AccuRev®

Getting Started with AccuRev

Version 7.3

Revised 1-February-2019

Copyright and Trademarks

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

This product incorporates technology that may be covered by one or more of the following patents:
U.S. Patent Numbers: 7,437,722; 7,614,038; 8,341,590; 8,473,893; 8,548,967.

Table of Contents

4. Issue Management with AccuWork™ 23

Preface

This book gets you started, quickly and easily, with AccuRev®, the software configuration management solution designed to unleash the power of software development teams to deliver superior results.

The first chapter introduces you to some general concepts about the AccuRev usage model and the four basic commands that allow you to accomplish much of your day-to-day work. Subsequent chapters describe typical Developer and Release Engineer experiences using AccuRev configuration management.

Audience

This book is intended for new AccuRev users, in particular, developers who will be using AccuRev on a daily basis.

Before You Begin

This book assumes that you or your AccuRev administrator have performed a complete AccuRev installation, including the initial setup of an AccuRev depot and workspaces, and the installation of a local AccuRev client. You will use your local AccuRev client to perform the procedures described in this book.

Using This Book

This book assumes you are familiar with your operating system and its commands. It does not assume any experience with AccuRev.

The chapters in this book provide an introduction to basic AccuRev concepts and procedures:

Chapter	Description
Chapter 1 Day-to-Day Usage of AccuRev®	Describes basic AccuRev concepts and how AccuRev fits into the developer's daily routine.
Chapter 2 The Developer Experience	Describes procedures for setting up and working with AccuRev workspaces.
Chapter 3 The Release Engineer Experience	Describes how to use the StreamBrowser to create and compare streams, create snapshots, and manage stream hierarchy.
Chapter 4 Issue Management with AccuWork™	Describes how to use AccuWork, AccuRev's integrated issue tracking system.

Typographical Conventions

This book uses the following typographical conventions:

Convention	Description
`blue sanserif`	Used for sample code or output.

Convention	Description
`red monospace`	Used for examples.
bold	Used for command names, and button names in the AccuSync Web user interface.
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

Micro Focus 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 Micro Focus 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. Day-to-Day Usage of AccuRev®

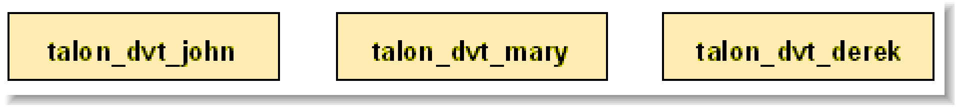
This chapter introduces the basic AccuRev concepts and procedures that will enable you to be comfortable and productive using AccuRev on day one. More detailed information is always just a click away: pressing F1 or clicking the help button (?) anywhere in the AccuRev GUI provides you with access to the entire AccuRev documentation set in both HTML and PDF formats.

The AccuRev® Usage Model

AccuRev's flexibility makes it easy to use for a variety of development scenarios. But like every software system, AccuRev has usage models that were foremost in the minds of its architects. This section describes the most common usage model.

AccuRev is a *software configuration management* (SCM) system, designed for use by a team of people who are developing a set of files. This set of files might contain source code, images, technical and marketing documents, audio/video tracks, or any other digital content the user puts into the system. The files and directories in AccuRev are said to be "version-controlled" or "under source control".

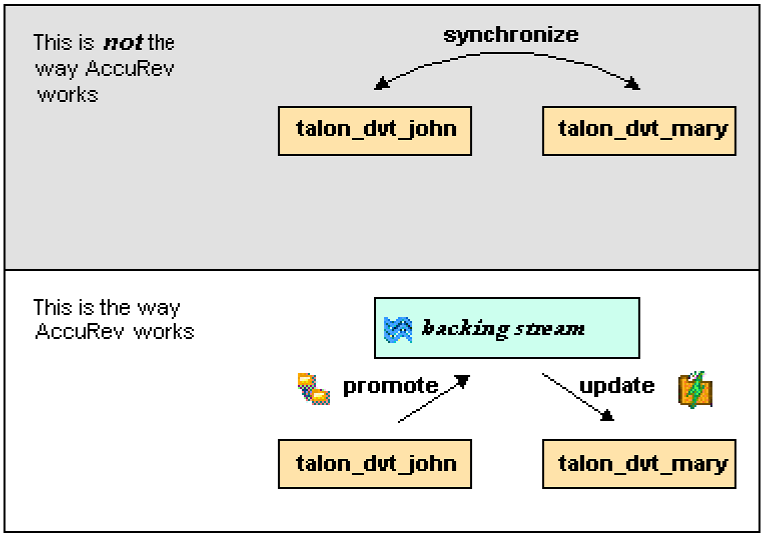
For maximum productivity, the team's members must be able to work independently of each other -- sometimes for just a few hours or days, other times for many weeks. Accordingly, each user has his own private copy of the version-controlled files. The private copies are stored on the user's own machine (or perhaps in the user's private area on a public machine), in a directory tree called a *workspace*. We can picture the independent workspaces for a three-user team comprised of John, Mary, and Derek as follows:



This set of users' workspaces uses the convention of having like names, suffixed with the individual usernames. AccuRev enforces this username-suffix convention. In this example, **talon_dvt** might mean "development work on the Talon product"; **john**, **mary**, and **derek** would be the users' login names.

From AccuRev's perspective, development work in this set of workspaces is a continual back-and-forth between getting "in sync" and "out of sync":

- Initially, the workspaces are completely synchronized: each workspace has copies of the same set of version-controlled files.
- The workspaces become unsynchronized as each user makes changes to some of the files.
- Periodically, users share their changes with each other. When **john** incorporates some or all of **mary**'s changes into his workspace, their two workspaces become more closely (perhaps completely) synchronized.



You might assume that the workspace synchronization process involves the direct transfer of data from one workspace to another. But this is not the way AccuRev organizes the work environment. Instead of transferring data directly between private areas (that is, between users' workspaces), AccuRev organizes the data transfer into two steps:

1. One user makes his changes public -- available to all the other members of his team. This step is called *promoting*.
2. Whenever they wish, other team members incorporate the public changes into their own workspaces. This step is called *updating*.

The first step involves a public data area, called a stream. A *stream* is an AccuRev structure with two very important features that support and simplify parallel development. First, as implied by the preceding illustration, numerous individual contributors can create workspaces on a single stream, allowing them to easily share their work with others, and to get changes from them when they wish. As you will see later, AccuRev has numerous safeguards that prevent one user from overwriting another user's changes, and tools that help you resolve conflicting changes when this occurs. Second, in a stream hierarchy, *inheritance* lets child streams automatically inherit changes from the parent streams above them, reducing the need for tedious merges that might otherwise be required when several developers are working on the same code.

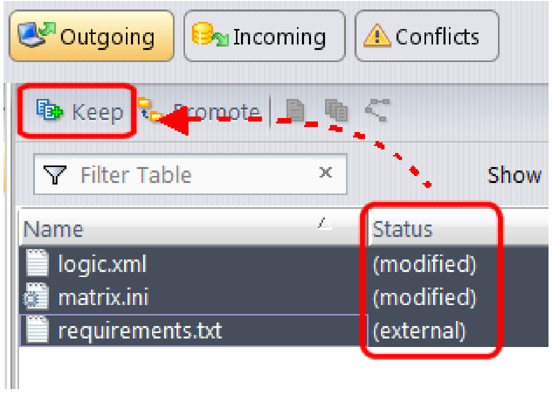
AccuRev has several kinds of streams, but the most common one is the backing stream. Later, we will show you how the data in this public stream "is in back of" or "provides a backstop for" all the private workspaces of the team members.

