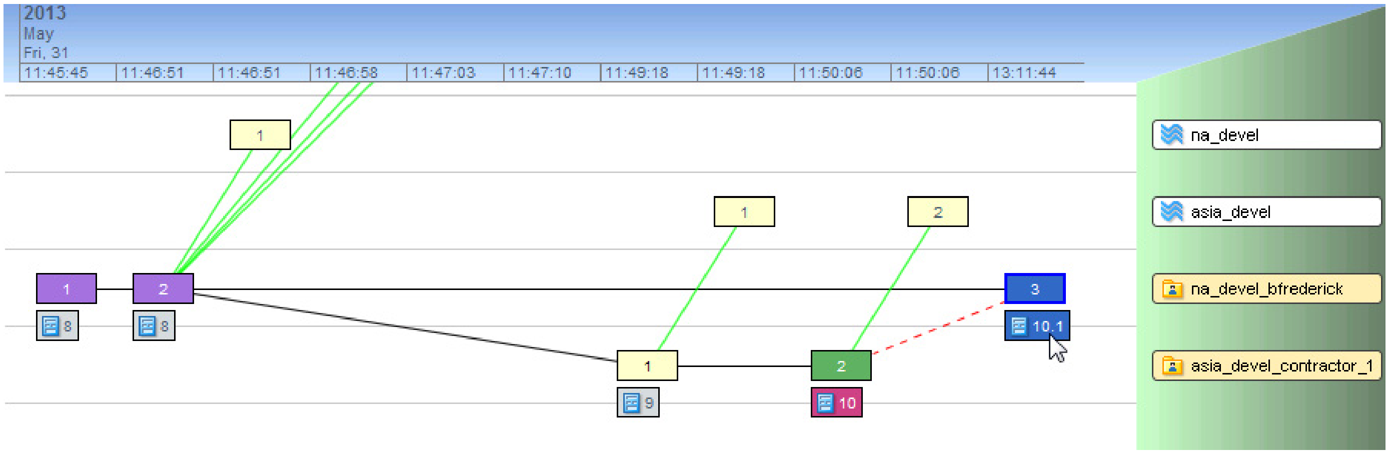


Issue	Short Description	Status	Assigned To
9	Bug fix #1	Scheduled	contractor_1
10	Enhancement for customer	Scheduled	contractor_1

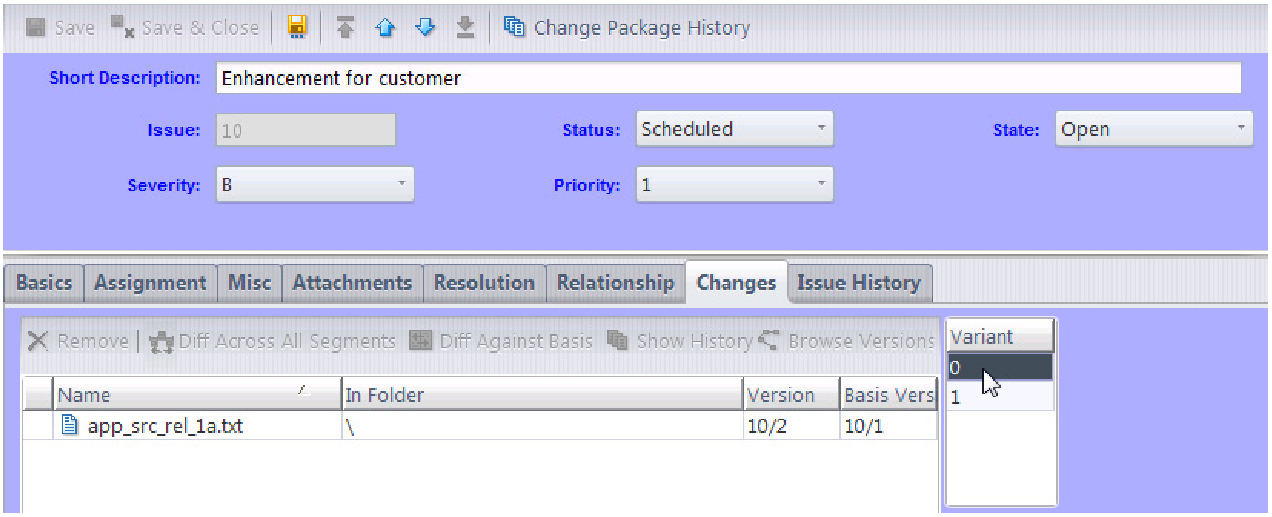
Total items: 2

6. This brings up the Diff view so the changes can be merged into the developer's copy of the file. He chooses "Take their change" and saves the file.

7. If you bring up this file in the Version Browser, you can see that the result of the patch is recorded in a variant of Issue #10 labeled "10.1".



8. Double-clicking the issue box for 10.1 brings up the Issue Form for Issue #10, focused on the Changes tab, showing the original change package for this issue (variant "0"), showing the basis and head versions from the original `asia_devel` workspace (#10).



Short Description: Enhancement for customer

Issue: 10

Status: Scheduled

State: Open

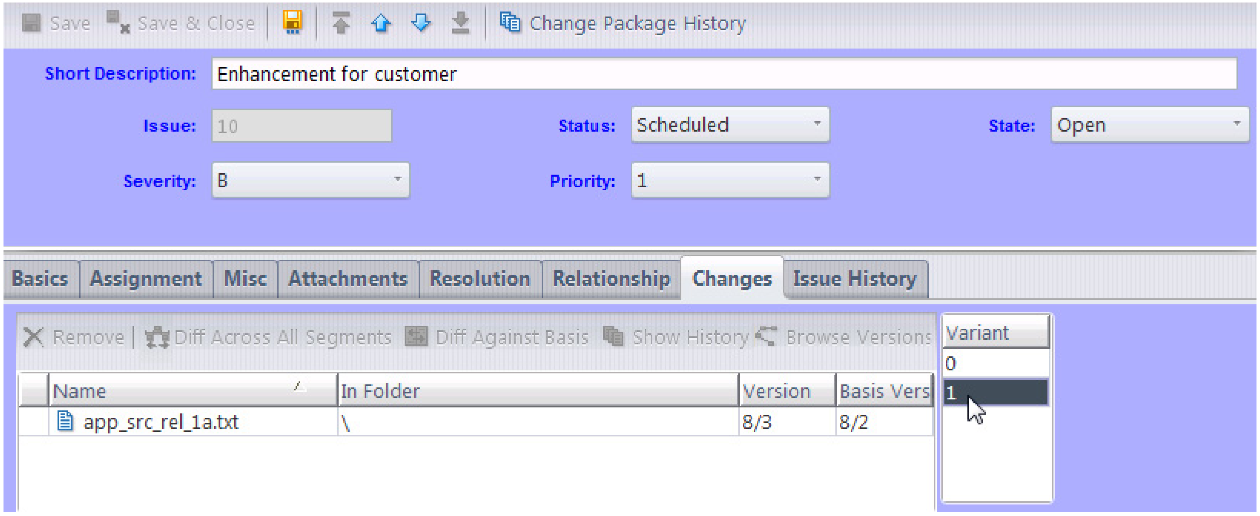
Severity: B

Priority: 1

Name	In Folder	Version	Basis Vers
app_src_rel_1a.txt	\	10/2	10/1

Variant
0
1

9. Clicking on variant “1” switches the Changes tab to show the patched basis and head versions from the `na_devel` workspace (#8):



Short Description: Enhancement for customer

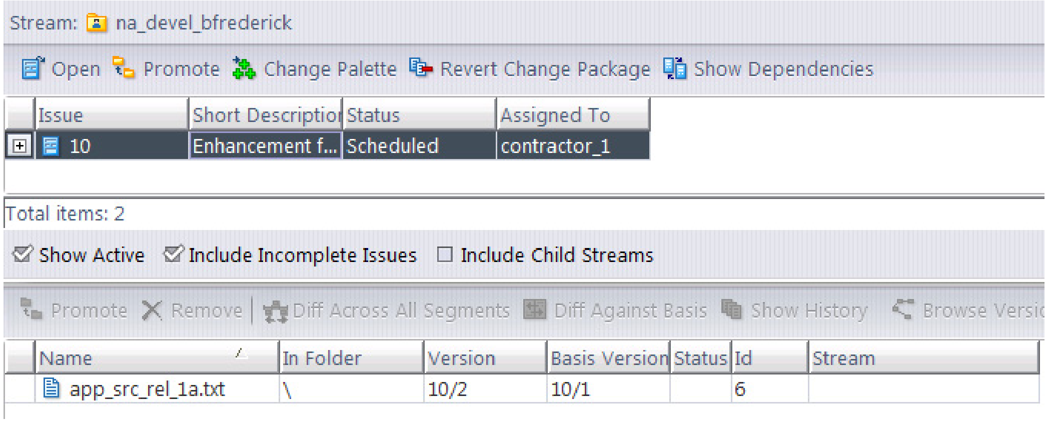
Issue: 10 Status: Scheduled State: Open

Severity: B Priority: 1

Name	In Folder	Version	Basis Vers
app_src_rel_1a.txt	\	8/3	8/2

Variant
0
1

10. A similar display is available if you select the patched workspace and click **Show Active Issues**. The display comes up by default on the main issue:



Stream: na_devel_bfrederick

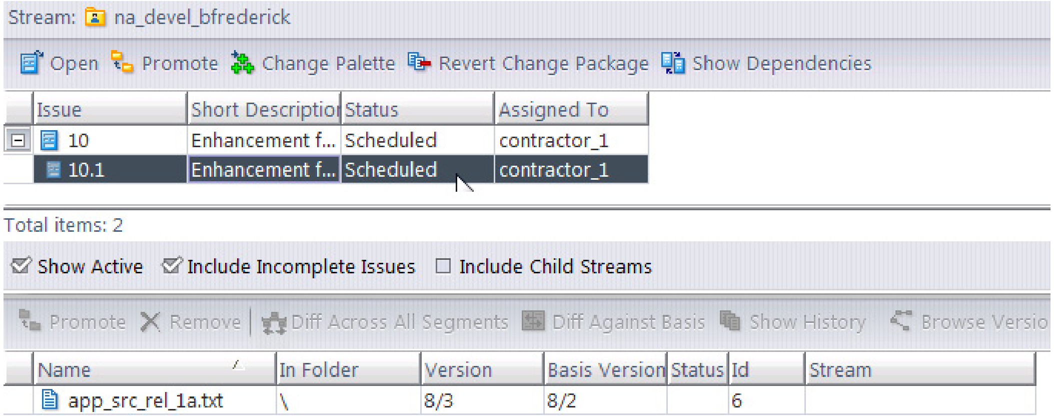
Issue	Short Descriptio	Status	Assigned To
⊞ 10	Enhancement f...	Scheduled	contractor_1

Total items: 2

☑ Show Active ☑ Include Incomplete Issues ☐ Include Child Streams

Name	In Folder	Version	Basis Version	Status	Id	Stream
app_src_rel_1a.txt	\	10/2	10/1		6	

11. Click the “+” icon to expand the display of any variants, then select the variant line to show its change package entries in the lower pane.



Stream: na_devel_bfrederick

Issue	Short Descriptio	Status	Assigned To
⊟ 10	Enhancement f...	Scheduled	contractor_1
10.1	Enhancement f...	Scheduled	contractor_1

Total items: 2

☑ Show Active ☑ Include Incomplete Issues ☐ Include Child Streams

Name	In Folder	Version	Basis Version	Status	Id	Stream
app_src_rel_1a.txt	\	8/3	8/2		6	

Notes About More Complex Scenarios

The previous example showed a simple case of patching a single issue. More complex scenarios can have the following ramifications regarding variants:

- Incomplete variants will cause the issue to be incomplete (see *Incomplete Change Packages* on page 344). A variant is considered to be "complete" only if all the relevant elements (eids) and all the relevant versions of those eids are present in the stream. If a variant is incomplete, then the change package for that issue is considered incomplete in that stream. This means that if you have defined a rule that prohibits the promotion of incomplete issues, an incomplete variant will prevent the promotion of its issue.
- If you patch by dragging an issue from a stream where it has multiple variants, AccuRev creates a new variant in the target stream containing the contents of all the variants source variants. For example, say that stream "Stream1" contains issue #5, with variants 5.1 and 5.2. If you patch this issue to "Stream2", Stream2 will end up with Issue #5 with a variant labeled 5.3, containing all the eids and versions that were contained in variants 5.1 and 5.2

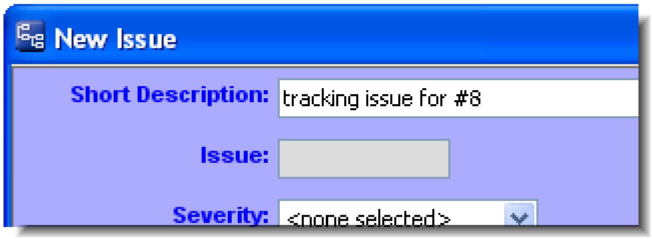
Tracking Issues

In AccuRev versions prior to 6.0, tracking issues were required to track the changes that are now handled automatically by variants.

Although tracking issues are no longer required for many of these scenarios, they are still sometimes required after you cross-promote and patch multiple issues to a destination stream. And you may still choose to use them if for some reason you wish to have a patch operation (for example) tracked by its own unique issue rather than by a variant of an existing issue.

A description of how you might be prompted by the Change Palette to specify a tracking issue after a multi-issue patch operation is available in the example in *Working with the Issue Change Palette* on page 259.

A single tracking issue can keep track of the additional changes for any number of original issues. Thus, it makes sense to select **Create Issue** the first time you need to specify a tracking issue, and select **Use Existing** on subsequent uses of promote-by-issue in the same workspace. You might also need to select **Use Existing** if you are not allowed to create new issues.



Short Description: tracking issue for #8

Issue:

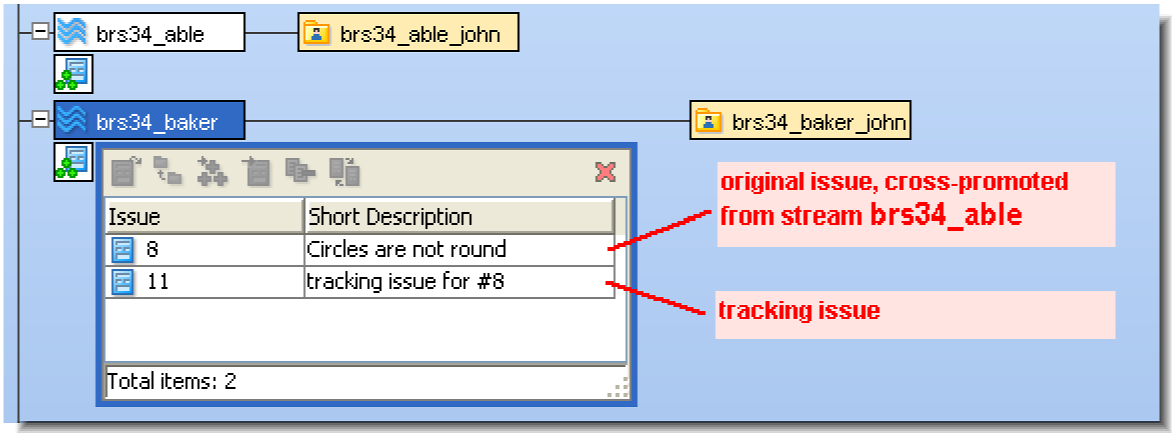
Severity: <none selected>

Either way, an AccuWork Issue form appears, in which you fill in the fields of the new or existing tracking issue. As usual, click the form's **Save** button when you are finished.

AccuRev automatically proceeds with your original **Promote**. It prompts you for a comment, then sends the versions from the workspace to the original destination stream. The cross-promotion of the original issue(s) is now complete.

Working with a Tracking Issue

The Stream Browser shows that the original issue (in our example, **#8**) has been promoted from the source stream (**brs34_able**) to the destination stream (**brs34_baker**). The tracking issue (**#11**) records the additional changes to issue #8's elements that have been propagated to the destination stream.



Issue	Short Description
8	Circles are not round
11	tracking issue for #8

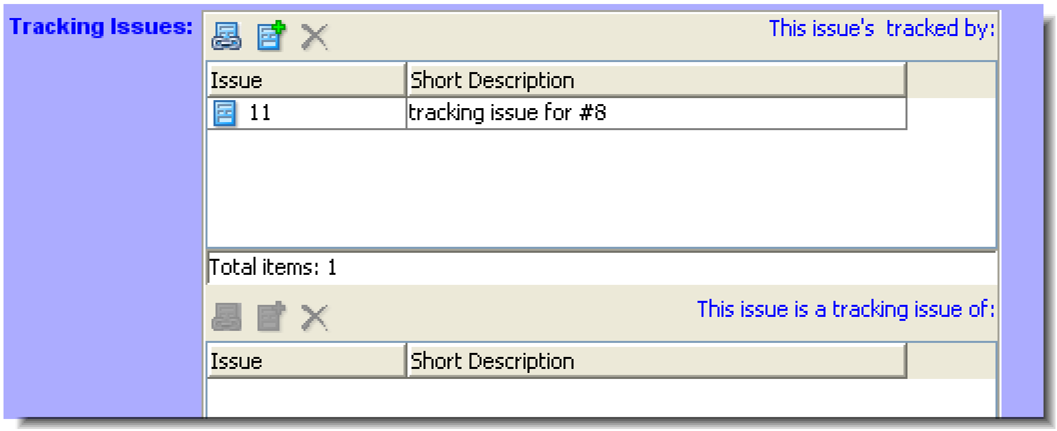
Total items: 2

original issue, cross-promoted from stream brs34_able

tracking issue

AccuRev provides *change package dependency* tracking, which ensures that a **Promote** operation sends a self-contained, consistent set of changes to the destination stream. A tracking issue and the corresponding original issue(s) are connected by a change package dependency: the tracking issue depends on the original issue(s).

