AccuRev Plug-In for Visual Studio PE 2017.2

User's Guide

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

2017-10-13

Contents

Overview

This book is your guide to using AccuRev Plug-In for Visual Studio PE. It is intended for AccuRev and Visual Studio users.

AccuRev Plug-In for Visual Studio PE is a package that integrates AccuRev's source control management capabilities with Visual Studio. It enables you to execute AccuRev commands from within Visual Studio.

AccuRev Plug-In for Visual Studio PE supports all versions of Visual Studio products except the Express version.

AccuRev Plug-In for Visual Studio PE implements the SCC provider interfaces that are part of the Visual Studio Package Technology (VSPT) instead of using the Microsoft Common Source Code Control Interface (MCSCCI). This enables the integration to provide better access to AccuRev's unique software configuration management features.

AccuRev Plug-In for Visual Studio PE is referred to as AccuRev Visual Studio Plug-in in the `Market Place`.

For a short introduction to AccuRev source control concepts, see *AccuRev Usage Model*.

Changes to the Visual Studio Interface

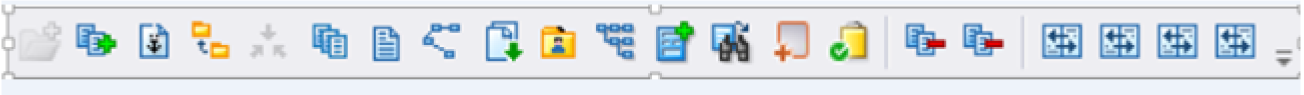
AccuRev Plug-In for Visual Studio PE extends Visual Studio's user interface in several ways:

- AccuRev toolbar.
- AccuRev submenu on the Visual Studio main menu (also available under **File** > **AccuRev**).
- AccuRev submenu on **Solution Explorer** context menus.
- AccuRev submenu on the Visual Studio editor context menu (when editing source and text files).
- AccuRev status indicators and tool-tips available in the **Solution Explorer**.
- AccuRev window, available from the Visual Studio **View** menu, containing **History**, **Searches**, and **Synchronize** tabs.
- AccuRev **Depot Explorer**, available from the Visual Studio **View** menu.
- AccuRev **Workspace Explorer**, available from the Visual Studio **View** menu.
- AccuRev pane in Visual Studio's **Output** window.

AccuRev Toolbar

AccuRev Plug-In for Visual Studio PE provides an AccuRev toolbar, which you can display with the **View** > **Toolbars** > **AccuRev** command.

Commands that are not applicable to the currently selected item(s) in the **Solution Explorer** are disabled. The **Toolbar Options** control (down-arrow) at the end of the toolbar lets you add or remove buttons.



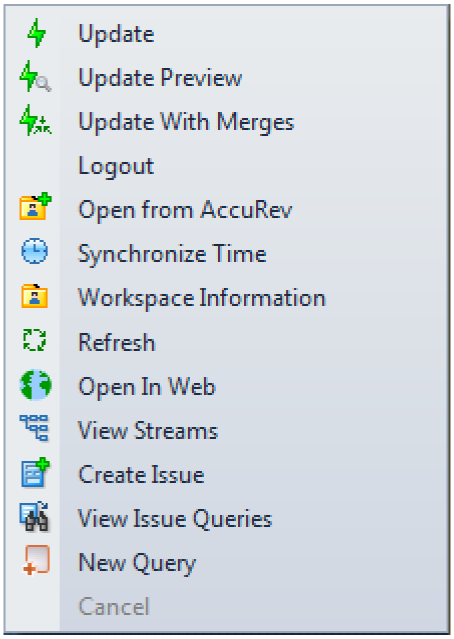
The **AccuRev** toolbar includes buttons for these AccuRev commands:

- *Add to Depot*
- *Keep*
- *Anchor*
- *Promote*

- *Merge*
- *History*
- *Annotate*
- *Version Browser*
- *Populate*
- *Workspace Information*
- *View Streams*
- *Create Issues*
- *View Issue Queries*
- *Pending Changes*
- *New Query*
- *Revert to Most Recent Version*
- *Revert to Backed Version*
- *Diff Against Most Recent Version*
- *Diff Against Backed Version*
- *Diff Against Basis Version*
- *Diff Against File on Disk*

AccuRev Menu

AccuRev Plug-In for Visual Studio PE adds an **AccuRev** submenu to Visual Studio's main menu. This submenu is also accessible through **File** > **AccuRev**. See *Commands Available on the AccuRev Menu*. The menu contains *global* AccuRev commands, which are not specific to any file selection.