Change and Synchronization: The Four Basic Commands

With the usage model described above, you will be able to accomplish most of your AccuRev work with four simple commands: **Keep**, **Promote**, **Update**, and **Merge**. These commands are described in the following sections.

Keep: Preserving Changes in Your Private Workspace

An AccuRev workspace is just a normal directory tree, in which you make changes to version-controlled files. You can work with the files using text editors, build and test tools, IDEs, and so on, just as if the files were not version-controlled at all. For example, you might edit a source file and invoke the editor's **Save** command a dozen times over the course of an hour or two. These operations do not involve AccuRev at all -- the operating system changes the contents and the timestamp of the file in your workspace.



Name	Status
logic.xml	(modified)
matrix.ini	(modified)
requirements.txt	(external)

You do not need to perform a "check out" operation or otherwise get permission from AccuRev before editing a file in your workspace. (Some legacy SCM systems do impose such a regimen, and AccuRev can be configured to require checkouts, if your organization requires them.)

Every so often, you want AccuRev to preserve the current contents of the file as an official new *version* of the file. You accomplish this using AccuRev's **Keep** command. This figure shows how to invoke the **Keep** command from the File Browser toolbar; **Keep** is also available from the **Actions** and context (right-click) menus. Note that you can also keep new files, like **requirements.txt** in the preceding illustration -- its status is **(external)** because it has not yet been added to AccuRev.

Tip: The File Browser helps organize your workspace using Outgoing Changes (your work), Incoming Changes (others' work), and Conflicts modes (conflicting changes in your work and that of a teammate).

You can continue modifying the file, using **Keep** to preserve the latest changes, as often as you like. Other team members will not complain about "thrashing" because these new versions stay within your workspace and do not affect any other user's workspace.

AccuRev retains all the versions that you **Keep**. This makes it possible for you to roll back to any previous version you created.

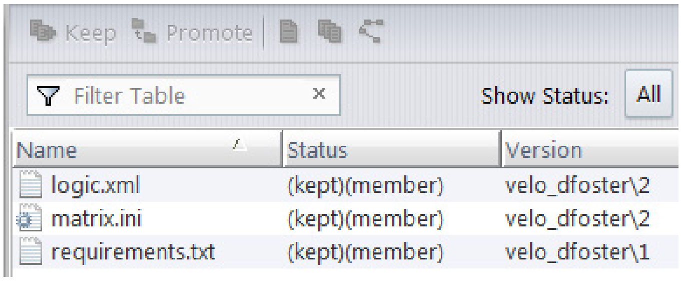
Several other operations are similar to **Keep**, in that they create a new version of a file in your workspace, without affecting any other user's workspace. The most important of these are:

- **Rename** and **Move**: You can rename a file or move it to a different directory (or both), using AccuRev commands. Other users will continue to see the file at its original pathname in their workspaces.
- **Defunct**: You can remove a file from your workspace with the AccuRev command **Defunct**. When you promote a defuncted file, other users will continue to see the file until they update their workspaces, at which point the file will be removed. (**Defunct** differs from a simple **Delete** in that **Defunct** removes the file from your workspace, while **Delete** removes the physical file from your local directory.)

More About Keep

We said earlier that AccuRev "retains all the versions that you **Keep**". But where? Each time you **Keep** a file, its current contents are copied to the AccuRev repository, located on the machine where the AccuRev Server runs. You do not need to care about the name and precise location of this copy. Each version you create has a *version-ID*, such as **velo_dfoster\2** (which translates as "the **2**nd version of this file created in workspace **velo_dfoster**").

AccuRev keeps track of the *status* of each file in a workspace. After you **Keep** a file, the Status column in the AccuRev File Browser contains the indicator **(kept)**. It also contains the indicator **(member)**, meaning that the file belongs to the set of files you are actively working on. (See *Active and Inactive Files* for more information.) The Version column displays the version-ID.



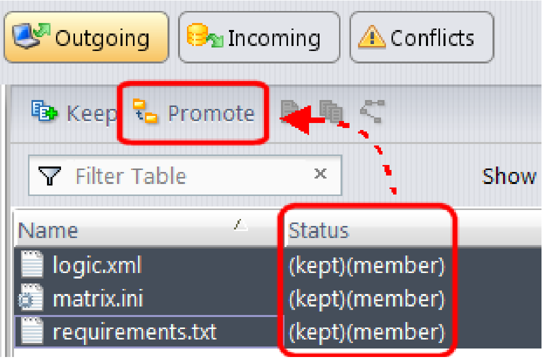
Name	Status	Version
logic.xml	(kept)(member)	velo_dfoster\2
matrix.ini	(kept)(member)	velo_dfoster\2
requirements.txt	(kept)(member)	velo_dfoster\1

A change to the data within a file, recorded by **Keep**, is termed a *content change*; the change made by **Rename**, **Move** or **Defunct** is termed a *namespace change*. (Many SCM systems do not handle namespace changes at all, or have very limited capabilities in this area.) As noted previously, AccuRev saves a new copy of the file in the repository whenever you make a content change. But it does not need to copy the file when you make a namespace change; rather, the AccuRev Server just records the change in its database.

To perform version control on directories, AccuRev only needs to keep track of namespace changes -- renaming, moving, or deleting a directory. Unlike some legacy SCM systems, AccuRev does not need to record a new directory version when you make a content change -- for example, adding a new file to the directory.

Promote: Making Your Private Changes Public

At some point, after you have used **Keep** to create one or more new, private versions of a file in your workspace, you typically want to share the changes you have made with the other team members. To make your (most recent) new version "available to the public", you *promote* it. This figure shows how to invoke the **Promote** command from the File Browser toolbar; **Promote** is also available from the **Actions** and context (right-click) menus.

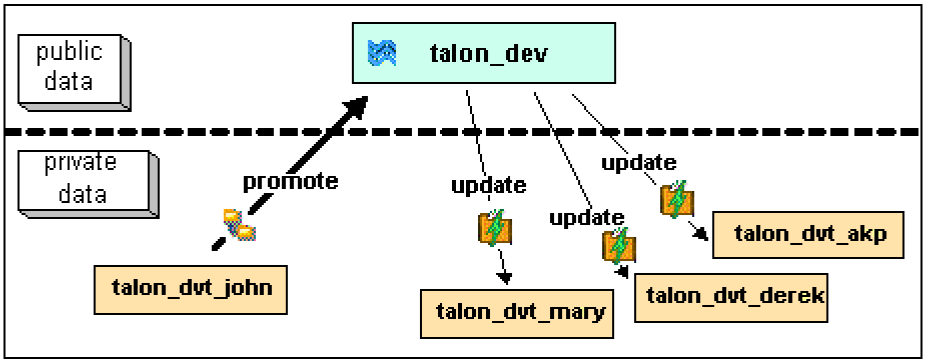


Name	Status
logic.xml	(kept)(member)
matrix.ini	(kept)(member)
requirements.txt	(kept)(member)

Tip: You can **Promote** files that you have not yet kept -- AccuRev performs the necessary **Keep** action for you. In the case of new files, AccuRev first adds the file to the AccuRev repository.