The best way to monitor such connections is with a Relationship field whose type is **Track**. (For details about Relationship fields, see *Chapter A The Schema Editor (Administrators Only)*.) For example, when issue #8 is viewed in an Issue form, its connection to issue #11 might be displayed like this:


Tracking Issues:

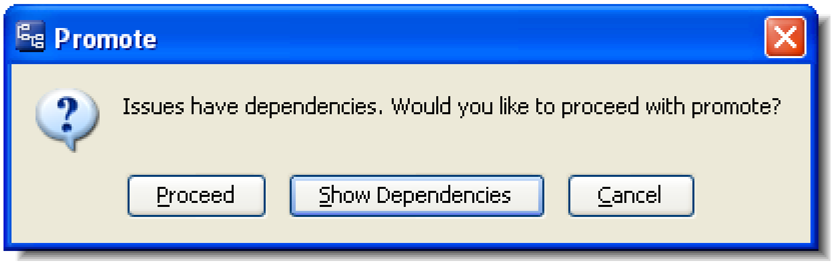
This issue's tracked by:

Issue	Short Description
11	tracking issue for #8

Total items: 1

This issue is a tracking issue of:

Issue	Short Description



Issues have dependencies. Would you like to proceed with promote?

As with all change package dependencies, AccuRev warns you if you attempt to promote a tracking issue without its dependencies, the original issue(s). In this situation, do NOT click **Proceed**. It is important to always promote both the original issue(s) and the tracking issue at the same time. So the correct procedure is to **Cancel**, select both the original and tracking issues, then invoke **Promote** again.

As you propagate issues up (or across) the stream hierarchy, you must continue to obey this same rule:

> **Always promote original issues and the corresponding tracking issues at the same time.**

If you attempt to **Promote** an original issue alone, without including its tracking issue, AccuRev prompts you to promote the tracking issue as well.

A. The Schema Editor (Administrators Only)

Administrators use the Schema Editor to configure AccuWork, including issue record layout and validation. Each depot can contain a separate AccuWork configuration, including a custom-designed Issue form, through which users create and modify the issue records. You can make the Issue form "smart" by defining validations (edit checks) that specify default values, required fields, and interrelationships among multiple fields.

"No Schema" Error

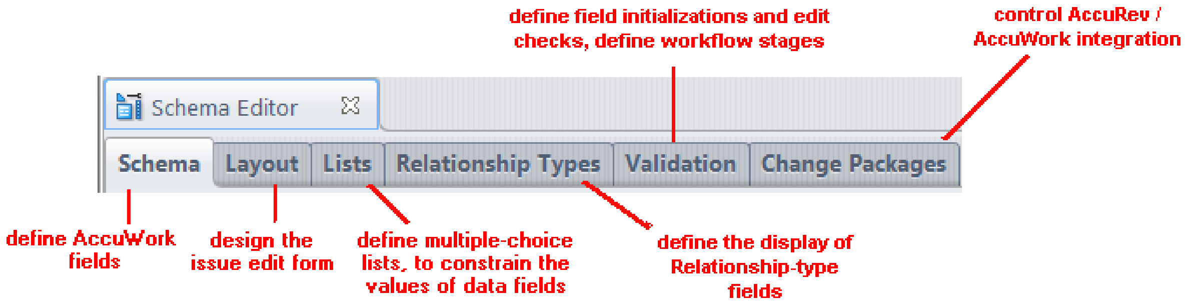
This error box appears when AccuWork has not been configured in the current depot. Use the procedures described in this chapter to define a schema.

Invoking the Schema Editor

Choose ***Admin > Schema Editor*** from the GUI main menu. The first time you invoke this command in a particular depot, AccuWork offers to use the repository's default schema. Accepting this offer copies a set of XML-format configuration files from the *site_slice/ dispatch/config* subdirectory to this depot.

> **Note:** The default schema does not actually become the schema for this depot until you click the Schema Editor's Save button.

The Schema Editor tab includes these subtabs:


define field initializations and edit checks, define workflow stages
control AccuRev / AccuWork integration
Schema Editor
Schema
Layout
Lists
Relationship Types
Validation
Change Packages
define AccuWork fields
design the issue edit form
define multiple-choice lists, to constrain the values of data fields
define the display of Relationship-type fields

Saving Changes to the Schema

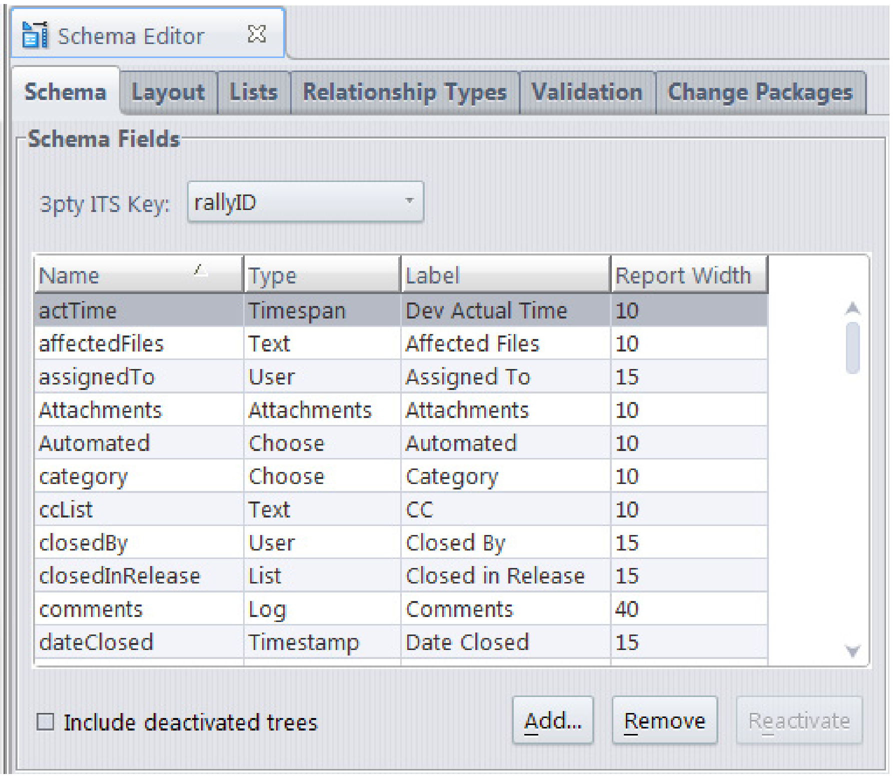
At any time while working in the Schema Editor, you can click the *Save* button in the lower right corner of the Schema Editor tab. This saves the current state of the schema to a set of XML-format files in subdirectory *dispatch/config* of the depot directory (slice) in the AccuRev repository :

- Contents of the *Schema* subtab: *schema.xml*
- Contents of the *Layout* subtab: *layout.xml*
- Contents of the *Lists* subtab: *lists.xml*
- Contents of the *Relationship Types* subtab: *relation_types.xml*

- Contents of the *Validation* subtab: *logic.xml*
- Contents of the Change Package Results section of the *Change Packages* subtab: *cpk_fields.xml*
- Contents of the Change Package Triggers section of the *Change Packages* subtab: *cpk_promote_queries.xml*

Schema Tab

The Schema tab of the Schema Editor contains a table that details all the data fields defined for AccuWork in the current depot.



3pty ITS Key: rallyID

Name	Type	Label	Report Width
actTime	Timespan	Dev Actual Time	10
affectedFiles	Text	Affected Files	10
assignedTo	User	Assigned To	15
Attachments	Attachments	Attachments	10
Automated	Choose	Automated	10
category	Choose	Category	10
ccList	Text	CC	10
closedBy	User	Closed By	15
closedInRelease	List	Closed in Release	15
comments	Log	Comments	40
dateClosed	Timestamp	Date Closed	15

☐ Include deactivated trees

If you are not using the repository's default schema, AccuWork initializes the Schema Fields table with two fields.

- **issueNum**: An integer-valued field that records the position of the issue record in the depot. This number is assigned when a user creates the record (i.e. at the first *Save* on the Issue form), and it never changes.
- **transNum**: An integer-valued field that records the transaction number of the most recent update to the issue record. The transaction appears in the depot's transaction log -- the same log that records AccuRev transactions, such as *keep* and *promote*. The type of the issue-record-update transaction is **dispatch**.

Notes:

issueNum vs. transNum

A record's *issueNum* value never changes, but its *transNum* value changes each time a user *Save*'s the record. *issueNum* and *transNum* are indexes into two different databases.

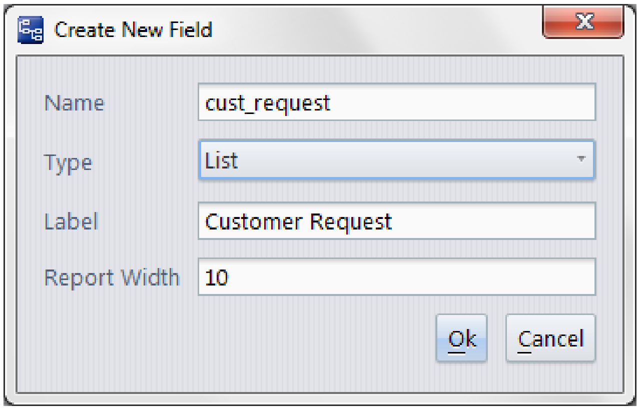
AccuRev/AccuWork integration

The change package-level integration between AccuRev and AccuWork also updates issue records, and so changes the value of the *transNum* field. See *Change-Package-Level Integration between AccuRev and AccuWork* on page 400.

Adding Fields to the Schema

You can define any number of additional fields in the schema. Follow these steps for each new field:

1. Click the *Add* button at the bottom of the Schema subtab.
2. Fill in the *Create New Field* window that appears:



Create New Field

Name: cust_request

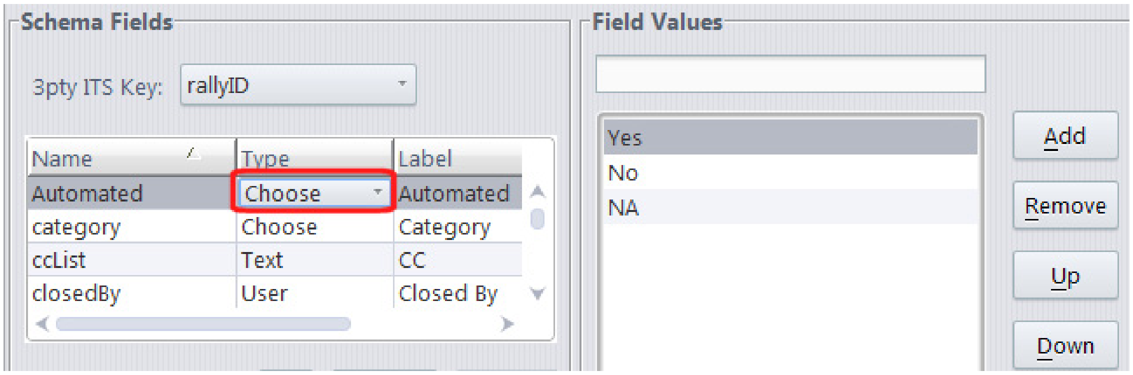
Type: List

Label: Customer Request

Report Width: 10

- *Name*: The official *field-name* of the field. AccuWork users don't ever need to know this name -- they know the field by its label. The field-name must not contain any SPACE characters. Validations must be expressed in terms of the field's name, not its label.
- *Type*: One of the data types supported by AccuWork. See *Data Types* on page 380 for more information.
- *Label*: The field-label character string that identifies the field on the Issue form. A field-label can contain SPACE characters (e.g. *Last Name*).
- *Report Width*: An integer that determines the relative width of the field in the HTML table created when the user clicks *Export* on the Issue form of an individual issue record.

3. Click *Ok* to close the *Create New Field* window.
4. In the Field Values box to the right of the Schema Fields table, specify additional information about the field. The kind of information required varies with the data type.



3pty ITS Key: rallyID

Name	Type	Label
Automated	Choose	Automated
category	Choose	Category
ccList	Text	CC
closedBy	User	Closed By

Yes
No
NA

Repeat the steps above as often as required to create new fields. Your field definitions are not saved until you click the *Save* button in the lower-right corner of the Schema Editor tab. You cannot save your work until you place at least one field in the Issue form (Layout subtab).

Data Types

Each field you define in the *Create New Field* window must have one of the AccuWork data types listed in the table below.

Field Type	Possible Values	Required Additional Information in "Field Values" Box
Text	Any character string. For multiple-line fields, the string can include line-terminators.	*Height*: number of lines displayed for the field in the Issue form (default: 1). For multiple-line fields (height > 1), the Issue form includes an expand/contract button to increase/decrease the number of lines displayed. *Width*: relative width of field in the Issue form. **Note**: Text fields can be configured to support rich text. See *Enabling Rich Text for Text and Log Fields* on page 384 for more information.
Choose	One of the strings specified in the Field Values box for this field.	A set of strings. In the Issue form, a list-box containing this set of strings is offered to the user. See *Defining Multiple-Choice Lists* on page 383 for more information on creating fields that displays lists.
Timestamp	An AccuRev timestamp.	Granularity (year, month, day, hour, minute, or second). In the Issue form, the user fills in fields that indicate a time, to the granularity you specify here.
User	One of the principal-names in the user registry maintained by the AccuRev server.	None. In the Issue form, a list-box is offered to the user, containing the names of all registered AccuRev users.
Timespan	A numeric value, indicating a number of hours. Decimal values (e.g. 4.5) are allowed.	None.
List	One of the strings specified in the definition of a particular named list.	The name of an existing list, defined on the ***Lists*** subtab. In the Issue form, a list-box containing the set of strings defined in the named list is offered to the user. See *Defining Multiple-Choice Lists* on page 383 for more information on creating fields that displays lists.

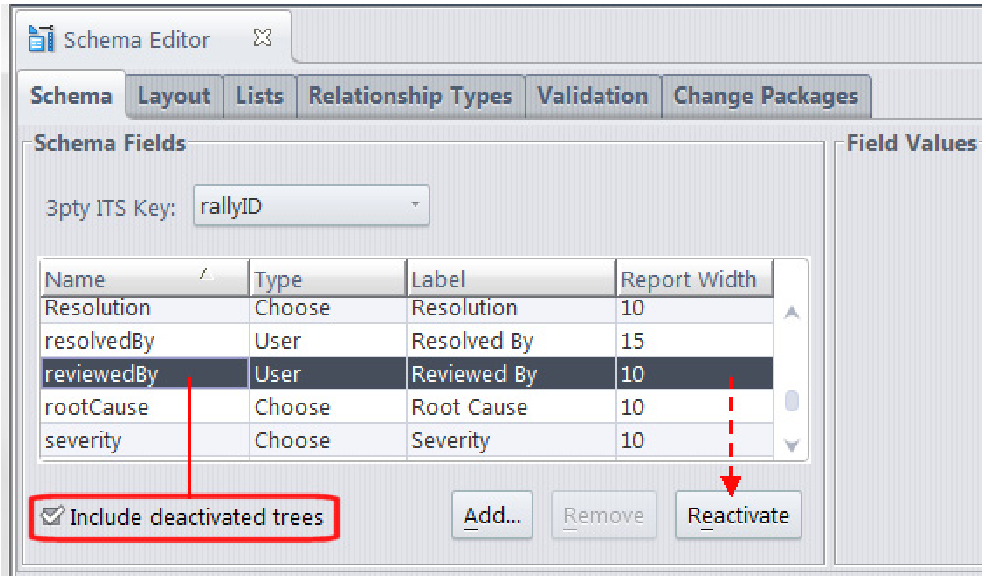
Log	Any character string (variant of text field type)	*Height*: number of lines displayed for the field in the Issue form (default: 1). For multiple-line fields (height > 1), the Issue form includes an expand/contract button to increase/decrease the number of lines displayed. *Width*: relative width of field in the Issue form. *Enable Rich Text*: Log fields can be configured to support rich text. See *Enabling Rich Text for Text and Log Fields* on page 384 for more information. *Add new text to*: Where in the log you want new text entered. Choices are at the beginning or at the end.
Attachments	A set of attachment definitions.	*Height*, *Width*: the height (number of lines) and width (approximate number of characters) of the edit-form field that lists the attachment definitions.
Relationship	A set of issue records.	See *Defining Issue Record Relationships* on page 384 and *Relationship Types Tab* on page 391 for more information.
Internal	A positive integer.	None. You cannot create a field with this data type; it is used only by the built-in fields *issueNum* and *transNum* .

Removing a Field / Restoring a Field

To remove an existing field (except for *issueNum* and *transNum*), select it and click the *Remove* button. The field disappears from the Schema tab and can no longer be used in the Issue form. But any data stored in existing issue records is preserved.

When you remove a field from the schema, AccuWork checks whether the field is used in the Issue form. If so, it removes the field from the Issue form

You can restore a removed field to the schema:



3pty ITS Key: rallyID

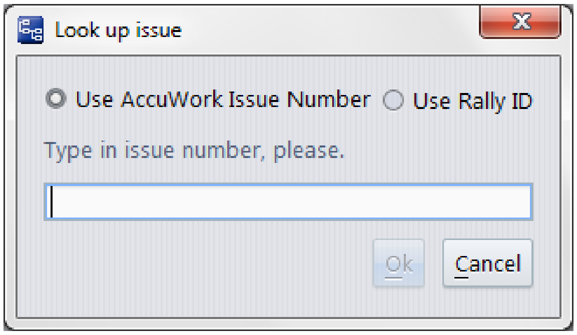
Name	Type	Label	Report Width
Resolution	Choose	Resolution	10
resolvedBy	User	Resolved By	15
reviewedBy	User	Reviewed By	10
rootCause	Choose	Root Cause	10
severity	Choose	Severity	10

1. Check the *Include deactivated trees* checkbox. All removed fields appear in the list, with a gray background.

2. Select the field to be restored, and click the *Reactivate* button.

Specifying the Lookup Field

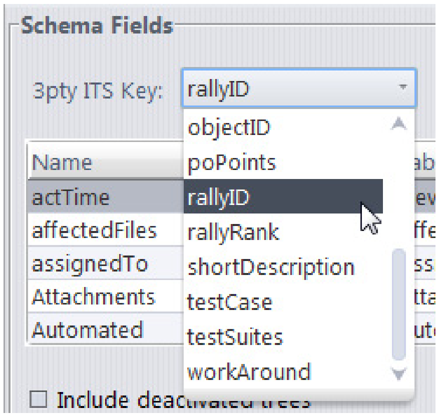
The ***Issues > Look Up*** command prompts the user to enter a value, then uses that value to locate an issue record.


Look up issue
● Use AccuWork Issue Number ○ Use Rally ID
Type in issue number, please.
Ok
Cancel

By default, this lookup operation uses the **issueNum** field (whose user-visible label is **Issue**).