The **AccuRev** submenu contains these commands:

- *Update*
- *Update Preview*
- Update with Merges
- *Login (Logout)*
- *Open from AccuRev*
- *Synchronize Time*
- *Workspace Information*

- *Refresh*
- *Open in Web*
- *View Streams*
- *Create Issue*
- *View Issue Queries*
- *New Query*
- Cancel

If some commands on this menu are disabled, either you need to log in to the AccuRev Server, or your solution is not in an AccuRev workspace. If Login fails, check your connection to the AccuRev Server process.

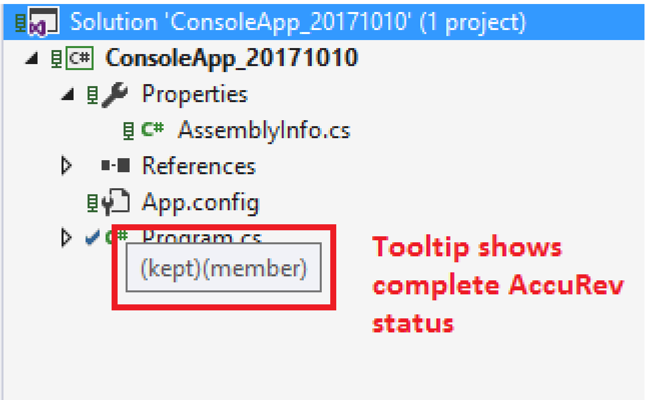
Note: You can specify the path to the AccuRev executable in the **AccuRev Executable** field on the **Configuration** tab of the **AccuRev Options** page (**Tools** > **Options** > **Source Control** > **AccuRev Options Page**). The path to your AccuRev executable usually takes the following form: `<AccuRev_install_dir>\bin\accurev.exe`. If this field is left blank, AccuRev Plug-In for Visual Studio PE uses the value in your system `PATH` environment variable. If your system `PATH` environment variable contains multiple entries for AccuRev executables, ensure that the current path appears first.

AccuRev Status Indicators

When you are working on a project located in an AccuRev workspace that belongs to you, AccuRev Plug-In for Visual Studio PE adds a decoration to each item's icon in the **Solution Explorer**, according to its AccuRev status.

An AccuRev element can have several status indicators; if you hold the cursor over an item's icon, the complete set of AccuRev status indicators appears in a tool-tip.

If your project is not within an AccuRev workspace that belongs to you, no icon decorations appear, and the tool-tip **Not in a workspace** appears when you hold the mouse over an item's icon.



Tooltip shows complete AccuRev status

This following lists the icon decoration scheme:

(modified)	white wrench
(backed)	green computer
● **(member)**	blue dot
(overlap)	white exclamation mark

Directory (folder) items do not have icon decorations. Project and solution items do have icon decorations, indicating the status of the project file and the solution file.

Getting Started

This chapter describes tasks that get you up and running with AccuRev Plug-In for Visual Studio PE.

Establishing Your Identity

All AccuRev commands must be executed by an AccuRev user. Before you can begin using AccuRev Plug-In for Visual Studio PE, you must establish your identity to AccuRev. Your AccuRev installation can use either of these user-authentication methods:

- In traditional authentication mode, AccuRev Plug-In for Visual Studio PE defaults to using your operating system username as your AccuRev username, and stores authentication information, user preferences and other information in the `.accurev` subdirectory of your home directory.

  To customize that behavior, set these two user-level (not system-level) environment variables before starting Visual Studio:

  - To use a different username, set `ACCUREV_PRINCIPAL` to that username.
  - To place your `.accurev` directory elsewhere, set `ACCUREV_ HOME` to the new location.
- In `accurev_login` authentication mode, you must explicitly log in to an AccuRev Server. You can do so using Visual Studio's **File** > **AccuRev** > **Login** command.

See the *AccuRev Administrator's Guide* for complete information on AccuRev user management.

Establishing AccuRev Source Control

The commands executed by AccuRev Plug-In for Visual Studio PE move data between the central source code repository (a depot) and a personal AccuRev work area (a workspace). To establish AccuRev source control over your Visual Studio project, it must be contained in a workspace, and the project's files must be explicitly placed under source control with the *Add to Depot* command.