Promoting your new version of a file does not automatically "push" it into the workspaces of the other team members. When a user decides that she is ready to incorporate versions of files that other team members have promoted, she "pulls" them into her workspace with the **Update** command. This process is described in the following section, *Update: Incorporating Others' Changes into Your Workspace* on page 6.

Streams



The **Promote** command sends data to -- and the **Update** command gets data from -- an AccuRev data structure called a *stream*. The stream (named **talon_dev** in this illustration) acts as a central data exchange for the set of workspaces used by a development team. A stream also has a bit of "traffic controller" built in, preventing team members' efforts from colliding and providing other mechanisms to control the flow of data.

A stream is not, as you might initially suppose, a set of copies of promoted files. Rather, it is more like a list of version-IDs.

- The 4th version created in workspace **talon_dvt_akp** of file **logic.xml**
- The 7th version created in workspace **talon_dvt_mary** of file **matrix.ini**
- and so on...

In SCM vernacular, a stream is a *configuration* of a collection of version-controlled files. The term "stream" is apt, because it implies the ongoing changes happening in a development project. Each time a user promotes a version of file **logic.xml**, the stream configuration changes for that file -- for example, from "the 5th version created in workspace **talon_dvt_derek**" to "the 7th version created in workspace **talon_dvt_mary**".

Promotion and Parallel Development

Sometimes, AccuRev does not allow you to promote a file to the development team's stream, because another team member has already promoted the same file (after modifying it in their own workspace). AccuRev prevents you from overwriting your colleague's change to the team's shared stream. This situation is called an *overlap*: two users working at the same time on the same goal, to create the stream's next version of a particular file.

Before you can promote your changes to the stream, you must first perform a *merge* on the file that has an overlap. This command is described in *Merge: When Changes Would Collide* on page 7.

Active and Inactive Files

As you work with a file using the commands described above, AccuRev considers the file to alternate between being *active* in your workspace and *inactive*:

- The file is initially *inactive*.
- When you create a new version in your workspace, using **Keep**, **Rename**, **Move**, or **Defunct**, the file becomes *active*.
- When you make your private version public, using the **Promote** command, the file becomes *inactive* again.

Later, you might restart this cycle, making the file active again by creating another new version of it. Alternatively, updating your workspace might overwrite your inactive file with a newer version that another team member promoted.

AccuRev keeps track of the set of active files in your workspace. Officially, this set is called the *default group*. You might find it easier to think of it as the workspace's "active group".

More About Promote

The **Promote** command does not copy the promoted version to the AccuRev repository. It does not need to. Promotion just gives an additional name to a version that *already exists* in the repository -- having been placed there by a previous **Keep** command (or **Rename**, **Move**, or **Defunct**). For example, promoting "the 7th version created in workspace **talon_dvt_mary**" might give that version the additional name "the 3rd version promoted to stream **talon_dvt**".

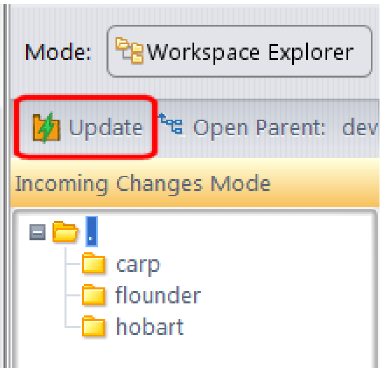
Just to emphasize the previous point: a stream does not reside in the file system, but in the database used by the AccuRev Server. Promoting a version to a stream does not create a copy of a file; it just creates an additional file-reference in the database.

It might seem strange at first that deleting a file with the **Defunct** command makes the file *active*. The File Browser continues to list the file -- with a **(defunct)** status -- even though the file has been removed from your workspace's disk storage. This design feature enables AccuRev to implement the file-deletion operation using the same private-change/public-change scheme as all other changes.

We have discussed *the* stream that is the basis for a set of workspaces. But a typical development project has many streams, organized into a hierarchy. Promoting a version to a higher-level stream from a lower-level stream makes that version "even more public" -- for example, available to users outside your local development team.

Update: Incorporating Others' Changes into Your Workspace

As users work independently of each other, the contents of their workspaces increasingly diverge. Typically, some of the differences between workspaces are inconsistencies. For example, changes that John makes in a report-library routine might cause errors in the report program that Mary is writing. To minimize the time and effort required to resolve inconsistencies during the integration phase of a project, it makes sense to have users synchronize their workspaces on a regular basis.



With AccuRev, synchronization does not mean incorporating data into your workspace directly from one or more other workspaces. Instead, synchronization involves copying data into the workspace from the stream to which all team members **Promote** their changes. This figure shows how to invoke the **Update** command from the File Browser toolbar; **Update** is also available from the **File** and context (right-click) menus.

Tip: You can **Update** your workspace from any File Browser mode, but the Incoming Changes mode shows you exactly how your workspace will change.

> **Note**: the stream's role as a provider of data -- through **Update** operations -- to a set of workspaces motivates the term *backing stream*. Think of restocking a store's shelves with merchandise retrieved from "the back room".

So updating your workspace copies versions of certain files from the backing stream to the workspace, overwriting/replacing the files currently in the workspace. But which files? **Update** changes a file if:

- There is a newer version in the backing stream, and
- The file is *not* currently active in your workspace.

Update will not overwrite an active file, even if there is a new version of it in the backing stream. No matter how good someone else's code is, you do not want his changes to wipe out the changes that you have been making! This situation is another instance of an *overlap*, which was introduced in *Promote: Making Your Private Changes Public* on page 4. You can encounter an overlap during a promote (if you are trying to make your private changes public), or during an update (if you are trying to bring already-public changes into your private workspace). You can use the *merge* operation to resolve all such overlap situations. See *Merge: When Changes Would Collide* on page 7.

Update handles namespace changes as well as content changes. Thus, if your colleague renamed a file and promoted the change, an update will cause the file to be renamed in your workspace. And if your colleague removed a file using the **Defunct** command, an update will cause the file to disappear from your workspace.

More About Update

Here is how AccuRev prevents an update from overwriting your changes: the first thing **Update** does is to analyze your workspace, determining whether each version-controlled file is "active" or "inactive". Initially, all the files in a workspace are inactive -- each one is a copy of some version in the repository. (For each version-controlled file, AccuRev keeps track of *which* particular version.)

A file is deemed to be active in your workspace if you have created a new version of it, using the **Keep**, **Rename**, **Move**, or **Defunct** command. (There are other AccuRev commands that "activate" a file.) When **Update** copies versions from the repository into your workspace, it skips over all active files.

> **Note***:* **Update** can tell if you have modified a file but have not yet stored the changes in the repository as a new **Keep** version. It uses timestamps and checksums to determine this. The presence of such files prevents the update from proceeding if updating would overwrite one or more of them with the backing-stream version. You can use the **Anchor** command to activate such files, enabling **Update** to do its work.