You can configure another field to be used for issue-record lookups, using the ***3pty ITS Key*** listbox. (This feature is designed to facilitate AccuBridge and AccuSync integrations with third-party issue-tracking systems.)

Select another field from the listbox, and save the schema.



Thereafter, the ***Issues > Look Up*** command will prompt for a value to be matched against the specified field.

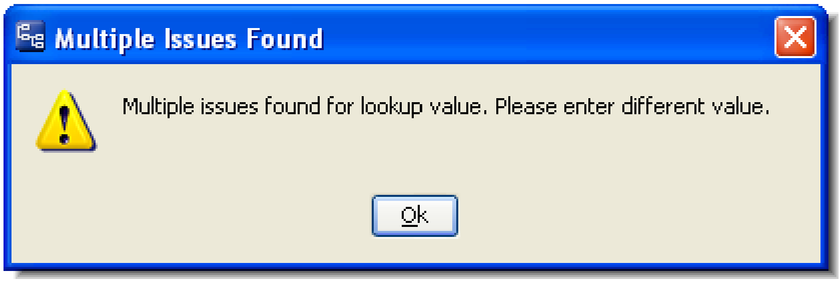
Notes:

Value-matching on lookups

The ***Issues > Look Up*** command requires an exact match between the user's entry and the value in the issue-record field. Exception: upper-case/lowercase differences are ignored. This can cause problems if the lookup field is a Text field.

Uniqueness of lookup values

AccuRev guarantees uniqueness of the ***issueNum*** field value only. It allows duplicate values in all other fields. Be sure to specify an alternative lookup field whose values are rendered unique by some other mechanism. If a user performs a lookup on a non-unique value, an error occurs:



Multiple issues found for lookup value. Please enter different value.

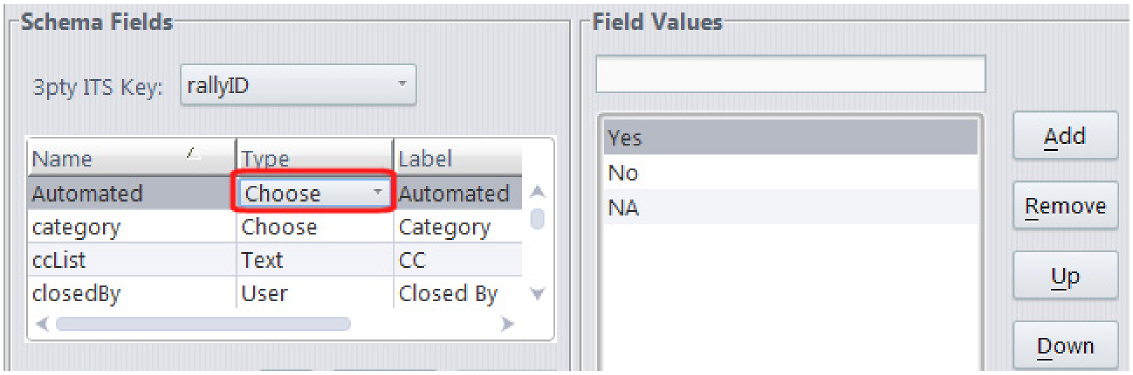
Integrating Configuration Management and Issue Management

AccuWork records each promoted version in the issue record, in a special section named *Changes*. This section is maintained automatically by AccuWork -- you don't need to define any fields to enable this change package integration. See *Change-Package-Level Integration between AccuRev and AccuWork* on page 400 for more information.

Defining Multiple-Choice Lists

The *choose* and *list* data types are similar:

- For a *choose* field, the set of possible values -- an ordered list of strings -- is part of the individual field definition. You enter the possible-values list in the Field Values box for that field.

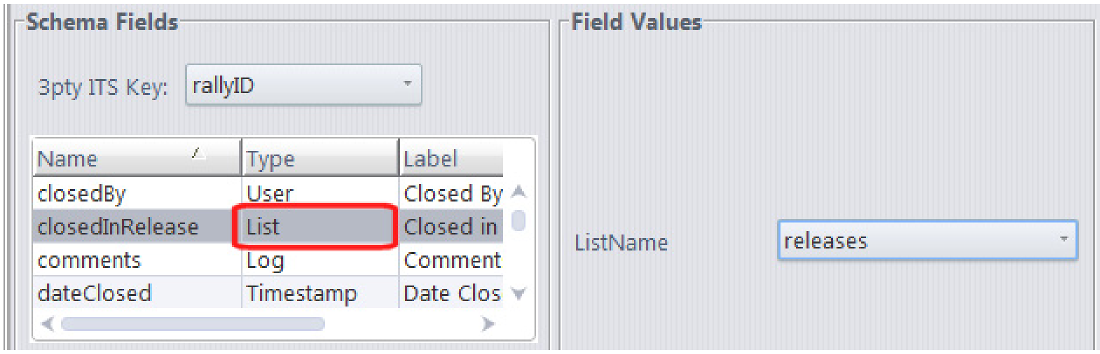


3pty ITS Key: rallyID

Name	Type	Label
Automated	Choose	Automated
category	Choose	Category
ccList	Text	CC
closedBy	User	Closed By

Yes
No
NA

- For a *list* field, the set of possible values is also an ordered list, but it is *not* part of the individual field definition. In the Field Values box, you specify the name of one of the lists created on the ***Lists*** subtab. Any number of *list* fields in AccuWork (within a single depot) can use the same named list.



3pty ITS Key: rallyID

Name	Type	Label
closedBy	User	Closed By
closedInRelease	List	Closed in
comments	Log	Comment
dateClosed	Timestamp	Date Clos

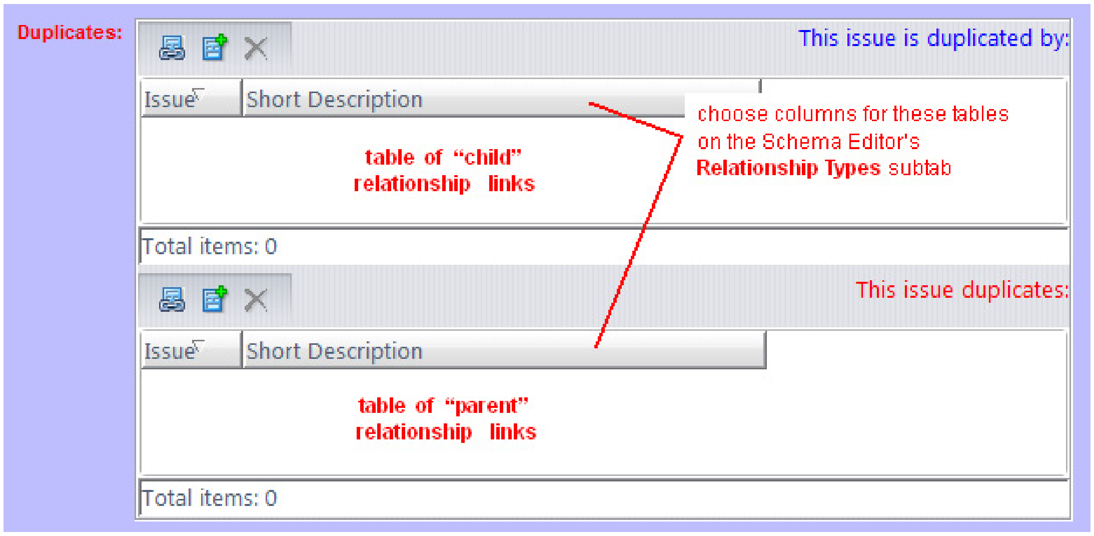
ListName: releases

The mechanics of defining the ordered list is similar in the "local" case (for an individual *choose* field) and in the "global" case (on the Lists subtab, for use by any number of *list* fields). On the Lists subtab, you must supply a ListName for the list; for an individual *choose* field, the possible-values list doesn't need or have a name.

Defining Issue Record Relationships

When you create a field of type *Relationship*, you must select the *Duplicate*, *Subtask*, or *Dependency* relationship type for the field. On an Issue form, the field's edit-widget is a pair of tables. The upper table displays "inward" or "child" relationship links -- that is, links *from* other issue records to the current record. The lower table displays "outward" or "parent" relationship links -- that is, links *to* other issue records from the current record. (Multiple-link chains are not allowed -- each issue record can be related to others by child links or parent links, but not both.)

The figure below show how a *Duplicate* relationship field appears in an Issue form. A relationship field of type *Subtask* or *Dependency* appears similarly.



Duplicates:
This issue is duplicated by:
Issue
Short Description
choose columns for these tables on the Schema Editor's Relationship Types subtab
table of "child" relationship links
Total items: 0
This issue duplicates:
Issue
Short Description
table of "parent" relationship links
Total items: 0

Enabling Rich Text for Text and Log Fields

AccuWork supports rich text for fields with data types of *Text* and *Log*. Fields for which you wish to enable rich text must have a Height of at least 1.

When you enable rich text for a field, AccuWork:

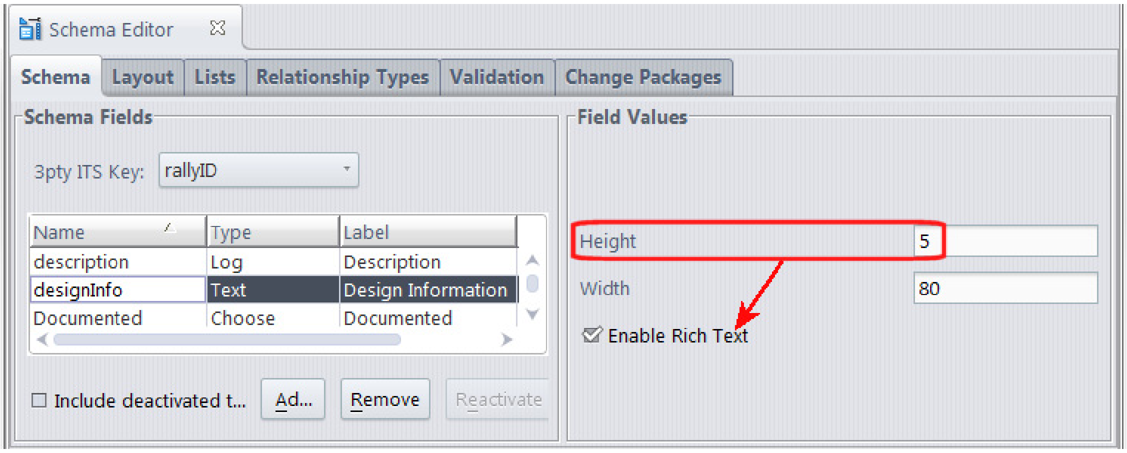
- Displays tools for working with rich text
- Renders formatted text (and not tagged HTML, for example)

See *Rich text fields* on page 298 for more information and examples.

To enable rich text for an AccuWork field:

1. Close any open issues.
2. On the Schema tab, select the field for which you want to enable rich text.

   Tip: Only fields with a data type of *Text* or *Log* support rich text.
3. In the Field Values pane:
   1. Ensure that the field's Height is 1 or more.
   2. Click the **Enable Rich Text** check box.
   3. Click the **Save** button.



3pty ITS Key: rallyID

Name	Type	Label
description	Log	Description
designInfo	Text	Design Information
Documented	Choose	Documented

☐ Include deactivated t...

Height 5

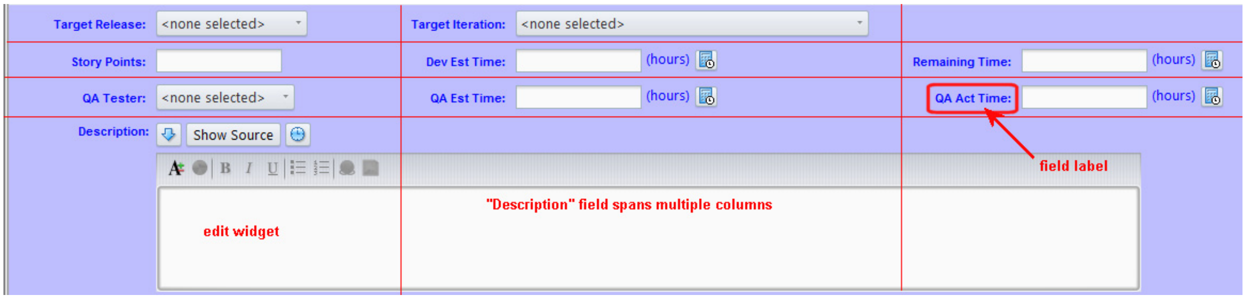
Width 80

☑ Enable Rich Text

4. Close the Schema Editor tab.

Layout Tab

An Issue form consists of field-labels and corresponding edit-widgets, arranged in rows and columns. The field-labels come from the Labels column of the Schema Fields table (Schema subtab). The edit-widget for a field depends on its data type and, in some cases, on the additional Field Value information.



Target Release: <none selected>

Target Iteration: <none selected>

Story Points:

Dev Est Time: (hours)

Remaining Time: (hours)

QA Tester: <none selected>

QA Est Time: (hours)

QA Act Time: (hours)

Description:

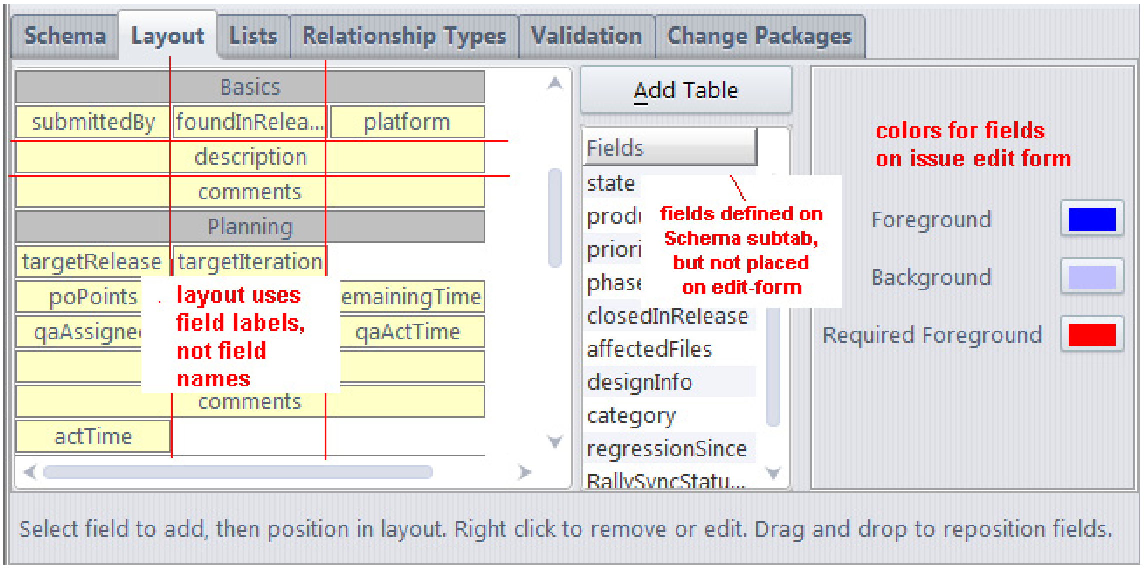
field label

"Description" field spans multiple columns

edit widget

Contents of the Layout Subtab

The Layout subtab provides a drag-and-drop canvas on which you design the Issue form's rows and columns. On this canvas, each field-label/edit-widget pair is represented by a yellow box.



layout uses field labels, not field names

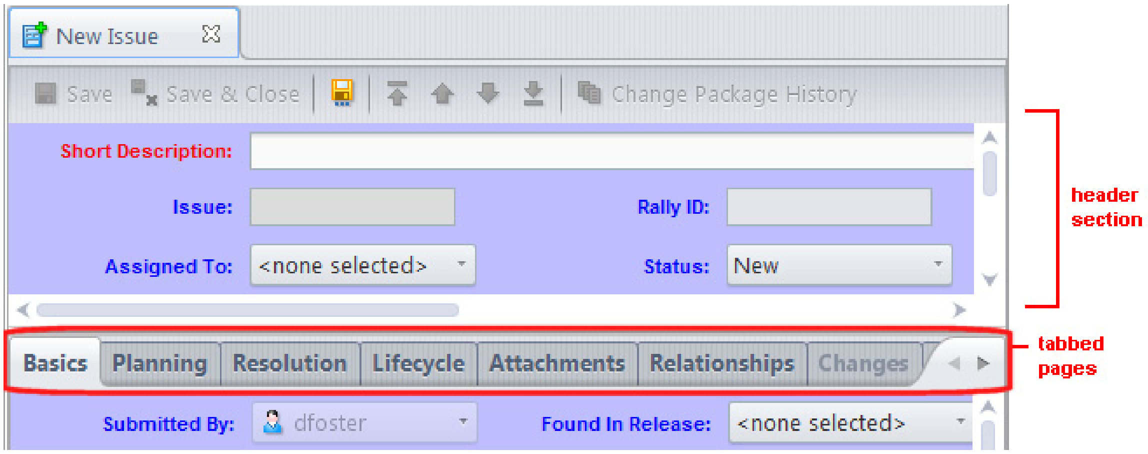
fields defined on Schema subtab, but not placed on edit-form

colors for fields on issue edit form

Select field to add, then position in layout. Right click to remove or edit. Drag and drop to reposition fields.

As the screen-shot annotations indicate, the yellow box contains a field-name (e.g. **actTime**), not a field-label (**Actual Time**). On the Issue form, the field-label and the corresponding edit-widget will appear, side by side, at this position. You can also expand a yellow box to make it span two or more columns.

You can organize the fields into multiple "tables", each of which defines a separate tabbed page in the Issue form. One of the tables can define a header section, which always appears on the Issue form, no matter which tabbed page is currently visible.