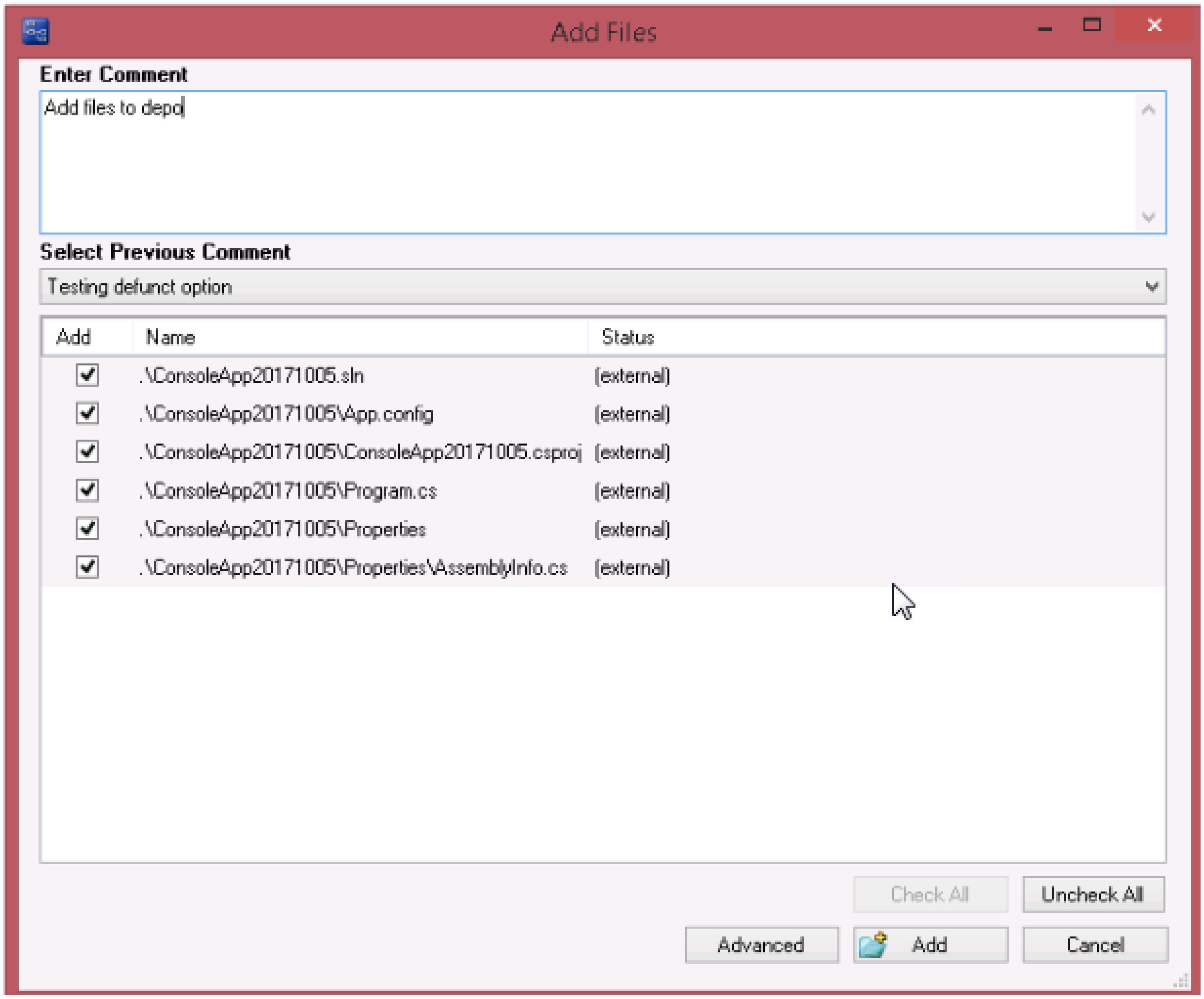
If you have been working with a previous version of AccuRev Plug-In for Visual Studio PE, installing the new version automatically updates existing Visual Studio projects or solutions already under AccuRev source control to use the new version. The sections below describe how to initialize AccuRev source control for new and existing projects. The instructions assume that any needed depots and/or streams have already been created and properly configured.

Placing an Existing Project Under Source Control

To place an existing project under AccuRev source control:

1. Open the project in Visual Studio.
2. Select the project files you want to place under AccuRev control.

   Tip: To place all of a project's files under AccuRev control, select the project folder in the Visual Studio **Solution Explorer**.
3. Right-click one of the selected files in the Solution Explorer, and choose **AccuRev** > **Add to Depot**.