Merge: When Changes Would Collide

The preceding sections on the **Promote** and **Update** commands both discuss the situation in which two users concurrently work on the same file. Their changes to the file are said to *overlap*. Both **Promote** and **Update** decline to process a file with overlap status, because doing so would cause one user's changes to overwrite another's changes.

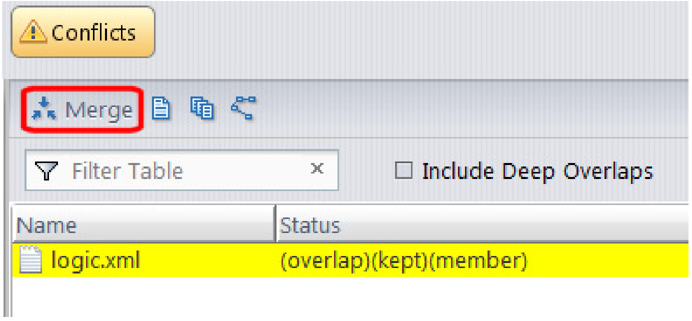
For example:

- Team members John and Mary both Keep one or more new private versions of **logic.xml** in their respective workspaces.
- Mary Promotes her latest new version of **logic.xml** to the backing stream.
- At this point, AccuRev:
  - Will not allow John to **Promote** his version of **logic.xml** to the backing stream (there is no reason to assume that John's changes should take precedence over the changes Mary has already made and promoted).
  - Will not overwrite John's copy of **logic.xml** when he updates his workspace (again, AccuRev does not assume that a version in the backing stream should take precedence over changes made to the

version in the workspace). The **Update** command skips over this file, but continues its work on other files.

- "Flags" the overlap condition by highlighting **logic.xml** in yellow.

Before John can either promote or update **logic.xml**, he must incorporate, or merge, the version in the backing stream -- which contains Mary's changes -- into his own copy of the file. The **Merge** command is essentially a text editor, which combines the contents of two versions of a text file. The resulting "merged version" replaces the file in John's workspace.



Name	Status
logic.xml	(overlap)(kept)(member)

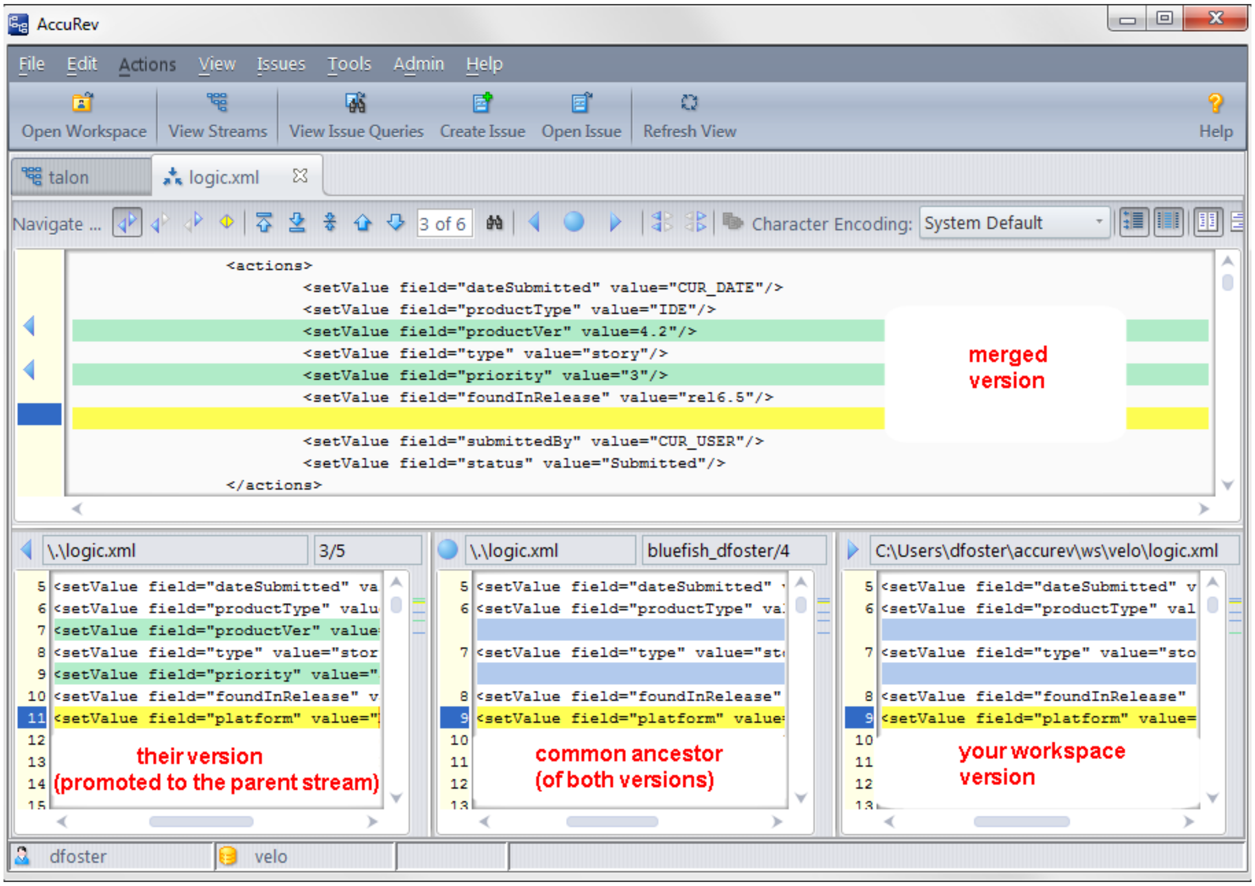
This figure shows how to invoke the **Merge** command from the File Browser toolbar; **Merge** is also available from the **Actions** and context (right-click) menus.

Tip: Use the File Browser Conflicts mode to view overlap and other conditions that prevent files from being updated and promoted.

Often, a merge operation is unambiguous, and so can be performed automatically. For example, suppose Mary's changes to file **logic.xml** all occur in lines 30–50, and all of John's changes occur in lines 125–140. In this case, merging the two versions involves replacing some or all of John's 20 lines with Mary's. Now, the edited version of **logic.xml** in John's workspace contains both users' changes.

> **Note***:* We do not claim that the two sets of changes are semantically consistent with each other. That is what the build-and-test cycle is for!

If both John and Mary have made changes to the same part of the file -- say, lines 2-10 -- then John must decide how to resolve this *conflict*. The graphical **Merge** tool makes this easy:



```
<actions>
        <setValue field="dateSubmitted" value="CUR_DATE"/>
        <setValue field="productType" value="IDE"/>
        <setValue field="productVer" value=4.2"/>
        <setValue field="type" value="story"/>
        <setValue field="priority" value="3"/>
        <setValue field="foundInRelease" value="rel6.5"/>

        <setValue field="submittedBy" value="CUR_USER"/>
        <setValue field="status" value="Submitted"/>
</actions>
```

merged version

```
<setValue field="dateSubmitted" va
<setValue field="productType" valu
<setValue field="productVer" value
<setValue field="type" value="stor
<setValue field="priority" value="
<setValue field="foundInRelease" v
<setValue field="platform" value="
```

their version (promoted to the parent stream)

```
<setValue field="dateSubmitted" 
<setValue field="productType" va

<setValue field="type" value="st

<setValue field="foundInRelease"
<setValue field="platform" value
```

common ancestor (of both versions)

```
<setValue field="dateSubmitted" v
<setValue field="productType" val

<setValue field="type" value="sto

<setValue field="foundInRelease"
<setValue field="platform" value=
```

your workspace version

After performing a merge, AccuRev automatically Keeps the merged version to preserve the results of the merge operation. You can then Promote the merged version to the backing stream. After that, other team members can use Update -- perhaps in conjunction with Merge -- to bring all the changes into their workspaces.