Short Description:

Issue:

Rally ID:

Assigned To: <none selected>

Status: New

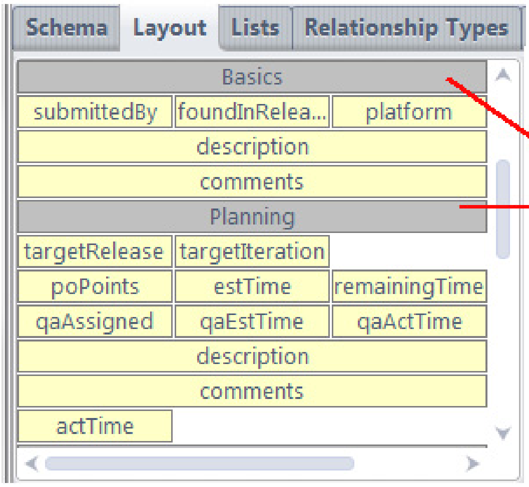
Submitted By: dfoster

Found In Release: <none selected>

header section

tabbed pages

The layout for a multiple-page Issue form looks like this:



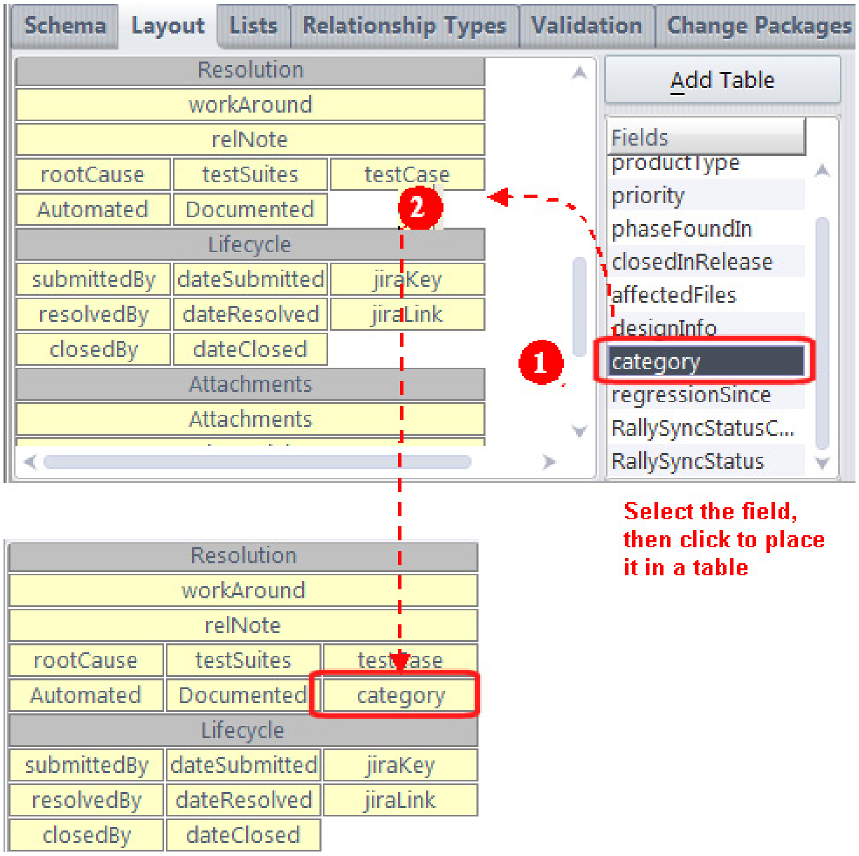
each "table" defines a separate tabbed page on edit form

Form Layout Operations

This section lists the edit-form design operations available on the Layout subtab.

Adding a field to the Issue form

The *Fields* list-box offers all fields that do not currently appear in the Issue form. Click to select a field in this list-box. Then, click at the empty location on the canvas where you want to place the field.



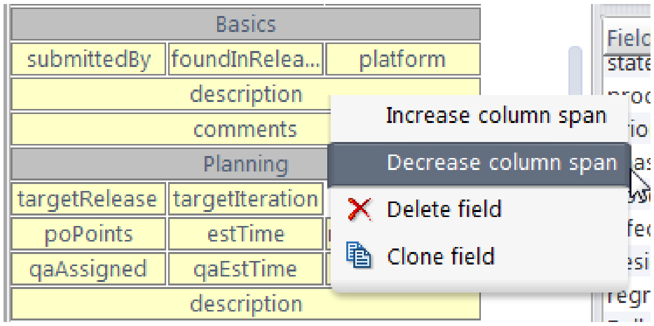
Select the field, then click to place it in a table

Arranging fields into rows and columns

Drag-and-drop the yellow boxes to the desired location on the canvas. Drop a box on top of another one to push it rightward or downward.

Changing column-spanning

Right-click a yellow box, and select one of the spanning operations from the context menu.



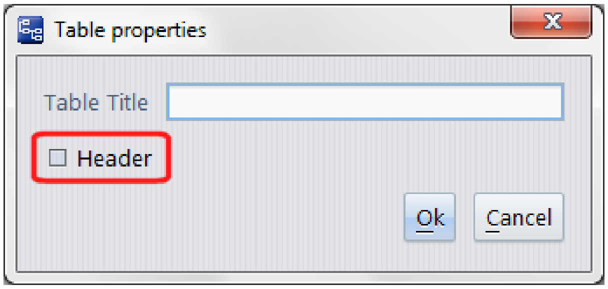
Creating a new page in the Issue form

Click the *Add Table* button above the *Fields* list-box. Fill in the Table Title field in the dialog box that appears, then click *OK* . This title will appear on the page's tab in the Issue form.

A gray box with the specified title appears at the bottom of the layout. Fields that you place below this box will appear on the new page of the Issue form.

Creating a header table for the Issue form

When creating a new page for the Issue form (see above), check the *Header* checkbox if you want it to be the header table.



An Issue form can have at most one header table. If another table is currently selected as the header table, it is automatically deselected. Its fields always appear at the top of the Issue form, even if it is not the topmost table on the Layout tab.

You can also redefine an existing page to be a header table: right-click the gray box and select *Properties*. All fields in the header table will appear at the top of the database's Issue form, above every tabbed page.

Renaming a page

Right-click on the gray box, select *Properties* , and change the entry in the Table Title box.

Arranging fields into multiple pages

Drag-and-drop the yellow boxes (fields) and the gray boxes (pages). Fields that you place below a gray box will appear on the corresponding page of the Issue form.

Removing a field or page

Right-click the yellow box (field) or gray box (page), and select *Delete* from the context menu. Deleting a page doesn't delete the fields currently on the page -- it just deletes the gray box, effectively merging the fields into the page above. **Note:** You cannot delete the first gray box; an Issue form must have at least one page.

Setting the Issue form's colors

Click one of the colored buttons at the right side of the Layout subtab. Then select a color from the *Pick a color* window. You can set these colors:

- *Foreground*: the color of the Issue form's field-labels.
- *Required Foreground*: the field-label color for fields where a value must be specified. Required fields are defined on the ***Validations*** subtab.
- *Background*: the color of the Issue form's background.

Lists Tab

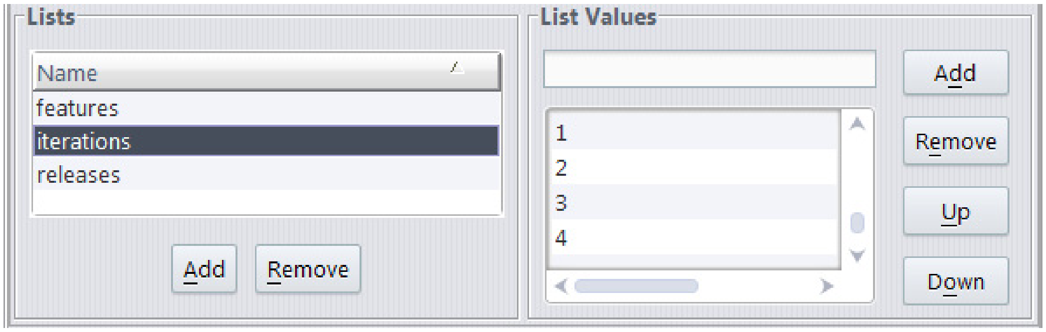
A field whose data type is **List** can have any of the values in a predefined list. Any number of fields can get their values from the same list.

The Lists subtab maintains the set of predefined lists for AccuWork within a given depot .

Lists Subtab Layout

The Lists subtab has two sections:

- In the "Lists" section, you create and delete named lists.
- In the "List Values" section, you specify the values in each list.

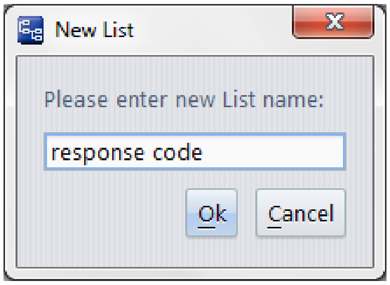


Lists Subtab Operations

This section describes the edit-form design operations available on the Layout subtab.

Creating a New List

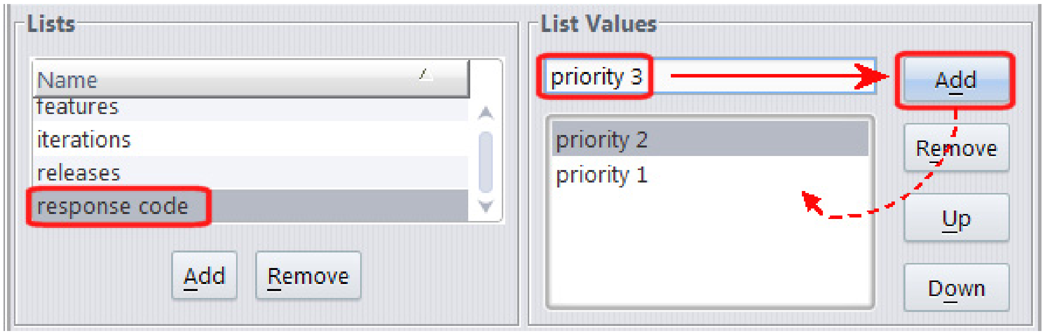
Click the ***Add*** button in the Lists section, and type a name into the popup window.



response code

Specifying the Values for a List

Select one of the list names in the Lists section. Then, in the List Values section: type a name in the input field at the top, and click the ***Add*** button.



Name
features
iterations
releases
response code

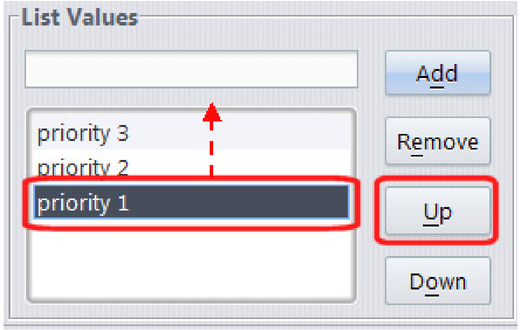
priority 3

priority 2
priority 1

The new name is added to the listbox below the input field, allowing you to type another name and click ***Add*** again.

Reordering the Values in a List

Select a name in the listbox showing the current values, and click the ***Up*** or ***Down*** button.



priority 3
priority 2
priority 1

The order of the names in this listbox determines the order in which they will appear in the edit-widget on an Issue form.

Removing a List

In the Lists section: select a name, then click the ***Remove*** button.

Renaming a List

There is current no provision for renaming an existing list.

Relationship Types Tab

An issue record can be related to one or more other issue records, through a variety of relationship types:

- **Duplicate**: You can specify that issue record B duplicates issue record A, so that no work need be done on B. (Perhaps the same bug was reported twice.) Any number of records (B, C, D, ...) can duplicate a given record (A). The default AccuWork schema requires that a Duplicate relationship be established when an issue record's State field is set to the value **Duplicate**.
- **Dependency**: You can specify that issue record B depends on issue record A (for example, from a project management viewpoint). AccuRev imposes no further semantics on such dependencies; this is a good application for user-defined scripts or practices.

  Note: This relationship is independent of change package dependency relationships among issue records, which are maintained automatically by AccuRev.
- **Subtask**: You can specify that issue record B defines a subtask of the work to be performed for issue record A. In this case, AccuRev considers issue record A to be "in" a particular stream only if issue record B is "in" that stream, also.
- **Track**: This relationship type is used internally by AccuWork to track change package gaps.

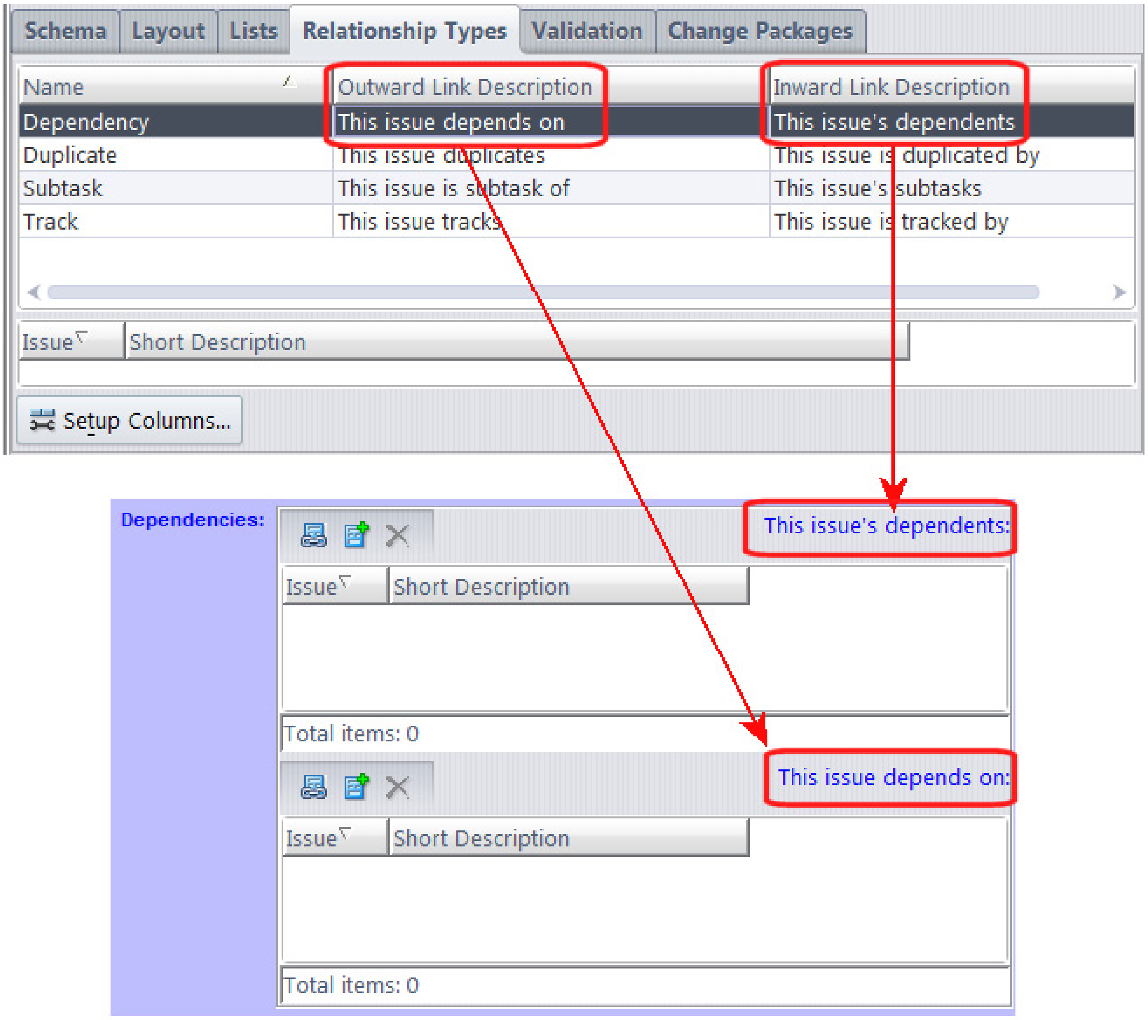
The edit-widget for a field whose data type is **Relationship** is a table. It displays selected fields from all the other issue records that are related, through a particular relationship type, to the current record.

On this subtab, you specify -- for each relationship type -- which fields from a set of related issue records are to be displayed in a **Relationship** field.

Relationship Types Tab Layout

The Relationship Types subtab contains two sections:

- The upper section lists the several relationship types, along with the captions that appear in the edit-widget (a pair of tables) for a **Relationship** field of that type. (The captions cannot be edited.)
- The lower section specifies the set of fields to be displayed from each related issue record, along with their order and display widths.



Name	Outward Link Description	Inward Link Description
Dependency	This issue depends on	This issue's dependents
Duplicate	This issue duplicates	This issue is duplicated by
Subtask	This issue is subtask of	This issue's subtasks
Track	This issue tracks	This issue is tracked by

Working on a Relationship Types Tab

1. In the upper section, click to select one of the relationship types.
2. In the lower section, click the ***Setup Columns*** button to bring up a dialog box in which you select a set of database fields, to be displayed as columns in a Relationship field's edit-widget. (See *Set Up Columns* on page 321.) You can also specify the order of the fields.
3. After closing this dialog box, you can adjust column widths and change the order of columns (see *Working with Tables* on page 14).

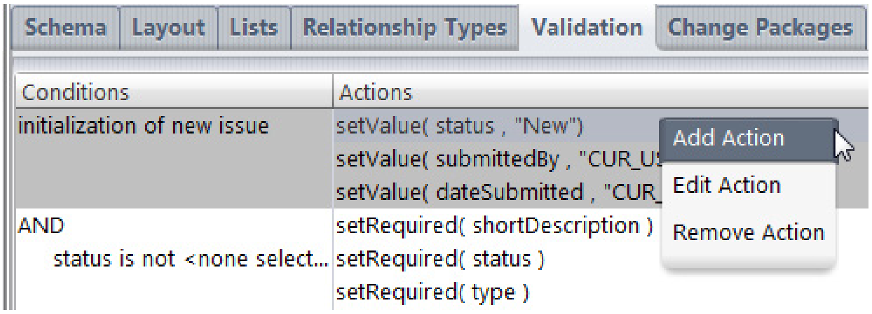
Validation Tab

Users of AccuWork create and modify issue records through the Issue form. To increase the efficiency and accuracy of this process, you can create a set of validations to be applied as the user works with the Issue form. (Validations are sometimes called "edit checks".)

You create and maintain the set of validations using a point-and-click interface on the Validation subtab. AccuWork displays the current validations in tabular format; each one takes the form "if a certain condition is true, then perform a particular set of actions".

Initializing Field Values in a New Issue Record

The first entry in the table of validations is special. Its condition is always "initialization of new issue", so that its actions are performed exactly once: between the time the user invokes the *New Issue* command and the time the new Issue form appears.