Enter Comment

Add files to depo

Select Previous Comment

Testing defunct option

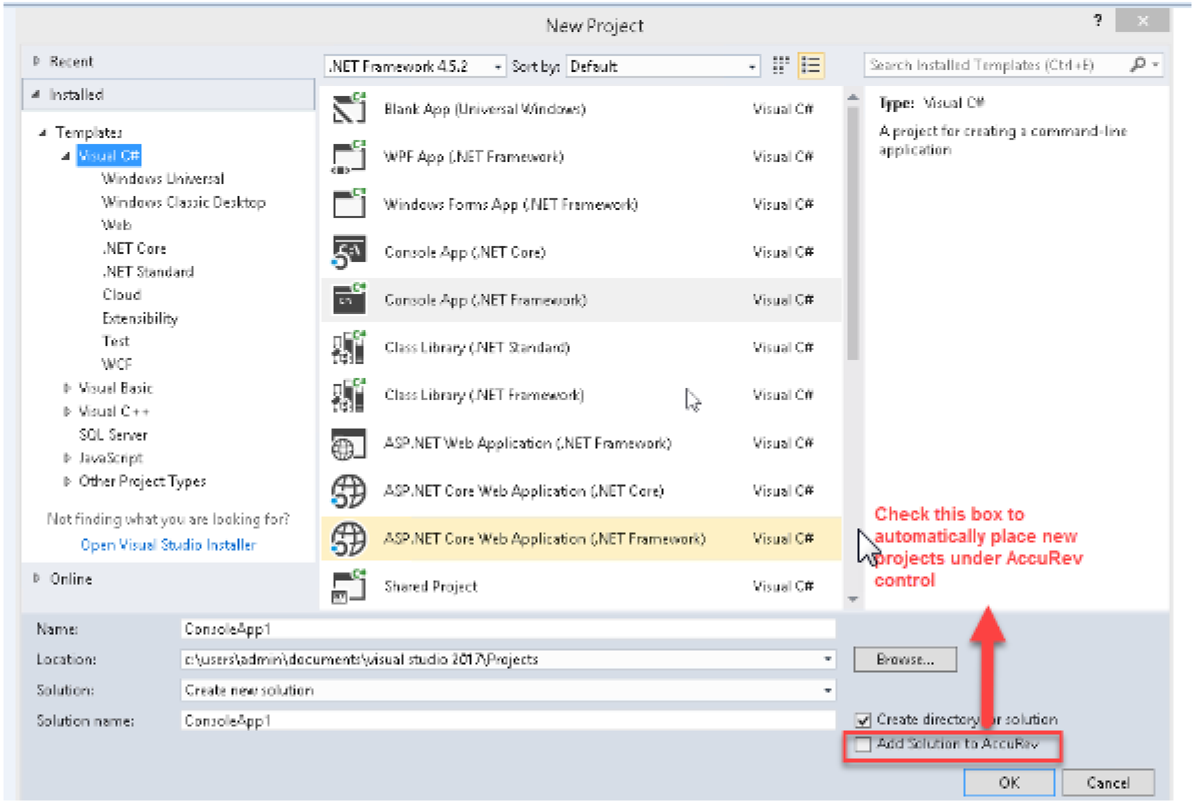
Add	Name	Status
☑	.\ConsoleApp20171005.sln	(external)
☑	.\ConsoleApp20171005\App.config	(external)
☑	.\ConsoleApp20171005\ConsoleApp20171005.csproj	(external)
☑	.\ConsoleApp20171005\Program.cs	(external)
☑	.\ConsoleApp20171005\Properties	(external)
☑	.\ConsoleApp20171005\Properties\AssemblyInfo.cs	(external)

4. Optionally, add a comment to record what you are placing under AccuRev control and why; if there are files that you do not want to place under AccuRev control at this time, clear their check boxes.
5. Click the **Advanced** button if you want to specify file type and file locking settings.
6. Click the **Add** button to add the solution files to the AccuRev depot.

Placing a New Project Under Source Control

Use this procedure to create a new Visual Studio project and place it under AccuRev control in an existing AccuRev workspace. If you want to create a new AccuRev workspace, see *Creating an AccuRev Workspace*.

1. Create a new project in Visual Studio.
2. On the **New Project** dialog box, check the **Add Solution to AccuRev** check box if it is not already selected.