More About Merge

The graphical Merge tool performs a "three-way merge", which uses the common ancestor of the two versions being merged. This algorithm helps to automate the merge operation, often completely eliminating the need for human intervention. AccuRev performs merge operations on text files only. Binary files are "merged" by choosing which version to take.

AccuRev keeps track of all merge operations. This greatly simplifies subsequent merge operations on files that have been merged previously: you do not need to resolve the same conflicts over and over again.

The most common overlap situation happens when AccuRev prevents you from promoting a file, because someone else "got there first" in creating a version in the backing stream. AccuRev can also detect *deep overlaps*, in which another user "got there first" in creating a version in the *parent* of the backing stream, or in other higher-level streams.

Next Steps

Now that you have been exposed to the basic AccuRev concepts of keep, promote, update, and merge, it is time to start AccuRev and try some simple procedures yourself. Go to *Chapter 2 The Developer Experience* to get started.

2. The Developer Experience

The Developer experience includes accessing workspaces, creating new versions of files (edit/keep/promote), updating workspaces, and merging two versions of a file.

Starting AccuRev

Perform this step if AccuRev is not already running.

To start AccuRev, double-click the AccuRev shortcut on your desktop.

Alternative: In a Command Prompt window, enter **acgui**.

Going to Your Workspace

Perform this step if your workspace is not already open in the AccuRev GUI.

1. Click the **Open Workspace** button on the AccuRev toolbar.

   Alternative: Select **File > Open Workspace**.

2. Select the workspace that you want to open from the Open Workspace dialog box and click **Ok**.

   The workspace is displayed in the default File Browser mode, *Explorer* mode.

   All of the files in your workspace have a status of ***(backed)****. This means that you have not yet modified these files and that they are in synch with the version in the backing stream.*

3. Click the Update button (Update) to ensure that your workspace is up-to-date.

Editing Your Source Files

1. If necessary, navigate through folders displayed in the Explorer pane; then select a file to edit.
2. Click the **Edit** button on the File Browser toolbar.

   Alternative: Right-click and choose **Edit** from the context menu. You can, of course, edit files directly from your IDE.

3. Make some changes to your source file.
4. Save the file, and close the editor.

   You did not have to "check out" the file before editing it. The status of the file is now ***(modified)***.

Keeping Your Modified Files

The Keep operation is a "private check in". It makes a copy of your modified file in secure storage maintained by the AccuRev Server. It also records the creation of a new version of the file in the AccuRev database. Keeping new versions in the workspace enables you to checkpoint and back up your own code, while preventing your changes from being seen by other developers.

1. Select the file you modified.
2. Click the Keep button (Keep) on the File Browser toolbar.

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

   Enter a comment in the Keep dialog box, and click **Keep**.

   Note: *The status of the file now includes the indicator* ***(kept)****.*

Sharing Changes with the Rest of the Team

The Promote operation allows you to share your changed file with other users.

1. Select the file you just modified and kept.
2. Click the **Promote** button (Promote) on the File Browser toolbar.

   Alternative: Right-click and choose **Promote** from the context menu.
3. Enter a comment in the Promote dialog box, and click **Promote**.

   Promote makes the version that you created with Keep available to be copied into other users' workspaces the next time they update. The status of the file is once again ***(backed)****.*

Creating a Second Workspace

1. Select **File** > **New** > **Workspace** from the AccuRev main menu.
2. From the **Depot** drop-down list, select your depot.
3. As the **Basis stream**, select the stream with the same name as the depot.
4. Click **Next**.
5. As the **Name of workspace to create**, enter a new name (example: **acme2**).
6. As the **Location** for the new workspace, click **Browse** and either choose an existing directory or create a new directory in the desired location (example: **C:\Documents and Settings\Derek** or **/usr/home/derek**).
7. Make sure the **Append workspace name to its location** checkbox is checked.
8. Click **Finish**.

   The new workspace is created, and Update is invoked to copy the files you placed under version control to the workspace. Click the Close button on the update notification dialog box.

Working on Files in the Second Workspace

1. Open the second workspace and select a text file to edit.
2. Click the **Edit** button on the File Browser toolbar.

   Alternative: Right-click and choose **Edit** from the context menu.
3. Make some changes to your text file.
4. Save the file, and close the editor.

5. Repeat steps 1-4 on one or more additional files.
6. Select the Outgoing Changes mode (Outgoing) in the File Browser.

   Tip: You can use the File Browser Outgoing Changes mode to list all the files that you are currently working on.
7. Select the files you modified.

   Tip: Use the Modified filter to display only those files you changed.
8. Click the **Promote** button (Promote) on the File Browser toolbar.

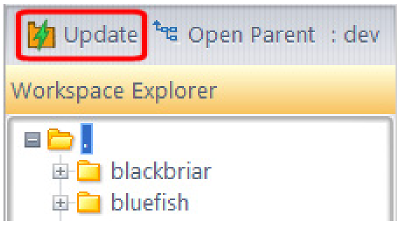
   Alternative: Right-click and choose **Promote** from the context menu.

Updating Your First Workspace

Since you worked on files in your second workspace and promoted those changes to the same backing stream shared with your first workspace, your first workspace is now out of synch. Update the first workspace now to get those changes.

1. Click the tab in the GUI window that contains your first workspace.

   If you have closed that tab, re-open the workspace using:
   File > **Open Workspace**
2. Click the **Update** toolbar button.



This brings the changes you promoted while working in your second workspace into your first workspace.

Merging Two Versions of a File

1. In your first workspace, select a file to edit.
2. Click the **Edit** button on the toolbar.

   Alternative: Right-click and choose **Edit** from the context menu.
3. Make some changes to your text file, save the file, and close the editor.
4. Select the file.
5. Click the Keep button (Keep) on the File Browser toolbar.

   Alternative: Right-click and choose **Keep** from the context menu.
6. Select the file.
7. Click the **Promote** button (Promote) on the File Browser toolbar.

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

8. Switch to your second workspace.

   *Note that the file's status now includes the indicator **(stale)**.*

9. Edit the *same* file in the second workspace and save your changes.

   *The file's status now includes the indicator **(overlap)**. The yellow highlight emphasizes this status. This means that someone else has promoted a version before you did. In this case, the "someone else" is you, working in your first workspace. The file in your second workspace requires a merge before you can promote it.*

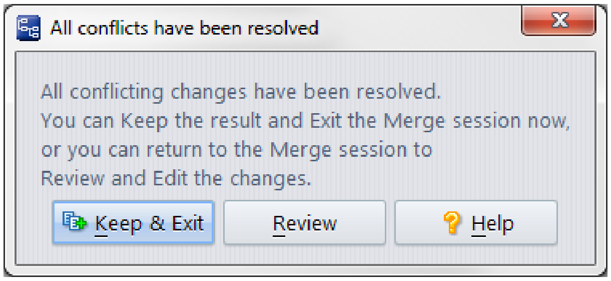
10. Select the Conflicts mode (⚠ Conflicts) in the File Browser.