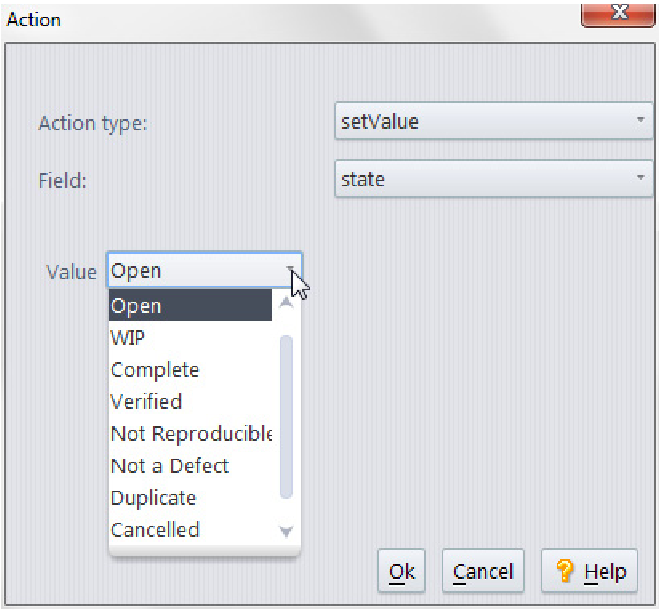


Conditions	Actions
initialization of new issue	setValue(status , "New") setValue(submittedBy , "CUR_U setValue(dateSubmitted , "CUR_

AND
status is not <none select...
	setRequired(shortDescription)
	setRequired(status)
	setRequired(type)

To define an action, right-click in the *Actions* column, then select *Add Action* from the context menu. An *Action* window appears, in which you define the action. For initialization of a new issue record, the only appropriate action type is *setValue*. (**Note:** Some releases of AccuWork allow you to include setRequired and setChoices actions in a record initialization. Such actions will be ignored when a new record is initialized.)

After setting the action type to *setValue*, select a field to be initialized and specify the initial value. The *Value* edit-widget adjusts to the selected field. In the example below, the ***data type*** of the *state* field is **choose**, so the list-box offers the field's predefined choices as the initial value.



Action type: setValue

Field: state

Value Open

Conditional Validations

The setting of initial field values is an unconditional validation: it happens every time a *New Issue* command is invoked. All other validations are conditional: a certain set of actions are performed if, and only if, a certain condition is met.

The unconditional setting of initial field values occurs just once; but the conditional validations are performed repeatedly: when an Issue form first appears *and* each time the user changes any field value. Each repetition involves:

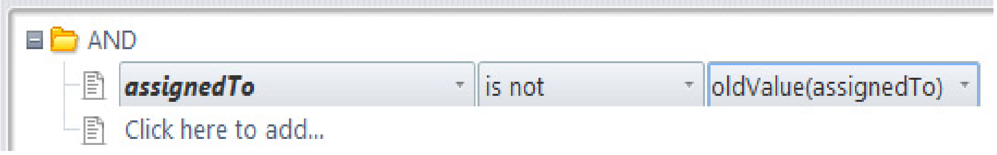
- Clearing the "required" status of all fields.
- Performing all validations (except for that first one: "initialization of new issue"). If a validation's condition is true, the corresponding actions are performed.

Specifying the Condition

To create a new conditional validation -- that is, a new row in the table -- right-click anywhere in the Conditions column, then select *Add Condition* from the context menu. Then proceed to construct the "if" clause. The same context menu provides the ***Edit Condition*** command (for revising an existing "if" clause), and the ***Remove Condition*** command (for removing a conditional validation).

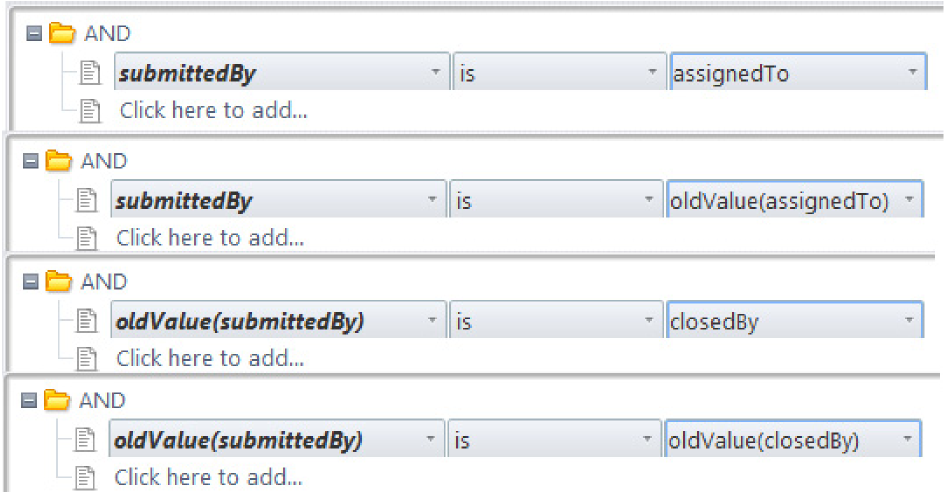
Specifying the "if clause" of a conditional validation is very much like specifying an AccuWork database query (see *Working in the Query Editor* on page 317). But there are some differences. In a validation condition, you can:

- Compare the current value of a field with its "old" value -- that is, compare the value currently displayed for the field vs. the value stored in the AccuWork database by the most recent *Save*.



AND
assignedTo | is not | oldValue(assignedTo)
Click here to add...

- Compare the (current or old) value of a field with the (current or old) value of another field in the same record.



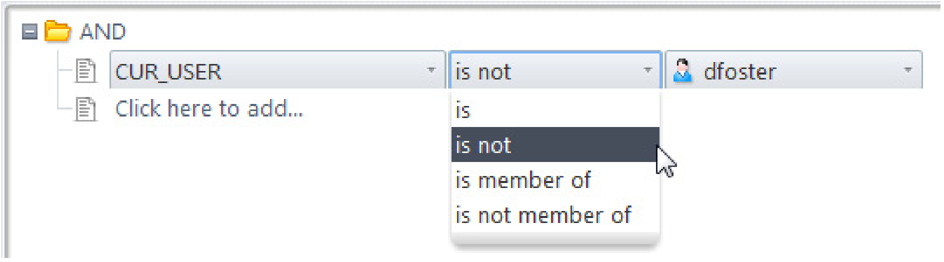
AND
submittedBy | is | assignedTo
Click here to add...

AND
submittedBy | is | oldValue(assignedTo)
Click here to add...

AND
oldValue(submittedBy) | is | closedBy
Click here to add...

AND
oldValue(submittedBy) | is | oldValue(closedBy)
Click here to add...

- Test the AccuRev user identity or group membership of the person using the Issue form (*CUR_USER*).



AND

CUR_USER is not dfoster

Query conditions cannot make such field-to-field comparisons; they can only compare field values to literal values.

Specifying the Actions

Each conditional validation can include any number of actions. You create an action by right-clicking in the Actions column of the validation, and selecting *Add Action* from the context menu. This displays a window in which you define the action.

After you've created one or more actions for a conditional validation, you can use the same context menu to revise or delete individual actions.

The following sections describe the actions that you can define to be performed if the validation condition is true. Each validation can invoke any number of actions, of any type.

- Setting a field value (*setValue*)
- Revising the choices for a "choose" field (*setChoices*)
- Requiring a value to be entered in a field (*setRequired*)
- Setting permissions on all or part of the issue record (*setIssuePermission*, *setTabPermission*, *setFieldPermission*)
- Requiring change set entries (*setChangesRequired*)
- Requiring a relationship with another issue record (*setParentRelationshipRequired*, *setChildRelationshipRequired*)

Setting a Field Value

In addition to initializing field values (see Initializing Field Values in a New Issue Record), you can set the values of fields while the user is working with the record, based on certain conditions. For example, if the user enters *ColorStar* in the *program_name* field, a validation could automatically set the *fix_priority* field to *high* . (Management has mandated rapid improvement in the robustness of the ColorStar application.)

To define a *setValue* action for a validation condition:

1. In the Action window, select *setValue* as the action type, and select the field to be set.
2. The *Value* edit-widget adjusts to the selected field, just as it does when you're initializing a field in a new issue record.

For a "text", "log", or "timespan" field, you enter a text string as the value; for a "date" field, you specify a date in the standard way; for other field types, you use a list-box to specify any of the field's predefined values.

Revising the Choices for a "choose" Field

The definition of each field of type "choose" includes an ordered list of strings. The user fills in this field by choosing one of the strings from a list-box.

In a validation, a *setChoices* action can change the "choice list" that the user will see when he opens the list-box: The changes to the choice list can include:

- Removing one or more of the existing choices
- Changing the order of the choices

You cannot add new strings to the choice list with a *setChoices* action. The changes made by a *setChoices* action are temporary: they last only until the next *setChoices* action on the same field, or until the user closes the Issue form. When an Issue form is opened on another issue record (or subsequently on this issue record), the user will see the field's original choice list, as defined on the *Schema* tab.

To define a *setChoices* action for a validation condition:

1. In the Action window, select *setChoices* as the action type, and use the *Field* list-box to select one of the database's fields of type "choose".
2. Use the controls at the bottom of the *Action* window to define the temporary change to the choice list:

Requiring a Value to be Entered in a Field

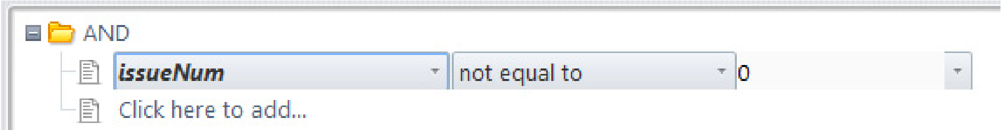
Like many other programs that process fill-in-the-blanks forms, AccuWork can treat certain fields as required fields -- the user must specify a value in each such field before AccuWork will create or update an issue record. (For a "choose", "list", or "user" field, the value must not be the <*none selected*> placeholder.)

AccuWork affords you great flexibility in controlling required fields. A field's "required" status is not part of the basic database schema. Instead, it is controlled by the conditional-validation facility. Thus, AccuWork repeatedly redecides whether a field is required: when the Issue form first appears *and* each time the user changes any field value on the Issue form. Whenever the user clicks the *Save* button, AccuWork uses the current set of required fields to allow or disallow the "save" operation.

To define a *setRequired* action for a validation condition:

1. In the Action window, select *setRequired* as the action type.
2. Select the field to be required.

If you want a field to be required in all circumstances, use the *setRequired* action along with a condition that's always true. For example, you can exploit the fact that every issue record has a positive integer as its issue number:



AND

issueNum | not equal to | 0

Setting Permissions on All or Part of the Issue Record

The *setIssuePermission*, *setTabPermission*, and *setFieldPermission* actions are essentially similar: they restrict the editability of some component of the issue record:

- ***setIssuePermission*** makes the entire issue record read-only.
- ***setTabPermission*** makes a particular page (tab) of the Issue form read-only.
- ***setFieldPermission*** makes a particular field of the issue record read-only.

There are two attributes that you can assign to these actions:

- ***ro***— Make read-only for all situations.
- ***auto***— (AccuRev Workflow only.) Make read-only for end-users, but allow AccuRev to make changes.

The "read-only" (*ro*) permission affects the user's current access to the issue record; the "read-only" status is not stored in the repository as part of any issue record. For example, user *allison* might find that she cannot change any fields on the Assignment page of any issue record, because of this validation:

This restriction affect *allison* 's access to issue records; other users' access remains unaffected. And if this conditional validation is subsequently removed, *allison* will regain access to the Assignment page of all the issue records.

Note that setting read-only permission on an issue record will affect the behavior of AccuRev commands such as ***cpkremove*** and ***promote***, in addition to restricting the user's ability to edit the issue.

The “automatic” (***auto***) permission is similar to ***ro***, but in AccuRev Workflow, AccuRev can perform actions that end-users are prevented from making as described above. For more information about AccuRev Workflow-specific details, see the *AccuRev Web Interface User’s Guide*.

We recommend that each change to the schema be tested thoroughly to prevent unexpected side-effects to your organization's processes.

Requiring Change Package Entries

The *setChangesRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the issue record's change package.

> **Note:** Be careful *not* to specify this action with the "initialization of new issue" condition (see *Initializing Field Values in a New Issue Record* on page 393).

Requiring a Relationship with Another Issue Record

The *setParentRelationshipRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the lower table of a particular *relationship* field.

The *setChildRelationshipRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the upper table of a particular *relationship* field.

You cannot set both these requirements for the same *relationship* field, because each field can have parent relationships or child relationships, but not both.

Revising and Removing Validations and Actions

Each conditional validation consists of a condition (left column) and a set of actions (right column). To revise or remove an entire validation, right-click anywhere within the condition, and select the appropriate command, *Edit Condition* or *Remove Condition*, from the context menu.

To revise or remove an individual action from a validation, right-click on that action and select the appropriate command -- *Edit Action*, *Remove Action*, or *Add Action* -- from the context menu.

AccuWork also has two unconditional validations -- "initialization of new issue" and "initialization of cloned issue". These two validations are performed in the specified order, and you cannot remove them from the Validation tab.

Tip: You might want to use the "initialization of cloned issue" actions to reset field values for an issue record when it is cloned. For example, rather than using the original issue record's dateSubmitted value for the cloned issue (AccuWork uses all of the original issue's values by default), you might want to use a setValue action to change dateSubmitted to "CUR_DATE". See *Cloning an Issue* on page 289 for more information on this topic.

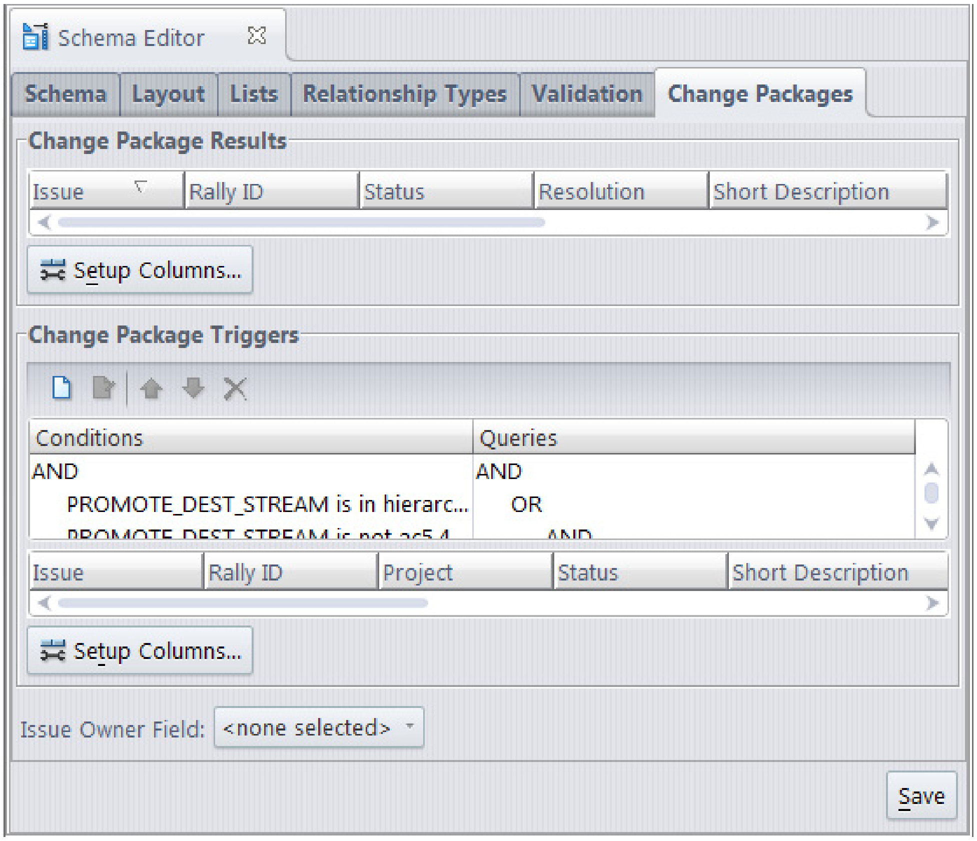
Change Packages Tab

AccuRev Enterprise supports an enhanced integration between AccuRev and AccuWork: any AccuWork issue record can act as a change package, keeping track of which versions were created to fix the bug (or implement the new feature) described in that issue record. For a thorough discussion, see the topic *Change-Package-Level Integration between AccuRev and AccuWork* on page 400.

The Change Packages subtab configures the way in which change packages are reported and updated, on a depot-by-depot basis.

Change Packages Subtab Layout

The Change Packages subtab specifies the change-package configuration for the current depot. It is organized as follows:



Issue	Rally ID	Status	Resolution	Short Description

Conditions	Queries
AND	AND
PROMOTE_DEST_STREAM is in hierarc...	OR
PROMOTE_DEST_STREAM is not ac5.4	AND

Issue	Rally ID	Project	Status	Short Description

Issue Owner Field: <none selected>

- In the ***Change Package Results*** section, you specify the format of the change-package table displayed by a *Show Issues* or *Show Diff By Issues* command. This set of fields is also used to identify issue records in the ***Issue Dependencies*** tab.
- In the ***Change Package Triggers*** section, you enable the change-package-level integration between AccuRev and AccuWork, by specifying one or more AccuWork queries, one of which will be invoked during execution of the ***Promote*** command. (The user chooses one or more of the issues records selected by the query; AccuRev adds the promoted versions to those issue record(s)' change packages.) In this section, you also specify the format of the table that displays the results of the ***Promote***-time query to the user.
- In addition, the ***Issue Owner Field*** drop-down list displayed at the bottom of the tab can be set to any AccuWork field that contains user/group data. The field specified here displays on the ***Send to Issue*** dialog, and allows the user to choose an issue belonging to another user or group.

Working in the Change Package Results Section

Click the ***Setup Columns*** button to bring up a dialog box in which you select a set of issue fields, to be displayed as table columns. See *Set Up Columns* on page 321. You can also specify the order of the fields.

After closing this dialog box, you can adjust column widths and change the order of columns. (See *Working with Tables* on page 14.)