New Project
Recent
Installed
Templates
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
Extensibility
Test
WCF
Visual Basic
Visual C++
SQL Server
JavaScript
Other Project Types
Not finding what you are looking for?
Open Visual Studio Installer
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
Blank App (Universal Windows) Visual C#
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Standard) Visual C#
Class Library (.NET Framework) Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
ASP.NET Core Web Application (.NET Core) Visual C#
ASP.NET Core Web Application (.NET Framework) Visual C#
Shared Project Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating a command-line application

Name: ConsoleApp1

Location: c:\users\admin\documents\visual studio 2017\Projects

Solution: Create new solution

Solution name: ConsoleApp1

☑ Create directory for solution

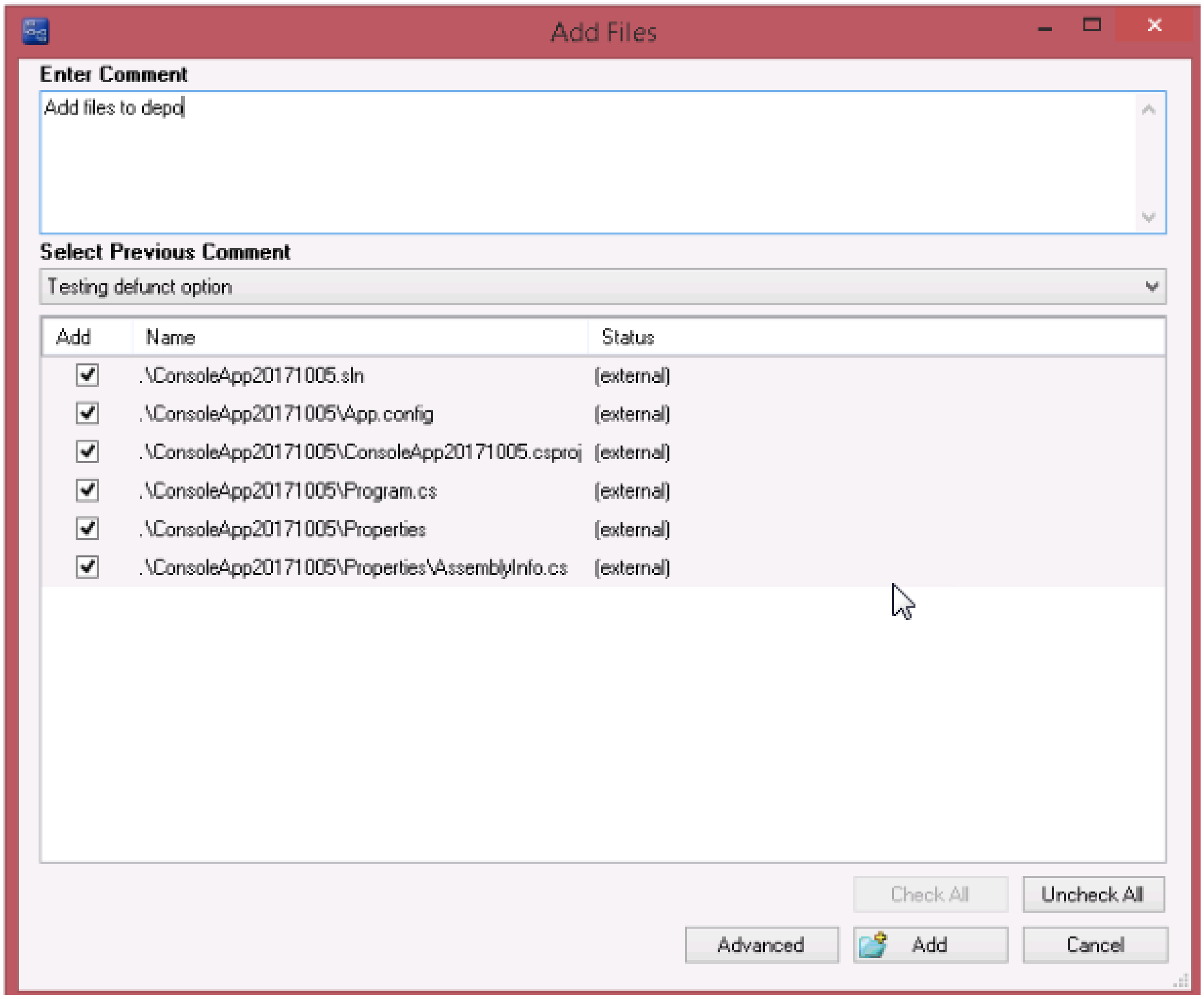
□ Add Solution to AccuRev

Check this box to automatically place new projects under AccuRev control

3. Specify a name and location for the new project.

   Tip: You must create the new project in an existing AccuRev workspace.