11. Select the file and click the **Merge** button (Merge) in the File Browser toolbar.

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

    The file is displayed in the AccuRev Merge tool.

12. Proceed to merge the file. Use the up (⇧) and down (⇩) toolbar buttons to navigate from conflict to conflict; use the left (◀) and right (▶) buttons to resolve the current conflict by choosing changes from one of the versions. You finish the Merge by preserving the merged file, selecting **Keep & Exit** when all the conflicts have been resolved.

    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.

    See [The AccuRev Merge Tool](#) on page 220 in the AccuRev On-Line Help Guide for more information.

    To finish resolving the conflict with your merged changes, promote the file.

13. Click the **Promote** button (Promote) in the File Browser toolbar.

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

3. The Release Engineer Experience

The AccuRev experience for a Release Engineer or Quality Engineer includes working with the StreamBrowser™, creating new streams and snapshots, comparing streams, and rearranging the stream hierarchy.

Opening the StreamBrowser™

The StreamBrowser allows you to easily see and control your development stream hierarchy graphically.

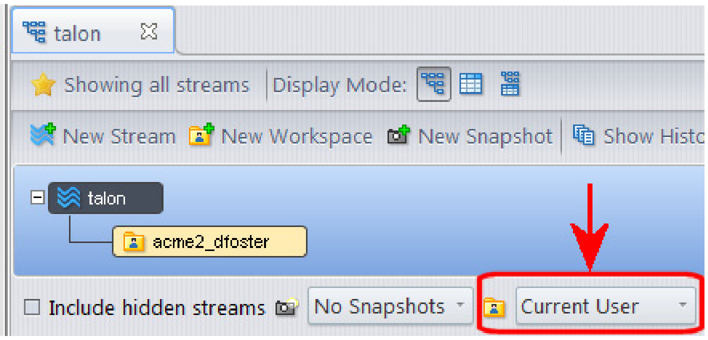
1. Click the View Streams button on the AccuRev toolbar.

   Alternative: Select **View** > **View Streams** from the AccuRev main menu.

2. If prompted, select your depot from the list of depots in the Please Choose Depot dialog box and click **Ok**.

   The StreamBrowser appears, showing the streams in your depot.

3. In the filter at the bottom of the StreamBrowser tab, make sure that **All Workspaces** or **Current User** is selected in the drop-down field.



   These settings include your "private" workspace streams in the display.

   Currently, there is only one "public" development stream: the depot's "base stream" or "root stream". It has the same name as the depot.

Creating an Integration Stream Based on the Base Stream

In environments with multiple congruent projects, it can be useful to create an integration stream and promote your changes there, rather than directly to the base stream.

1. Select the base stream.

2. Click the **New Stream** button (New Stream) on the AccuRev toolbar.

Alternative: Right-click and select **New Stream** from the context menu.

3. Give the new stream a name (example: **acme_int**).
4. Leave the Stream Type set to **Dynamic stream**.
5. Leave the Basis Time set to **None**.
6. Click **Ok**.

   The newly created integration stream appears in the StreamBrowser, as a direct "child" of the base stream.

Creating a Snapshot Stream to Capture the Current State of Development

The team just wrapped up a release and wants to save the version of the files that comprised that release. In AccuRev, you accomplish this by creating a snapshot stream.

1. Select the depot's base stream.
2. Click the **New Snapshot** button (New Snapshot) on the AccuRev toolbar.

   Alternative: Right-click and select **New Snapshot** from the context menu.
3. Give the new snapshot a name (example: **acme_1.0**).
4. Leave the Basis Time set to **Now**.
5. Click **Ok**.

   The newly created snapshot stream appears in the StreamBrowser.

Reparenting Your Workspaces

Your work on the first project has concluded, and you are ready to take advantage of the integration stream. AccuRev makes it easy to move your workspace from stream to stream -- from ***acme*** *to* ***acme_int****, for example -- using simple drag-and-drop gestures.*

1. In the StreamBrowser, drag your workspace off the base stream and drop it onto the integration stream.

   The StreamBrowser shows that your workspace stream is now based on the integration stream.
2. AccuRev displays an informational message letting you know that you need to update the reparented workspace. Click **Ok** to clear the message and then update your workspace as described in the following section.

Updating Your Workspace

An Update is always required after reparenting a workspace.

To update each workspace that you reparented.

1. In the StreamBrowser or Open Workspace dialog box, double-click the workspace to open it.

   Tip: Consider displaying the Incoming Changes mode to preview the changes that will occur in your workspace after you update.

2. Click the **Update** button (Update).

Promoting Your Changes from the Integration Stream to the Base Stream

1. Select a file in your workspace.
2. Click the **Edit** button on the File Browser toolbar.

   Alternative: Right-click and choose **Edit** from the context menu.
3. Make some changes and save them.
4. Click the **Keep** button (Keep) on the File Browser toolbar.

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

   Enter a comment, and click **Ok**.
5. Click the **Promote** button (Promote) on the File Browser toolbar.

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

   This promotes the new version of the file to the integration stream. The process remains the same no matter which stream the workspace is based on: > **Edit**; > **Keep**; > **Promote**.

   Tip: You can choose to skip the Keep operation: AccuRev automatically performs any required Keep operations when you Promote a file.
6. (*optional*) **Edit**/**Keep**/**Promote** additional files.
7. Switch back to the StreamBrowser tab (example: **acme**).
8. Click the **Show Active Files** button () under the integration stream (example: **acme_int**).

   This opens a subwindow showing the files you just promoted to the integration stream.
9. Select all of the files.
10. Click the **Promote** button (Promote) on the File Browser toolbar.

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

    The files have now been promoted from the integration stream to the base stream.

Creating a Second Snapshot of the Base Stream

Your work on release 2.0 is done. Create a snapshot of the base stream to capture the versions of all the files used to create 2.0.

1. Select the base stream.
2. Click the **New Snapshot** button (New Snapshot) on the AccuRev toolbar.

   Alternative: Right-click and select **New Snapshot** from the context menu.
3. Give the second snapshot stream a name (example: **acme_2.0**).
4. Leave the Basis Time set to **Now**.

5. Click **Ok**.

   The newly created snapshot stream appears in the StreamBrowser.

Comparing the Contents of the Two Snapshot Streams

1. Select the first snapshot stream (example: **acme_1.0**).
2. Click the **Show Diff By Files** button (Show Diff By Files) on the AccuRev toolbar.

   Alternative: Right-click and select **Show Diff By Files** from the context menu.

   A "show difference" icon is added to the mouse pointer.
3. Click on the second snapshot stream (example: **acme_2.0**).

   A list appears, containing the files you promoted above.

Observing the Differences Between Snapshots at the Individual File Level

1. Select a file from the differences list you just generated.
2. Right-click and select **Show Difference** from the context menu.