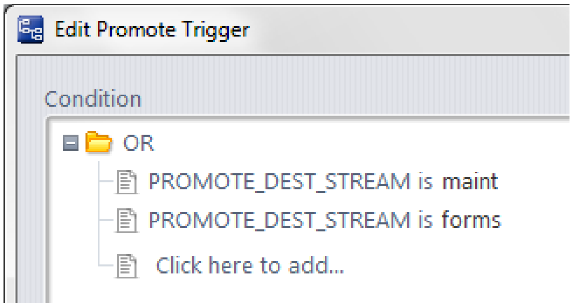
Working in the Change Package Triggers Section

This section contains a set of condition/query pairs, one of which will be executed at ***Promote***-time. If the first condition is satisfied, the first query will be executed; otherwise the second condition will be evaluated, and if it's satisfied, the second query will be executed; and so on. No error occurs if no condition is satisfied; the ***Promote*** command proceeds without triggering any change-package-level activity.

Note that there is no checkbox or button to enable the change-package-level integration. Rather, the existence of one or more condition/query pairs in this section implicitly enables the integration.

Composing a Condition

Each clause of a condition performs a test on the *Promote* destination stream (indicated by the keyword *PROMOTE_DEST_STREAM*). For example, this condition is satisfied if the user is promoting to either of the streams **maint** or **forms**:



PROMOTE_DEST_STREAM is maint

PROMOTE_DEST_STREAM is forms

The mechanics of composing the condition are similar to that of composing an AccuWork query (see *Working in the Query Editor* on page 317).

Composing a Query

Corresponding to each ***Promote***-destination condition is any AccuWork query, to selects a set of issue records. For the mechanics of composing the query, see *Working in the Query Editor* on page 317.

Formatting the Table Displayed by the Change Package Trigger

To specify the format of the table displayed by the ***Promote*** command when it invokes the change-package-level integration between AccuRev and AccuWork, click the ***Setup Columns*** button to bring up a dialog box. Use this dialog box to select fields to appear as columns, the order of the columns (fields), and the sort order of the rows (records). After closing this dialog box, you can adjust the column widths and rearrange the columns.

Change-Package-Level Integration between AccuRev and AccuWork

AccuRev provides a *change package* feature that integrates AccuRev's configuration management functionality with AccuWork's issue management functionality.

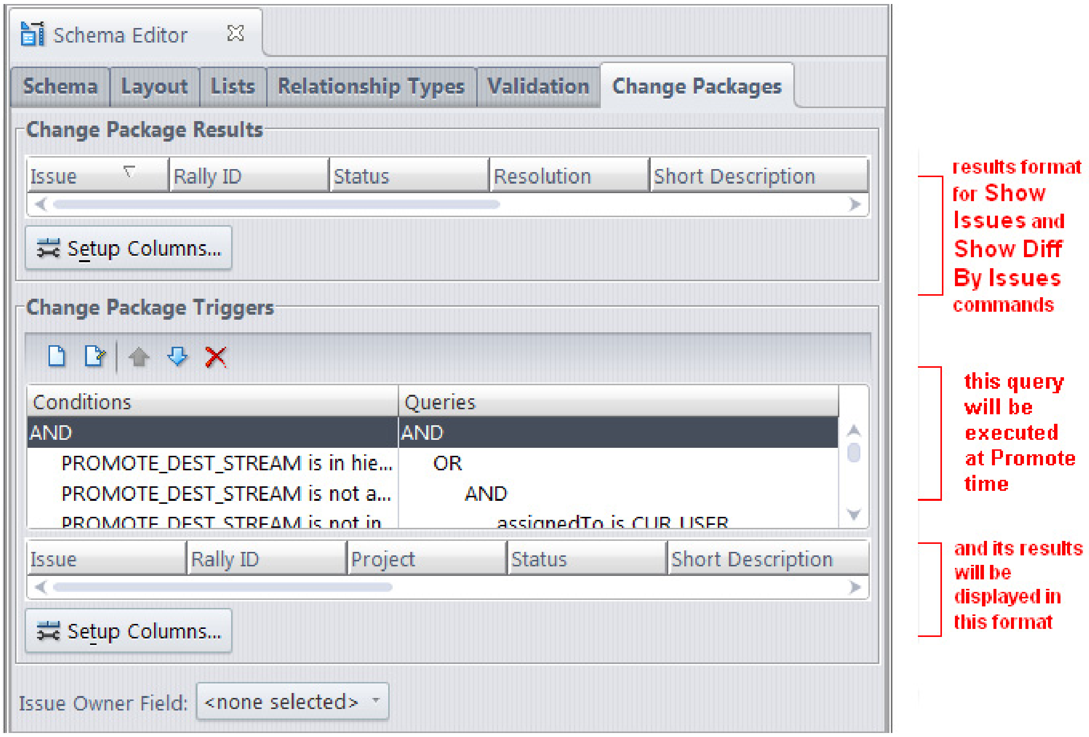
When a *Promote* command is executed a user's workspace (but not in a higher-level dynamic stream), the change-package-level integration records all the promoted versions on the ***Changes*** subtab of a user-specified AccuWork issue record.

Enabling the Integration

The change-package-level integration is enabled on a depot-by-depot basis. Open the Schema Editor for a particular depot, and go to the Change Packages subtab. Filling in the lower section, "Change Package Triggers", enables the integration for that particular depot.

The Change Package Triggers section is structured as a set of condition/query pairs. One of the queries will be selected for execution at ***Promote***-time. If the first condition is satisfied, the first query will be executed;

otherwise the second condition will be evaluated, and if it's satisfied, the second query will be executed; and so on.



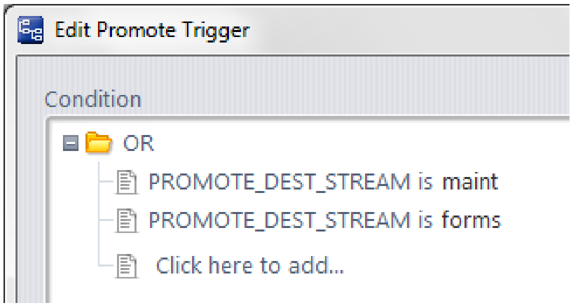
Conditions	Queries
AND	AND
PROMOTE_DEST_STREAM is in hie...	OR
PROMOTE_DEST_STREAM is not a...	AND
PROMOTE_DEST_STREAM is not in	assignedTo is CUR_USER

results format for Show Issues and Show Diff By Issues commands

this query will be executed at Promote time

and its results will be displayed in this format

Each clause of a condition performs a test on the *Promote* destination stream. For example, this condition is satisfied if the user is promoting to either of the streams **maint** or **forms**:



- OR
  - PROMOTE_DEST_STREAM is maint
  - PROMOTE_DEST_STREAM is forms

The condition must test the destination stream (*PROMOTE_DEST_STREAM*) of a *Promote* command. The query corresponding to each condition can be any AccuWork query, which selects a set of issue records.

The Change Package Triggers section also specifies the format of each query's results table -- the fields to appear as columns, the column widths, the order of the columns (fields), and the sort order of the rows (records).

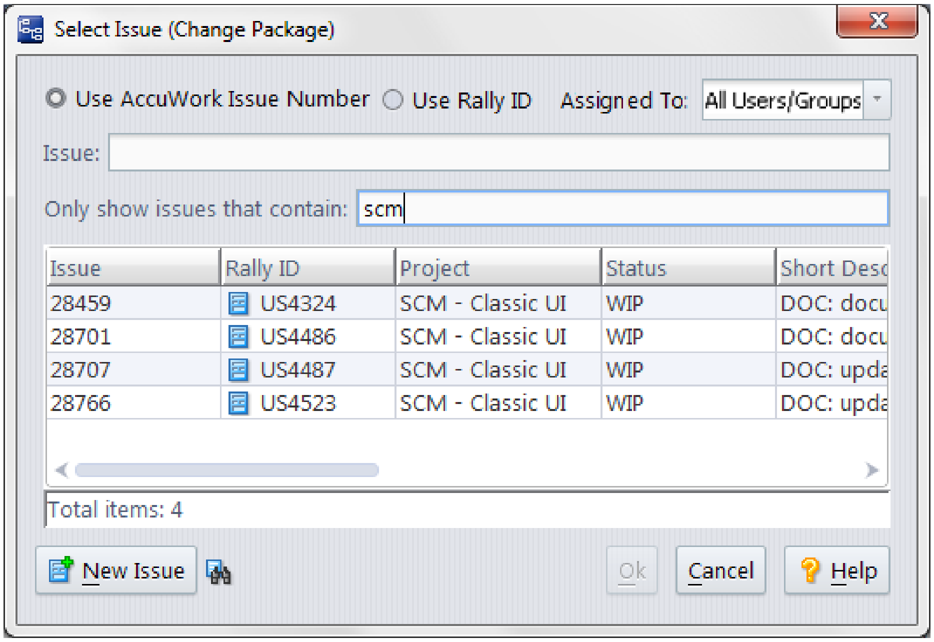
In addition, the ***Issue Owner Field*** drop-down list displayed at the bottom of the Change Packages tab can be set to any AccuWork field that contains user/group data. The field specified here displays on the ***Send to Issue*** dialog.

Triggering the Integration

Once the integration is enabled for a depot, it is triggered whenever a user performs a *Promote* command in a workspace associated with that depot:

1. One of the AccuWork queries specified in the Change Package Triggers section is executed, selecting a certain set of issue records.

2. Those records are displayed to the user in the results table format specified in the Change Package Triggers section. In this example, AccuWork has been configured to allow the user to choose from issues assigned to another user or group if necessary.



Select Issue (Change Package)

● Use AccuWork Issue Number ○ Use Rally ID Assigned To: All Users/Groups

Issue:

Only show issues that contain: scm

Issue	Rally ID	Project	Status	Short Desc
28459	US4324	SCM - Classic UI	WIP	DOC: docu
28701	US4486	SCM - Classic UI	WIP	DOC: docu
28707	US4487	SCM - Classic UI	WIP	DOC: upda
28766	US4523	SCM - Classic UI	WIP	DOC: upda

Total items: 4

Note: If you have enabled the use of third-party issue tracking system (ITS) keys in the AccuWork schema, the ***Select Issue (Change Package)*** dialog displays additional controls that let you specify whether you want to use AccuWork issue numbers or third-party keys when selecting issues to promote against. See the section called, "*Using Third-Party Keys in the AccuRev GUI*," in Appendix A (The Schema Editor (Administrators Only)) of the AccuRev Online Help Guide for more information.

3. The user selects one or more of the issue records, or types an issue identifier into the ***Issue*** # field.

   Promoting to Issues Other than Those Shown -- The dialog has a New Issue button, which enables the user to create a new issue record "on the fly". If the user hovers over the icon to the right of the New Issue button, it will display a tooltip containing the Change Package Triggers query. This makes it easier to create an issue that will be selected by the query, and thus available to promote against.

   There may also be a user field to the right of the Issue # field that can be used to show issues belonging to another user.

   Typing an Identifier when using a Third-Party ITS -- If a field other than the default issueNum is specified as the 3pty ITS Key (on the Schema Editor's Schema subtab), specifying issue records by typing their identifiers can lead to errors: if multiple issue records have the same identifier, AccuRev automatically adds the promoted version(s) to just one of those records — perhaps not the one you intended.

4. The command completes its work.

5. The versions involved in the command are recorded in the change package section (Changes page) of the selected issue record(s).

What happens if the user specifies no issue record or a non-existent issue record, such as 99999 or 0?

- If the user enters "0" (or equivalently, makes a blank or non-numeric entry), AccuRev checks whether issue record #0 is among the issues selected by the query executed Step 1. (**Note:** the query can select issue record #0, even though it doesn't exist -- for example with this clause: issueNum equal to 0.)
  - If the query *does* select issue record #0, the user's command completes but no information is recorded in the issue. This provides a way for the user to bypass the integration when performing the *Promote* command.
  - If the query *does not* select issue record #0, the user's command is cancelled, and no information is recorded in the issue.
- If the user specifies a non-existent issue record, such as "99999", the command is cancelled, and no information is recorded in the issue.

Configuring and Using Third-Party ITS Keys

If you use AccuRev with a third-party issue tracking system (whether through an integration, or just in parallel), you might find it convenient to use the third-party issue numbers (“keys”) along with or instead of AccuWork issue numbers.

To do this, you need to:

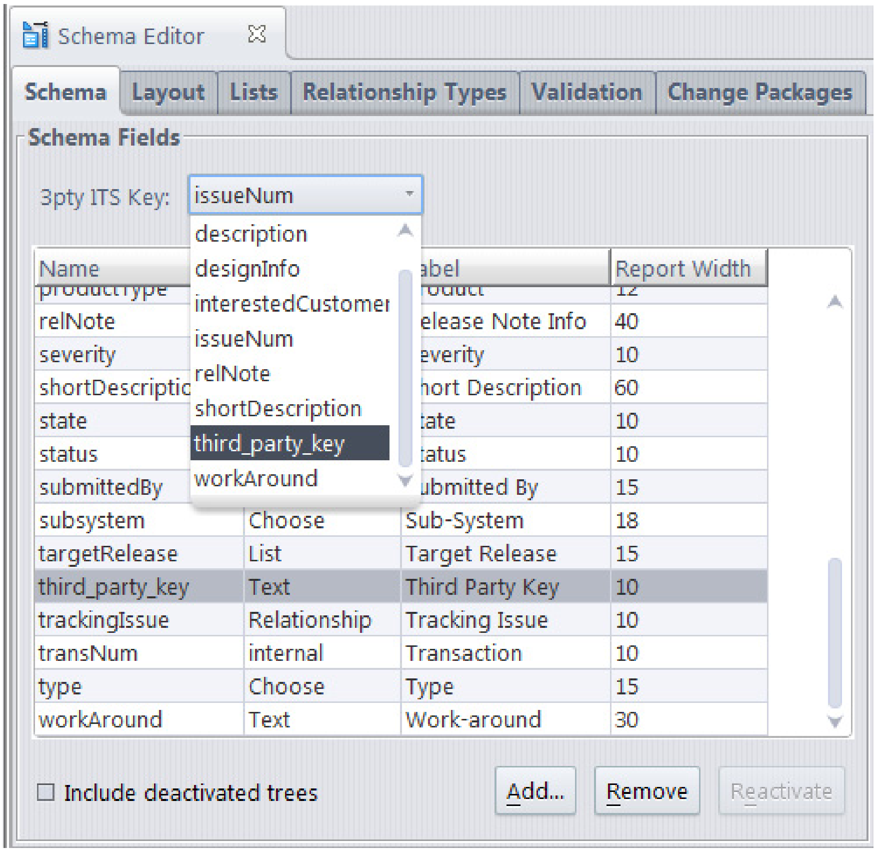
- Customize your AccuRev schema to define a new third-party issue field (or make use of an existing one).
- Specify a `-3` switch with CLI commands that make use of third-party tracking numbers (for example, `promote -3 -I` *`<3rd_party_ITS_key> <element_name>`*).

Modifying the schema

To make use of third-party ITS keys, you must first modify the AccuRev schema. You can do this by either manually editing the schema.xml file, or by using the Schema Editor in the AccuRev GUI (see the “Schema Editor” section of the AccuRev *On-Line Help Guide)*.

Using the Schema Editor

If you do not wish to edit schema.xml directly, you can use the Schema Editor in the AccuRev GUI. Identify an existing key field that can be used, or define a new one using the **Add** button. Then assign this key field using the **3pty ITS Key** dropdown menu.



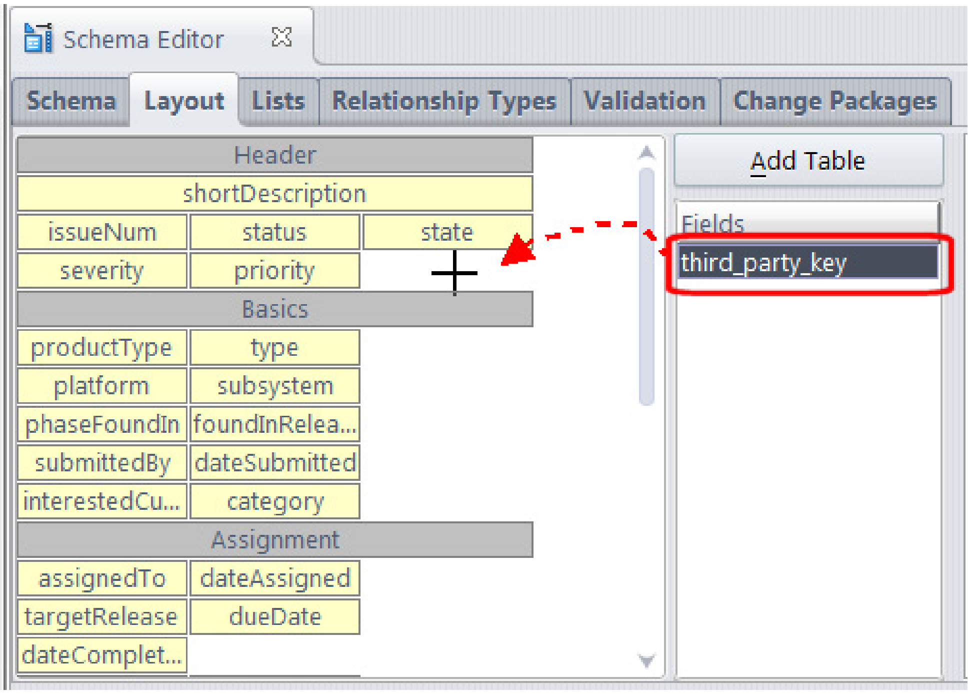
3pty ITS Key: issueNum

Name		abel	Report Width
productType		roduct	12
relNote		elease Note Info	40
severity		everity	10
shortDescriptic		hort Description	60
state		tate	10
status		tatus	10
submittedBy		ubmitted By	15
subsystem	Choose	Sub-System	18
targetRelease	List	Target Release	15
third_party_key	Text	Third Party Key	10
trackingIssue	Relationship	Tracking Issue	10
transNum	internal	Transaction	10
type	Choose	Type	15
workAround	Text	Work-around	30

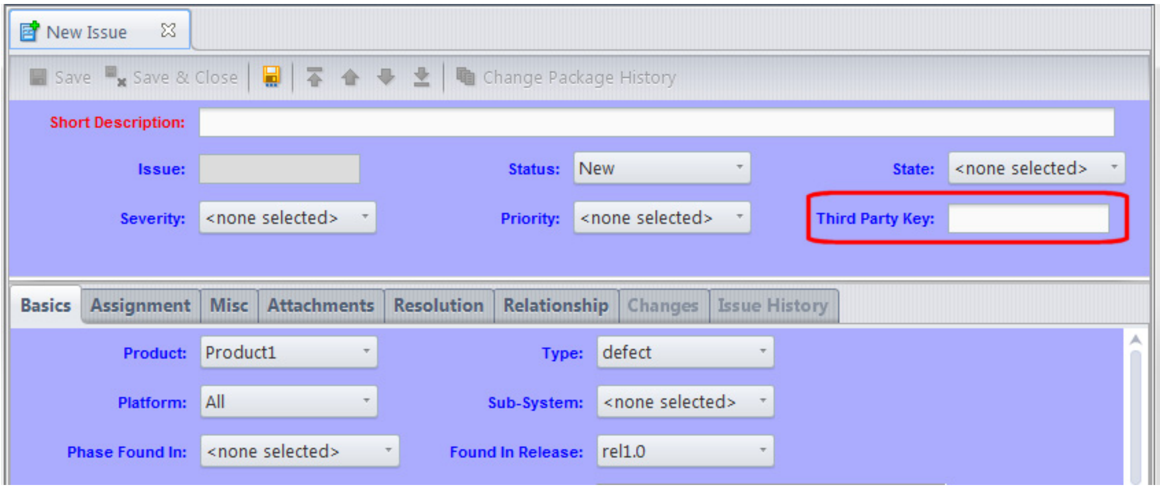
☐ Include deactivated trees

Note that some AccuRev schemas (like the schemas delivered with AccuSync) already have a third-party key defined, such as “jirakey”. If this is the third-party ITS key that you need to use, you just need to ensure that it is assigned to the **3pty ITS Key** dropdown menu as described above, and that it is positioned on the Issue form as described below.