Seeing Inheritance in the Stream Hierarchy

1. Go to the StreamBrowser tab (example: **acme**).
2. Drag-and-drop one workspace from its current parent, the integration stream, to the base stream. (Leave the other workspace under the integration stream.)
3. Double-click the workspace you just "reparented" to open it, then click the **Update** button (Update) on the toolbar.
4. Note the version-ID of some file in the workspace (example: **acme/1**). The direction of the separator character varies with the operating system.
5. Modify that file (right-click and choose **Edit** from the context menu).
6. Click the **Promote** button (Promote) on the File Browser toolbar.

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

   This is a shortcut—the Promote command in the GUI performs an automatic Keep on the modified file.
7. Switch back to the StreamBrowser tab.
8. Double-click the base stream to open it, then navigate to the file that you just promoted.

   Tip: The Outgoing Changes mode (Outgoing) displays all files that have been promoted from a child stream or workspace.

9. Note the new version of the file in the stream (example: **acme/2**).
10. Switch back to the StreamBrowser tab.
11. Double-click the integration stream to open it, then navigate to the file that you just promoted to the base stream.

    The version of the file in the integration stream (example: ***acme/2****) is the same as the version in the base stream. This version has been automatically inherited from the base stream.*
12. Switch back to the StreamBrowser tab.
13. Double-click the second workspace (the one that is still under the integration stream) to open it, and navigate to the same file.

    The version-ID of the file in this workspace is ***not*** *the same as in its parent, the integration stream. This fact is reflected in the file's* ***(stale)*** *status indicator. Inheritance by workspaces is not automatic, but occurs only when you Update the workspace.*
14. Click the **Update** button (Update) on the toolbar.

    The same file version (example: ***acme/2****) has now been loaded into the workspace. The file's status no longer includes the* ***(stale)*** *indicator.*

Using a Basis Time to Block Inheritance

1. Go to the StreamBrowser tab (example: **acme**).
2. Select the integration stream.
3. Right-click and select **Change Stream** from the context menu.
4. Select a Basis Time of **Now**, then click **Ok**.
5. Double-click the workspace below the base stream to open it.
6. Modify a file in the workspace (right-click and select **Edit** from the context menu).
7. Select the same file, then right-click and select **Promote** from the context menu.
8. Switch back to the StreamBrowser tab.
9. Double-click the base stream to open it, then navigate to the file that you just promoted.

    Tip: The Outgoing Changes mode (Outgoing) displays all files that have been promoted from a child stream or workspace.

    Note the version of the file in the stream (example: ***acme/3****).*
10. Switch back to the StreamBrowser tab.
11. Double-click the integration stream to open it, then navigate to the file that you just promoted to the base stream.

    The version of the file in the integration stream is ***not*** *the same as the version in the base stream. The basis time prevents automatic inheritance from the base stream.*
12. Switch back to the StreamBrowser tab.
13. Double-click the workspace below the integration stream to open it.

14. Click the **Update** button (Update) on the toolbar. Then click **View Full Log**.

    No new file versions are loaded into the workspace.

15. Switch back to the StreamBrowser tab.
16. Select the integration stream.
17. Right-click then select **Change Stream** from the context menu.
18. Select a Basis Time of **None** (cancelling the basis time setting), then click **Ok**.
19. Double-click the integration stream to open it, then navigate to the file that you just promoted to the base stream.

    *The version of the file in the integration stream (example: **acme/8**) is the same as the version in the base stream. As soon as the basis time is removed from the integration stream, it inherits the new file version from the base stream.*

20. Double-click the workspace below the integration stream, then click the **Update** button (Update) on the toolbar. Then click **View Full Log**.

    The new file version is loaded into the workspace.

Viewing the History of a Stream

1. Go to the StreamBrowser tab (example: **acme**).
2. Select the base stream.
3. Click the **Show History** button (Show History) on the AccuRev toolbar.

   Alternative: Right-click then select **Show History** from the context menu.

4. Select the most recent **promote** transaction.

   The file versions involved in the transaction appear in the bottom pane of the History Browser.

5. Experiment with changing the transaction range, searching for a comment string, etc.
6. Locate the oldest **promote** transaction, and note the timestamp for use in the next operation.

Using a Time-based Stream to Reproduce a Previous Configuration

1. Go to the StreamBrowser tab (example: **acme**).
2. Select the base stream.
3. Right-click then select **New Stream** from the context menu.
4. Specify the stream name (example: **acme_old_config**).
5. Specify the basis time by clicking the **Specified** radio button, then entering a time just after that of the base stream's first **promote** transaction (the time you noted above). Then click **Ok**.
6. Drag-and-drop the workspace from the integration stream to the newly created stream.

7. Double-click the workspace to open it, then click the **Update** button (Update) on the toolbar. Then click **View Full Log**.

   Older versions of files are loaded into the workspace, and now you are able to test the previous configuration.

8. Drag-and-drop the workspace from the newly created stream back to the integration stream.

9. Double-click the workspace to open it, then click the **Update** button (Update) on the toolbar.

   Newer versions of files are loaded into the workspace.

Making Fixes to a Previous Release

1. Go to the StreamBrowser tab (example: **acme**).

2. Select the second snapshot created (example: **acme_2.0**).

3. Click the New Stream button (New Stream) on the AccuRev toolbar.

   Alternative: Right-click then select **New Stream** from the context menu.

4. Enter a name for the new maintenance stream (example: **acme_2.0_maint**) and click **Ok**.

5. Drag-and-drop the workspace from the base stream to the new maintenance stream.

6. Double-click the workspace to open it, then click the **Update** button (Update) on the toolbar.

7. Modify some files in the workspace to implement the fix.

8. Select the Outgoing Changes mode (Outgoing) to display all files that have changed since the last time you updated your workspace.

   *All files in the workspace with **(modified)** status appear in the Details pane.*

9. Select all the modified files.

   Tip: Use the Modified status filter button (Modified) to show only those files with a **(modified)** status.

10. Click the **Promote** button (Promote) in the File Browser toolbar.

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

11. Switch back to the StreamBrowser tab.

    A default group icon appears below the new maintenance stream.

12. Click the button to display the versions that were promoted to the maintenance stream.

13. Click the icon again to close this display.

Promoting Changes from the Maintenance Stream to the Integration Stream

1. Drag-and-drop the icon below the maintenance stream to the integration stream.

   A Change Palette tab opens.

2. If any files in the Change Palette have **(overlap)** status (yellow highlight), select them and then click the **Merge** button (Merge) on the Change Palette toolbar.

3. In the pop-up window, choose the source or destination stream workspace you want to use to resolve the overlap and click **Ok**.

   The AccuRev Merge tool appears.

4. Proceed to merge the files, using the graphical Merge tool (just as in *Merging Two Versions of a File* on page 13).

5. Select all the files in the Change Palette tab, click the **Promote** button (Promote) in the File Browser toolbar.

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

   If you merged one or more files, they are displayed in a separate pane at the bottom of the Change Palette tab. You must Promote these files separately from the files in the upper pane, which did not need to be merged.

6. Close the Change Palette tab.

7. Once again, drag-and-drop the icon the maintenance stream to the integration stream.

   No files are displayed in the Change Palette tab, showing that all the changes in the maintenance stream have already been promoted to the integration stream.

8. Close the Change Palette tab.

4. Issue Management with AccuWork™

The AccuWork experience includes setting up an issues database format (“schema”), creating issue records, querying the issues database, and using the “change package” integration between configuration management and issue management.