Next go to the ***Layout*** tab and position this field in the Issue form (or confirm that it is already there). To position it, select the entry in the Fields column, and then click where you want it to appear in the left side of the tab (for example, in the Header section).



Save your changes and exit the Schema Editor. Now when you display the AccuWork Issue form, the third-party ITS key field will be
displayed.



Short Description:

Issue:

Status: New

State: <none selected>

Severity: <none selected>

Priority: <none selected>

Third Party Key:

Product: Product1

Type: defect

Platform: All

Sub-System: <none selected>

Phase Found In: <none selected>

Found In Release: rel1.0

Editing schema.xml

You can also directly edit the schema XML files (schema.xml and layout.xml). In a text editor, open the schema.xml file for your depot:

```
<ac_install>/storage/depots/<depot_name>/dispatch/config/schema.xml
```

First, determine if you need to define a new field to handle the third-party ITS key. Note that all AccuRev schemas have an entry for AccuWork issue numbers. For example:

```
<field
name="issueNum"
type="internal"
label="Issue"
reportWidth="10"
fid="1">
```

Do not use this for third-party ITS keys. Define a new one (or determine if you already have one available in your schema) and make sure that it is specified in the `lookupField` as described below.

Some AccuRev schemas (such as the schemas delivered with AccuSync) already have a third-party key defined, such as:

```
<field
name="jiraKey"
type="Text"
label="JIRA Key"
reportWidth="10"
width="10"
fid="3">
</field>
```

If you need to define a new key, do so using the above key as an example, making sure to assign it an unused `fid` (field ID) number.

After identifying a key field, or defining a new one, modify the `fid` value of the `lookupField` entry to reflect the new key field (the `lookupField` entry is typically found near the top of schema.xml:

```
<lookupField fid="1"/>
```

For example, if you were modifying this entry to use the `jiraKey` field shown above, you would change the `fid` value to “3”.

If you will be using this convention in new depots that you create in the future, you should also make this change to the master copy of schema.xml in:

```
    <ac_install>/storage/site_slice/dispatch/config
```

To position this field on the Issue form (or to confirm that its position is already defined), open layout.xml in a text editor. Use the existing entries in the file as a model for defining the third-party ITS key field.

Using Third-Party ITS Keys in the CLI

Once the schema has been configured as described in the previous section, you use third-party ITS keys with a variety of AccuRev CLI input commands by specifying the new “-3” switch in addition to the traditional “-I” switch (for example “-3 -I US23407”). Note that you can also still use AccuWork issue numbers by using the traditional “-I” switch by itself, but you cannot specify both on the same command line.

Here are the AccuRev commands that accept the -3 switch:

- cpkadd
- cpkdepend
- cpkdescribe
- cpkremove
- issuelist
- merge

- patch
- promote
- purge
- revert

Here are some examples of using the **promote** command with the -3 switch:

Promote a file and associate it with an issue using the third-party issue key:

```
accurev promote -I <3rd_pty_key> -3 <element_name>
```

Promote a file and associate it with several issues using third-party issue keys:

```
accurev promote -I "<3rd_pty_key_1> <3rd_pty_key_2> <3rd_pty_key_3>"
   -3 <element_name>
```

You can also use the third-party keys with the -Fx switch for XML. For example, to promote a file using the -Fx switch and associate it with several issues using third-party issue keys:

```
accurev promote -Fx -l <xml_filename>
```

where the XML file has the following format:

```
<issues third_party=”true”>
<id>[3rd_pty_key_1]</id>
<id>[3rd_pty_key_2]</id>
<id>[3rd_pty_key_3]</id>
</issues>
```

Commands That Return Third Party Issue Numbers

These commands return third-party key information in XML output (specified with the -fx switch):

- **hist**
- **cpkdepend**
- **xml -l**

cpkdepend also returns third-party key information in non-XML format. If you specify **-3** to query the dependency, then the values in non-fx will be displayed using third-party values. If specify standard issue numbers, they will be displayed in standard format. If you specify **-3** without **-fx** and an issue does not have a third-party value, an asterick ('*') appears before the issue number.

In the examples below, the third-party values are displayed in **bold**. "`thirdPartyName`" is the field name (not the label name) defined in the Schema Editor.

hist example output

```
<version

        path="\.\file.cpp"

        eid="3"

        virtual="4/2"

        real="5/2"

        virtualNamedVersion="s25263_proj1_dev/2"

        realNamedVersion="ws25263_testuser1/2"

        elem_type="text"

        dir="no">

      <issueNum
```

```
          thirdPartyName="jiraKey"
          thirdPartyValue="J1002">2</issueNum>
    </version>
```

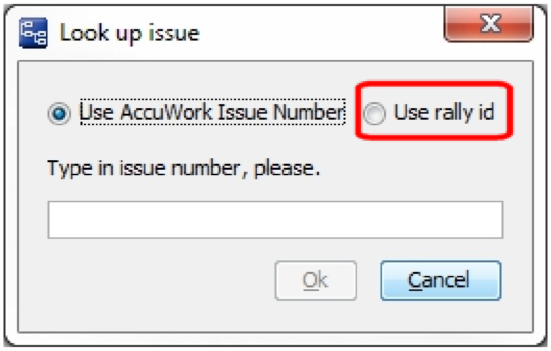
cpkdepend example output

```
<acResponse>
  <issueDependencies>
    <issueDependency>
      <issues>
        <issue
            number="2"
            thirdPartyName="jiraKey"
            thirdPartyValue="J1002"
            incomplete="false"/>
      </issues>
      <dependencies>
        <issue
            number="1"
            thirdPartyName="jiraKey"
            thirdPartyValue="J1001"/>
      </dependencies>
    </issueDependency>
  </issueDependencies>
</acResponse>
```

Using Third-Party Keys in the AccuRev GUI

Once you have enabled the use of third-party issue tracking system (ITS) keys in the AccuWork schema, the ***Select Issue (Change Package)*** dialog and the *Look Up Issue* dialog display additional controls that let you indicate whether you want to use AccuWork issue numbers or third-party ITS keys when specifying issues.

For example, when you bring up the *Look Up Issue* dialog, if third-party ITS keys are enabled AccuRev displays radio buttons that let you indicate that you want AccuRev to use the third-party key rather than the AccuWork issue number when looking up the issue.



● Use AccuWork Issue Number ○ Use rally id

Type in issue number, please.

The radio button for the ITS is labeled with whatever text is specified in the Schema Editor for the "Label" field (see *Using the Schema Editor* on page 404).

B. Support and Troubleshooting

Customer Care Notes

To assist in a speedy resolution of problems with AccuRev usage or performance, Customer Care personnel may ask you to provide data that you may not be accustomed to gathering on a day-to-day basis. The most frequently requested items are listed below, with instructions on how to gather the data. When you submit your initial Support Incident, please include the data items that you know (or suspect) are related to your specific issue.

Collecting metadata

UNIX or Linux

1. Run the ***accurev backup*** command. This command will back up your *metadata* into the AccuRev *site_slice/backup* directory.
2. Use ***accurev show slices*** to find the location of the relevant depot directory (slice).
3. Stop the AccuRev Server.
4. ***tar*** your *site_slice* directory (including the backed-up metadata from step 1). Example:

   ```
   cd /opt/accurev/storage/site_slice
   tar -cvf ~/site_slice.tar .
   ```

   Remember to include the dot '.' at the end of the tar command.
5. ***tar*** the relevant depot directory (slice), but exclude the *data* subtree. **(The data subtree contains the content of the files you have placed into AccuRev source control; please do not send it to us!)** Example:

   ```
   cd /opt/accurev/storage/depots/ACME
   tar -cvf ~/ACME.tar --exclude data .
   ```
6. Compress the *.tar* files using ***gzip***. Example:

   ```
   gzip ~/site_slice.tar ~/ACME.tar
   ```
7. Start the AccuRev Server.
8. Send the gzipped files (*.tar.gz* suffix) to Customer Care. We recommend using FTP rather than email, since the files may be too large for your outgoing email server. Please use your Support Incident to obtain a username and password for an FTP server.

Windows

1. Run the ***accurev backup*** command. This command will back up your *metadata* into the AccuRev *site_slice/backup* directory.
2. Use ***accurev show slices*** to find the location of the relevant depot directory (slice).
3. Stop the AccuRev Server.

4. Create a ZIP archive of the entire *site_slice* directory (including the backed-up metadata from step 1). Typically, it is *C:\Program Files\AccuRev\storage\site_slice*.
5. Create a ZIP archive of the relevant depot directory (slice). Typically, it's a subdirectory of *C:\Program Files\AccuRev\storage\depots*. Remove the *data* subtree from the ZIP archive. **(The data subtree contains the content of the files you have placed into AccuRev source control; please do not send it to us!)**
6. Start the AccuRev Server.
7. Send the ZIP files to Customer Care. We recommend using FTP rather than email, since the files may be too large for your outgoing email server. Please use your Support Incident to obtain a username and password for an FTP server.

Database inconsistencies

Gather the following information to begin troubleshooting the issue:

1. What changed in the environment? Power failure? Hardware failure?
2. What are the hardware specs on the AccuRev Server machine?
3. What is the AccuRev version on the AccuRev Server machine?
4. Get output of ***accurev info*** on the AccuRev Server machine.
5. Copy (ZIP or tar file) all AccuRev Server logs, located in subdirectory *.../storage/site_slice/logs*. Search for any messages that might be the cause of your issue: disk failure, power outage, etc.
6. Copy all operating system logs (location varies from OS to OS).
7. Stop the AccuRev Server.
8. Get output of ***maintain dbcheck***.
9. Start the AccuRev Server.
10. Gather AccuRev metadata.

Running the AccuRev GUI in debug mode

You might occasionally experience problems in the AccuRev GUI -- the GUI might appear to hang, or it might display errors that running the equivalent command in the AccuRev CLI does not, for example. Running the AccuRev GUI in Java debug mode can help you diagnose and troubleshoot problems you encounter, and being able to monitor the commands the GUI calls can help with scripting efforts.

The following sections describe how to run the AccuRev GUI on different platforms for different AccuRev versions. Executing these commands starts the AccuRev GUI; debug output is written to the command prompt. Note that paths shown are examples only. You might have to adjust them for your installation.

Windows

To run the AccuRev GUI in debug mode on Windows:

1. Open a CMD shell.
2. cd to the \bin directory where you installed AccuRev. For example:

   ```
   cd C:\Program Files\AccuRev\bin
   ```
3. To run the AccuRev GUI in debug mode, execute the following command:

   ```
   ..\jre\bin\java -cp
   "fw.jar;xercesImpl.jar;xml-apis.jar;oro.jar;logback-classic-1.0.1.jar;
   logback-core-1.0.1.jar;slf4j-api-1.6.4.jar;sam.jar;lax.jar;
   shef-accurev3.jar;jtidy-8.0.jar;bizlaf-cezanne-2.3.11.29.jar;
   AccuRevCommon.jar;jgrapht-0.8.3.jar;org-netbeans-swing-outline.jar"
   fw.MainApp
   ```

   Note: Any jar files with version numbers may change between releases, so change the command line to match the jar file names that you have installed.

UNIX/Linux

To run the AccuRev GUI in debug mode on UNIX/Linux:

1. Open a shell window.
2. cd to the /bin directory where you installed AccuRev. For example:

   ```
   cd /opt/accurev/bin
   ```
3. To run the AccuRev GUI in debug mode, execute the following command:

   ```
   ../jre/bin/java -cp
   "fw.jar;xercesImpl.jar;xml-apis.jar;oro.jar;logback-classic-1.0.1.jar;
   logback-core-1.0.1.jar;slf4j-api-1.6.4.jar;sam.jar;lax.jar;
   shef-accurev3.jar;jtidy-8.0.jar;bizlaf-cezanne-2.3.11.29.jar;
   AccuRevCommon.jar;jgrapht-0.8.3.jar;org-netbeans-swing-outline.jar"
   fw.MainApp
   ```

   Note: Note that any jar files with version numbers may change between releases, so change the command line to match the jar file names that you have installed.

User authentication problems

You can diagnose authentication problems with the ***accurev secinfo*** command:

```
accurev secinfo
```

The output is one of three keywords: anyuser, authuser, or notauth.

anyuser

The user is allowed to perform operations on streams except where permissions require an authenticated user.

authuser

The user is considered authenticated.

notauth

The user is not authenticated.

For more information on the ***accurev secinfo*** command, see the *AccuRev CLI User's Guide*.

Getting AccuRev Server and client version information

How can you get the version information about the AccuRev Server without having to log into the server machine? You can perform either of the following operations on any AccuRev client machine:

- To get the integer version number of the AccuRev server and client software:

```
> accurev info -v
...
client_ver:    58
...
server_ver:    58
 ...
```

- To get the major/minor/patch version numbers of the AccuRev Server, use the ***accurev xml*** command:

1. Create a file (say, *ver.xml*), containing this very short XML document:

```
<serverInfo/>
```

2. Submit this file as an argument to the ***xml*** command:

```
accurev xml -l ver.xml
```

The command output in this format:

```
<serverInfo>
  <serverVersion
    major="4"
    minor="7"
    patch="2"/>
  <handleCount>10</handleCount>
</serverInfo>
```

This example indicates that the AccuRev Server is Version 4.7.2.

Integration issues

If you are reporting a problem with an AccuRev integration with a third-party IDE, include the following in your Support Incident:

1. The name and version number of the IDE software.
2. The steps required to reproduce the problem. Be sure to include the type of project you're working with in the IDE and the location of the AccuRev workspace you're accessing through the IDE.
3. The version of AccuRev you are using.
4. (if applicable) The version of the AccuBridge integration you are using.

AccuRev license issues

Include copies of the following in your Support Incident:

- The *acserver.log* file (in *.../storage/site_slice/logs*).
- The license file, *accurev.lic* or *keys.txt* (in *.../storage/site_slice*). [**NOTE:** *keys.txt* is not supported as of AccuRev 5.4.]
- A screen shot of the error box or a copy of the error text.
- Output from ***accurev info***, executed on the AccuRev Server machine.
- Output from ***accurev show users***.

Potential AccuRev defects

Include the following in your Support Incident:

- A screen shot of the error dialog or a copy of the error text.
- The steps required to reproduce the problem. Include the depot's stream structure, if it is relevant.
- The operating system version on the client machine
- Are multiple users or multiple machines (client vs. server) involved?
- Output from these commands:

```
accurev
accurev info -v
accurev show wspaces
accurev show streams
accurev show slices
accurev show depots
accurev diag
```

Replication problems

Include the following in your Support Incident:

- A copy of the *acserver.cnf* file from both the master server and the replica server.
- A copy of the *acserver.log* file from both the master server and the replica server.
- Output from this command (if the problem is performance-related):

```
accurev diag
```

Performance problems

Include the answers to this questionnaire in your Support Incident:

1. Has performance always been poor? If not, what changed in the environment?
2. What exactly was the user trying to do? Does the performance problem occur with other users, on different client machines? Or is it a common issue on the same machine?
3. Does the operation actually finish? (A hang or lock-up can often be mistaken as performance problem.)
4. What are the AccuRev versions on the problem clients and server machines?
5. What are the hardware specs on the problem clients and server machines?
6. Is performance different in the AccuRev CLI and the AccuRev GUI?
7. Is the user's home directory on a remote disk? If so, consider moving the home directory so that it is on the machine where the AccuRev client software runs.

Include copies of the following in your Support Incident:

- Complete AccuRev server logs (in *.../storage/site_slice/logs*)
- Two samples of output from ***accurev diag***, run on the AccuRev Server machine in the depot storage directory.
- Two samples of output from ***accurev diag***, run on the client machine in the user's workspace directory.
- Output from ***accurev show triggers***.
- The contents of the trigger script files with the names listed below, located anywhere under the *storage* directory on the AccuRev Server machine:

```
server_admin_trig
server_preop_trig
```

  The filenames might include a suffix, such as *.bat* on Windows.
- The AccuRev Server task list.

Server task list

How do you create a listing of the AccuRev Servers currently active tasks (execution threads)?

1. Create a file (say, *tasks.xml*), containing this very short XML document:

```
<tasklist/>
```

2. Submit this file as an argument to the ***xml*** command:

```
accurev xml -l tasks.xml > tasklist.out
```

The contents of *tasklist.out* will resemble this example:

```
<tasklist>
  <task>
    <task_id>1299</task_id>
    <task_name>promote</task_name>
    <user>john</user>
    <host>192.168.89.222</host>
    <status>1</status>
    <starttime>2006/07/25 14:42:49</starttime>
    <seconds_on_server>2</seconds_on_server>
    <depot_write_lock>true</depot_write_lock>
    <progress_counter
      counter="0"
      count="6"
      max_count="11"/>
    <progress_counter
      counter="1"
      count="200"
      max_count="900"/>
  </task>
  <task>
    <task_id>1300</task_id>
    <task_name>xml</task_name>
    <user>john</user>
    <host>192.168.89.222</host>
    <status>1</status>
    <starttime>2006/07/25 14:42:51</starttime>
    <seconds_on_server>0</seconds_on_server>
  </task>
</tasklist>
```

If you are not seeing an element...