Setting up the Default Schema

The first step is identifying the schema you want AccuWork to use to record issues. You can create your own, but let’s use the default schema to get started.

1. Select **Admin** > **Schema Editor** from the AccuRev main menu.
2. If necessary, select your depot from a list of depots in a pop-up window and click **Ok**.
3. Click **Yes** (“Do you want to use the default schema?”).
4. Click **Ok** (“Please note …”).
5. Click **Save** (lower right-corner of Schema Editor tab).

Creating a Few New Issues

Once the AccuWork schema has been set up, you are ready to submit issues for enhancements and any bugs you find.

1. Click the **Create Issue** button ( ) on the AccuRev main toolbar.

   Alternative: Select **Issues** > **Create Issue** from the AccuRev main menu.
2. Enter data in all of the required fields (red labels).
3. Experiment with how the different types of fields manage various types of data.
4. Click the **Save and Close** button ( ) to save the issue record and close the tab.
5. Repeat this process a few times if you have more issues you report.

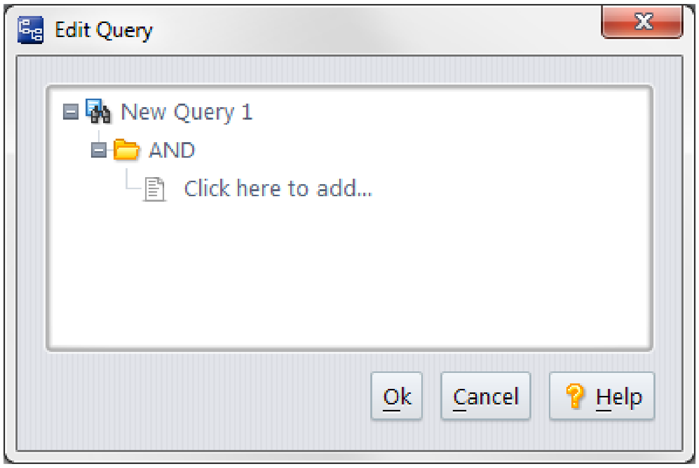
Creating a Query

1. Click the **View Issue Queries** button ( ) on the AccuRev main toolbar.

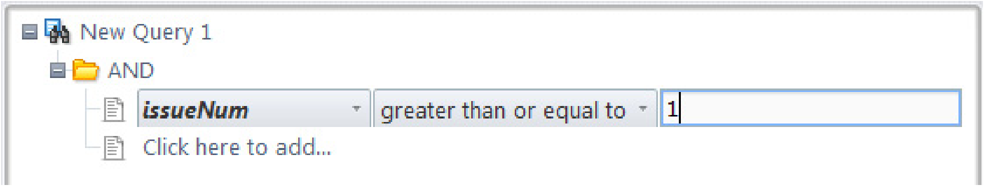
   Alternative: Select **Issues** > **View Issue Queries** from the AccuRev main menu.
2. If necessary, select your depot from the list of depots and click **Ok**.

   The AccuWork Queries tab appears; the name of the depot you selected in the previous step is displayed in the tab.
3. Click the **New Query** button ( ) to create an empty new query.

4. In the Edit Query dialog box, click **Click here to add…**.



5. Set the first choice box value to issueNum.
6. Set the second choice box value to **greater than or equal to**.
7. Enter the number **1** in the third query box.



issueNum | greater than or equal to | 1

8. Click **Ok**.

   The results of the query are displayed in the Query Results pane, on the right side of the Query tab.

   You might decide that you want to display more information about the issues that your query returns. Add more fields now.

9. Click the **Set up Columns** button (☰) in the Query Results pane.
10. Select additional fields to appear in the Query Results pane:
    a. Select a field from the **Available** list (**shortDescription**, for example).
    b. Click the right-arrow button (▶) to move the field to the **Selected** list.
    c. Repeat the steps above for all of the fields you wish to add to the query results output.
    d. Click **Ok**.
11. Give your query a name by clicking its name "New Query 1" twice (with a short pause in between) and then typing the name (for example, **All Defects**).

    The name of the new query appears in the left (Query List) pane, which lists all the queries that you have created.

12. Click the **Save** button (💾) above the Query List pane to save your query.

Making a Query Your Default Query

If you want, you can designate any query as your default query. When you do, AccuRev runs this query automatically when you open the query tab.

1. In the Query List pane, right-click the query you just created and select **Set as Default** from the context menu.
2. Close the query tab.
3. Click the **View Issue Queries** button ( ) on the AccuRev main toolbar.

   Alternative: Select **Issues** > **View Issue Queries** from the AccuRev main menu.

   The query you just created runs automatically, because you set it as your default query.

Enabling the Built-in Integration Between AccuRev Configuration Management and AccuWork Issue Management

1. Select **Admin** > **Schema Editor** from the AccuRev main menu (or return to the existing Schema Editor tab).
2. If necessary, select your depot from the list of depots and click **Ok**.
3. Go to the **Change Packages** tab.
4. In the **Change Package Results** section (top part), click the **Setup Columns** button ( ).
5. Select some fields from the **Available** list (Example: **issueNum**, **shortDescription**), and click the right-arrow button (▶) to move the fields to the **Selected** list.
6. Click **Ok**.
7. In the **Change Package Triggers** section (bottom part), click **New Trigger** button (🗋) in the toolbar.

   A new window opens, in which you specify a condition, along with a query to be performed if that condition is satisfied when the integration is invoked.
8. In the **Condition** section (top part), click **Click here to add…**.
9. Use the three choice boxes to form this condition (where ***<depot>*** is the name of your depot):

   ```
   PROMOTE_DEST_STREAM is in hierarchy <depot>
   ```
10. In the **Query** section (bottom part), click **Click here to add…**.
11. Use the choice boxes and the text field to form this condition:

    ```
    issueNum greater than 0
    ```

    This query selects all issue records.
12. Click **Ok**.
13. Click **Save** (lower right-corner of Schema Editor tab).
14. Open a workspace (or go to one that is already open).
15. Edit and Keep several files.

16. Promote all those files.

    The integration evaluates the condition you defined in Step 8 (it is TRUE), then runs the query you defined in Step10. The selected records are presented as a list of issues to choose from.

17. Select the issue that you want to associate the promoted files with, and click **Promote**.

18. Click the **Open Issue** button (📝) on the AccuRev main toolbar.

    Alternative: Select **Issues** > **Open Issue** from the AccuRev main menu.

19. Enter the number of the issue you chose when prompted (*Step 17*), and click **Ok**.

    The issue appears in a new tab.

20. Go to the **Changes** tab.

    *The files that you promoted are listed on this tab. The specific versions that you promoted are listed in the **Version** column.*

Turning Off the Integration Between Configuration Management and Issue Management

1. Select **Admin** > **Schema Editor** from the AccuRev main menu (or return to the existing Schema Editor tab)

2. If necessary, select your depot from the list of depots and click **Ok**.

3. In the **Change Packages** tab, locate the condition/query pair you created in the **Change Package Triggers** section and select it.

4. Click the **Delete Trigger** button (✖).

5. Click **Save** (lower right-corner of Schema Editor tab).

    With no query to be invoked at Promote time, the integration is effectively disabled.