If you cannot see an element that you believe that exists, there is a good chance that your AccuRev administrator has set element-level security using the `eacl` command, and you are being denied access to the element (or its container directory). For more information about EACLs and their behavior, see *Element ACLs (EACLs)* on page 264.

If you perform an ***Update***, and the version displayed in the File Browser is correct but the content of the file does not appear to have been updated, this is typically due to a timestamp problem. It usually occurs when a file is changed outside of the workspace and then copied back in. The timestamp of the file that was dropped back into the workspace has a date and time stamp older than the date and time stamp of the file in the workspace. To update the file, temporarily uncheck ***Timestamp Optimization*** and perform the ***Update*** again. See "*Chapter 5 Timestamp Optimization (TSO)*" in the *AccuRev Administrator's Guide* for more information.

Logging

Two types of logging are available in the AccuRev AccuRev GUI:

- The *Logs tab*
- The *accurev.log* log file

This section describes logging features in the AccuRev GUI and how to use them. Logging is also available for the AccuRev Server. See *Server Logging* on page 18 in the *AccuRev Administrator's Guide* for more information.

Note: Logging in the Logs tab and in the **accurev.log** are independent. Settings for the **accurev.log** file have no effect on the information displayed on the Logs tab, for example.

The Logs Tab

The Logs tab provides access to command logs created during the current GUI session by these commands:

- ***Update*** and ***Update Preview***
- ***Populate***
- commands that change ***include/exclude*** rules

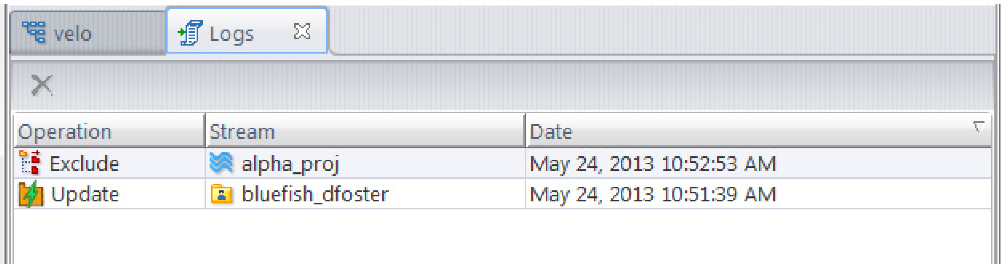
AccuRev creates these logs automatically, and deletes them when you close the AccuRev GUI.

Opening a Logs Tab

Choose **View** > **Logs** from the GUI main menu.

Logs Tab Layout

The Logs tab contains a table with Operation, Stream, and Date columns:



Operation	Stream	Date
Exclude	alpha_proj	May 24, 2013 10:52:53 AM
Update	bluefish_dfoster	May 24, 2013 10:51:39 AM

- **Operation -** The command that created the log: Update, Update Preview, Populate, Include, Exclude, Clear.
- **Stream -** The workspace or stream in which the operation took place.
- **Date -** A timestamp indicating when the operation took place.

Working in a Logs Tab

You can display, save and delete individual logs, using these commands:

- **View -** Display the contents of the selected log in a text editor window. You can use environment variable AC_EDITOR_GUI or EDITOR to control which text editor gets invoked. If neither variable is set, AccuRev selects a program provided by the operating system.
- **Save As -** Copy the selected log to another location.
- **Delete -** Remove the log file from disk storage (and from this table). Logs are stored in temporary files (for example, *C:\Documents and Settings\jjp\Local Settings\Temp\log3382.tmp*). These files are removed automatically at the end of the GUI session.

The accurev.log File

AccuRev records all operations performed in the AccuRev GUI in the **accurev.log** file on a daily basis; if you stop and start AccuRev multiple times during the same calendar day, the existing log is updated. The **accurev.log** file for a given day is archived upon creation of the first log message on the *following* day. To create the archive log file, AccuRev renames the daily log file with a *yyyy-mm-dd* timestamp; the archive log file for June 1, 2012 would be named **accurev.2012-06-01.log**, for example. The daily log file, **accurev.log**, is always used to record activity for the current day.

By default, AccuRev stores archive log files for 10 days; archive log files older than 10 days are deleted. Both the daily log file and archive log file are written to the **\log** directory where you installed the AccuRev client (**c:\Program Files (x86)\AccuRev\log**, for example).

Example. Imagine you started AccuRev on June 1, 2012 and used it every day for a couple of weeks. At 9 a.m. on June 2, you performed a **keep** of several modified files; AccuRev creates a message in the **accurev.log** file, and creates the archive log file for the activity for June 1 (**\log\accurev.2012-06-01.log**). Archive log files would be created each day in this fashion (**accurev.2012-06-02.log** would be created sometime on June 3, **accurev.2012-06-03.log** sometime on June 4, and so on). Upon the first logged event on *June 12*, however, two things would occur:

- An archive log file for June 11 activity would be created (**\log\accurev.2012-06-11.log**) as usual.
- The archive log file for June 1, **2012-06-01.log**, would be deleted because it is older than 10 days.

You can configure the number of days you want to keep archive log files and other aspects of the log. See *Configuring Logging* on page 420 for more information.

Sample Log

Following is a sample of the output that is written to the daily log file:

```
15:04:07.783 [AWT-EventQueue-0] DEBUG [RunProcess.java:341] fw.process.RunProcess -
ACAPI: [xml]  'C:\Program Files (x86)\AccuRev\bin\accurev.exe' 'show' '-p' 'depot1'
'-fxg' '-r' '-l'
'C:\Users\jhollows\AppData\Local\Temp\roots3557908586359241917.tmp' 'streams'


15:04:08.192 [AWT-EventQueue-0] DEBUG [RunProcess.java:341] fw.process.RunProcess -
ACAPI: [xml]  'C:\Program Files (x86)\AccuRev\bin\accurev.exe' 'show' '-fix'
'users'

15:04:08.515 [AWT-EventQueue-0] DEBUG [RunProcess.java:341] fw.process.RunProcess -
ACAPI: [xml]  'C:\Program Files (x86)\AccuRev\bin\accurev.exe' 'show' '-fx'
'wspaces'


15:04:08.890 [AWT-EventQueue-0] DEBUG [RunProcess.java:341] fw.process.RunProcess -
ACAPI: [xml]  'C:\Program Files (x86)\AccuRev\bin\accurev.exe' 'show' '-fx' 'locks'


15:04:09.615 [AWT-EventQueue-0] DEBUG [RunProcess.java:341] fw.process.RunProcess -
ACAPI:  'C:\Program Files (x86)\AccuRev\bin\accurev.exe' 'xml' '-l'
'C:\Users\jhollows\AppData\Local\Temp\kandoFinding8147860611326777358.tmp'
```

Logging Levels

AccuRev supports several logging levels, as summarized in the following table:

Level	Description
TRACE	Includes TRACE, DEBUG, INFO, WARN, and ERROR.
DEBUG	Includes DEBUG, INFO, WARN, and ERROR.
INFO	Includes INFO, WARN, and ERROR. INFO is the logging level default.
WARN	Includes WARN and ERROR.
ERROR	Includes only ERROR.
OFF	Turns off logging.

To change the logging level, modify the `level=` attribute of the `<root>` element in the **logback.xml** file. See *Configuring Logging* on page 420 for more information.

Configuring Logging

The **logback.xml** file contains several editable settings that allow you to configure logging in the **accurev.log** file as summarized in the following table:

If You Want to ...	Edit This XML Element Tag	Comments
Change the number of days AccuRev stores archive log files	`<maxHistory>`	By default, AccuRev stores archive log files for 10 days.
Change the logging level	`<root level="INFO">`	The default value for the `level=` attribute is "INFO". See *Logging Levels* on page 419 for more information.
Turn off logging altogether	`<root level="INFO">`	Set the value of the `level=` attribute to "OFF".

NOTE: You should not change other values in the **logback.xml** file.

To change a setting in the **logback.xml** file:

1. Open the **logback.xml** file. This file is in the **\bin** directory where you installed the AccuRev client.
2. Find the XML element tag whose value or attribute you need to change.
3. Save the **logback.xml** file.

   Changes to logging settings take effect after a minute or two; you do not have to restart AccuRev.

Additional Information

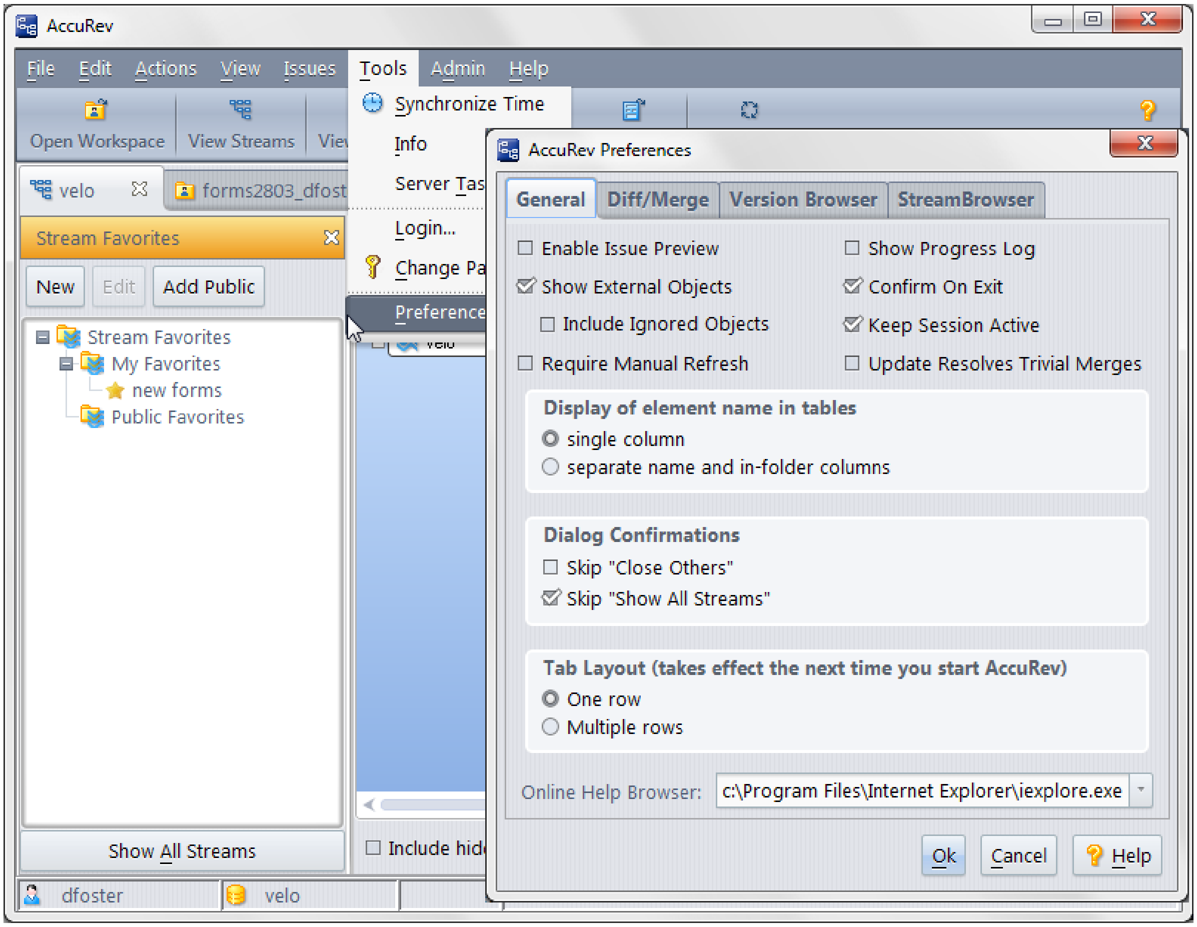
AccuRev uses logback third-party software to implement logging. You can find more information on logback, including logback user documentation, here: *http://logback.qos.ch*.

Configuring a Web Browser to View Help Topics

The AccuRev help system is JavaScript-based to provide advanced search and navigation capabilities. Invoking help automatically launches a web browser , or opens a new tab in an existing web browser window, depending on which browser you use. Any JavaScript-enabled web browser is capable of displaying AccuRev GUI help topics, although you may need to adjust some security settings in more recent browser releases (see below).

By default, AccuRev tries to launch the operating system's default web browser. However, you can tell AccuRev which browser to launch by setting a preference that specifies the complete pathname of the web browser you which to use. Each user can do this either in the GUI or through the Command Line Interface (CLI).

In the GUI, click **Tools** > **Preferences** and enter the full pathname of the browser in the **Online Help Browser** field on the **General** tab:



☐ Enable Issue Preview
☑ Show External Objects
☐ Include Ignored Objects
☐ Require Manual Refresh
☐ Show Progress Log
☑ Confirm On Exit
☑ Keep Session Active
☐ Update Resolves Trivial Merges

Display of element name in tables
● single column
○ separate name and in-folder columns

Dialog Confirmations
☐ Skip "Close Others"
☑ Skip "Show All Streams"

Tab Layout (takes effect the next time you start AccuRev)
● One row
○ Multiple rows

Online Help Browser: c:\Program Files\Internet Explorer\iexplore.exe

Or, use the ***setpref*** command from the CLI. For example:

```
accurev setpref AC_BROWSER "C:\Program Files\Mozilla Firefox\firefox.exe"
```

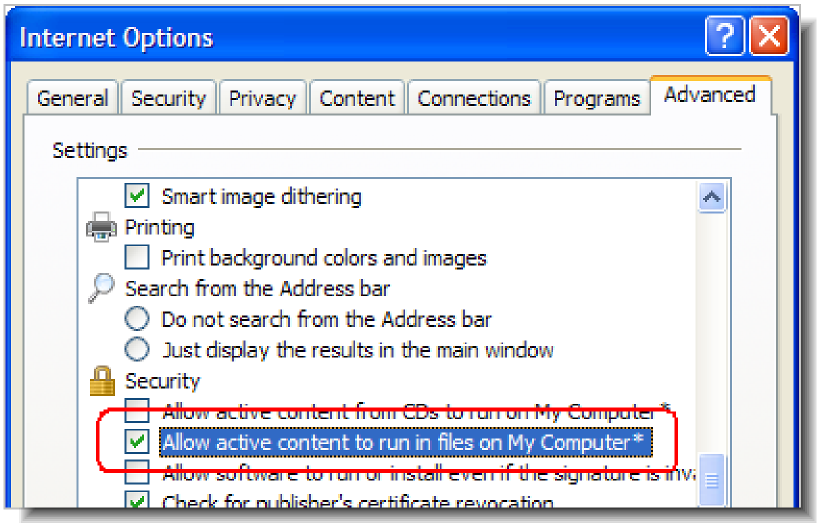
Either method updates the AC_BROWSER setting in *preferences.xml* in the *.accurev* subdirectory of the user’s home directory.

> **Note:** If you are upgrading from AccuRev 4.9 and are having difficulty getting the Help to work, particularly on a 64-bit Windows system, check to see if you have an old browser preference set. If so, clear out this setting and let AccuRev use its default value to get going.

Notes on Internet Explorer

Allowing "Active Content" with Internet Explorer 7

The first time you use AccuRev Help with MS Internet Explorer, you may get a pop-up blocker. We have placed instructions on the default screen that appears with the pop-up blocker to guide you through the Advanced Settings of MSIE that will allow it to work with AccuRev help. These instructions are:



☑ Smart image dithering

Printing

□ Print background colors and images

Search from the Address bar

○ Do not search from the Address bar

○ Just display the results in the main window

Security

□ Allow active content from CDs to run on My Computer*

☑ Allow active content to run in files on My Computer*

□ Allow software to run or install even if the signature is invi

☑ Check for publisher's certificate revocation

1. In MSIE, click ***Tools* > *Internet Options***.
2. In the Internet Options window, click the Advanced tab.
3. Scroll down to the Security settings, and ensure that **Allow active content to run in files on My Computer** is checked.
4. Exit and restart MSIE.

Notes on Google Chrome

Recent Chrome browsers have introduced a new security model which by default prevents them from working with Javascript-based web content residing on a local file system rather than on a server (see `http://code.google.com/p/chromium/issues/detail?id=47416`). This has caused several companies' help systems to stop working with these browsers.

Possible workarounds include:

- rolling back to an older version of Chrome
- using a different browser
- disabling the security feature in the AccuRev preferences
- disabling the security feature system-wide

Google has provided a command line switch to disable this new security feature. You can try disabling this feature just for AccuRev Help, by using the following procedure:

1. In the AccuRev GUI, go to ***Tools*** > ***Preferences***.
2. In the On-line Help Browser field, specify the start-up command line for Chrome with the "--allow-file-access-from-files" argument (Windows example shown):

```
"C:\Documents and Settings\<user>\Local Settings\Application
Data\Google\Chrome\Application\chrome.exe" --allow-file-access-from-files "%1"
```

If this does not work for your installation, another option is to try using this switch as the default for all instance of Chrome on your system. The following example shows how to use this fix in Windows. Adjust appropriately for other operating systems. The following procedure assumes that you have Chrome set as your default .htm/.html browser:

1. Open Windows Explorer.
2. Navigate to ***Tools*** => ***Folder Options*** => ***File Types***
3. Select "HTM" or "HTML", then click Advanced.
4. Select ***Open***, then click ***Edit...***
5. In the "Application used to perform action:" field, you should see a line similar to:

   ```
   "C:\Documents and Settings\<user>\Local Settings\Application
   Data\Google\Chrome\Application\chrome.exe" -- "%1"
   ```

6. Replace the empty "--" with "--allow-file-access-from-files":

   ```
   "C:\Documents and Settings\<user>\Local Settings\Application
   Data\Google\Chrome\Application\chrome.exe" --allow-file-access-from-files "%1"
   ```

7. Click ***OK***, ***OK***, and ***Close*** to save your changes and exit out of the dialogs.

Of course, if this line has already been customized at your site, you will need to adjust the above instructions as needed.

Also, note that implementing this fix will disable a security change introduced with Chrome, so the security policies of some organizations may prohibit them from doing this.

Index