4. Complete the **Name**, **Location**, and **Solution** name fields as appropriate and click **OK**.



Enter Comment

Add files to depo

Select Previous Comment

Testing defunct option

Add	Name	Status
☑	.\ConsoleApp20171005.sln	(external)
☑	.\ConsoleApp20171005\App.config	(external)
☑	.\ConsoleApp20171005\ConsoleApp20171005.csproj	(external)
☑	.\ConsoleApp20171005\Program.cs	(external)
☑	.\ConsoleApp20171005\Properties	(external)
☑	.\ConsoleApp20171005\Properties\AssemblyInfo.cs	(external)

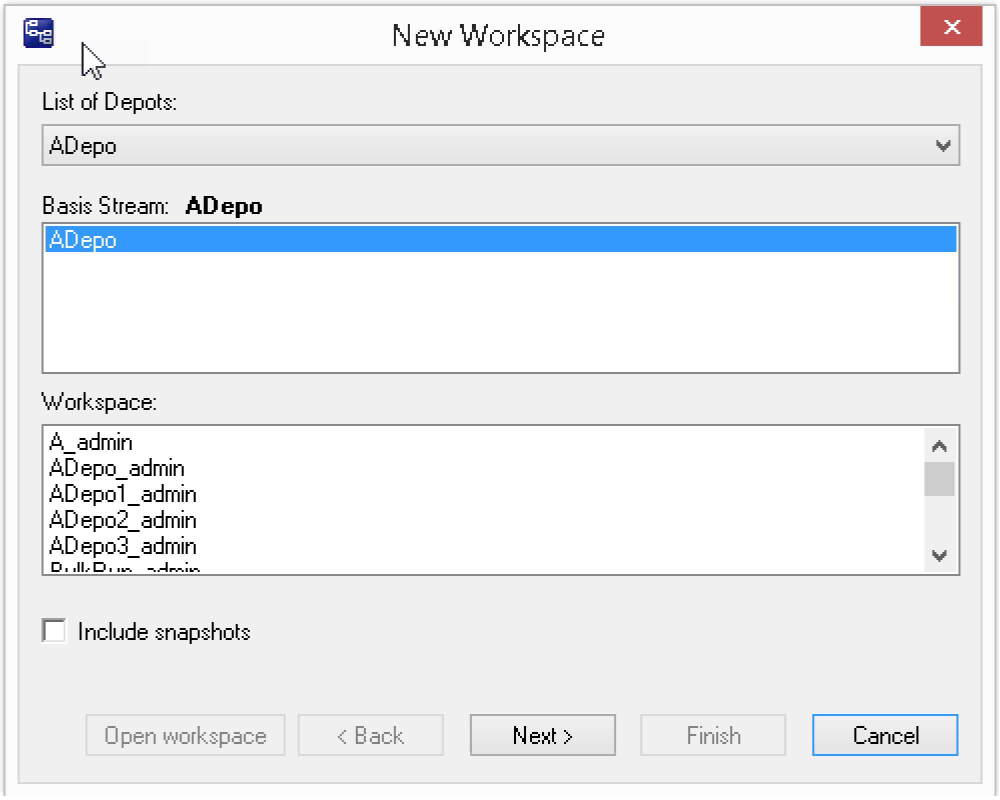
AccuRev Plug-In for Visual Studio PE displays the **Add Files** dialog box.

5. Optionally, add a comment to record what you are placing under AccuRev control and why; if there are files that you do not want to place under AccuRev control at this time, clear their check boxes.
6. Click the **Advanced** button if you want to specify file type and file locking settings.
7. Click the **Add** button to add the solution files to the AccuRev depot.

Creating an AccuRev Workspace

Use this procedure to create an AccuRev workspace using AccuRev Plug-In for Visual Studio PE:

1. Select **AccuRev** > **Open from AccuRev** from the menu. Alternative: Select **File** > **AccuRev** > **Open from AccuRev** from the menu. The first page of the **New Workspace** wizard appears.



List of Depots: ADepo

Basis Stream: **ADepo**

ADepo

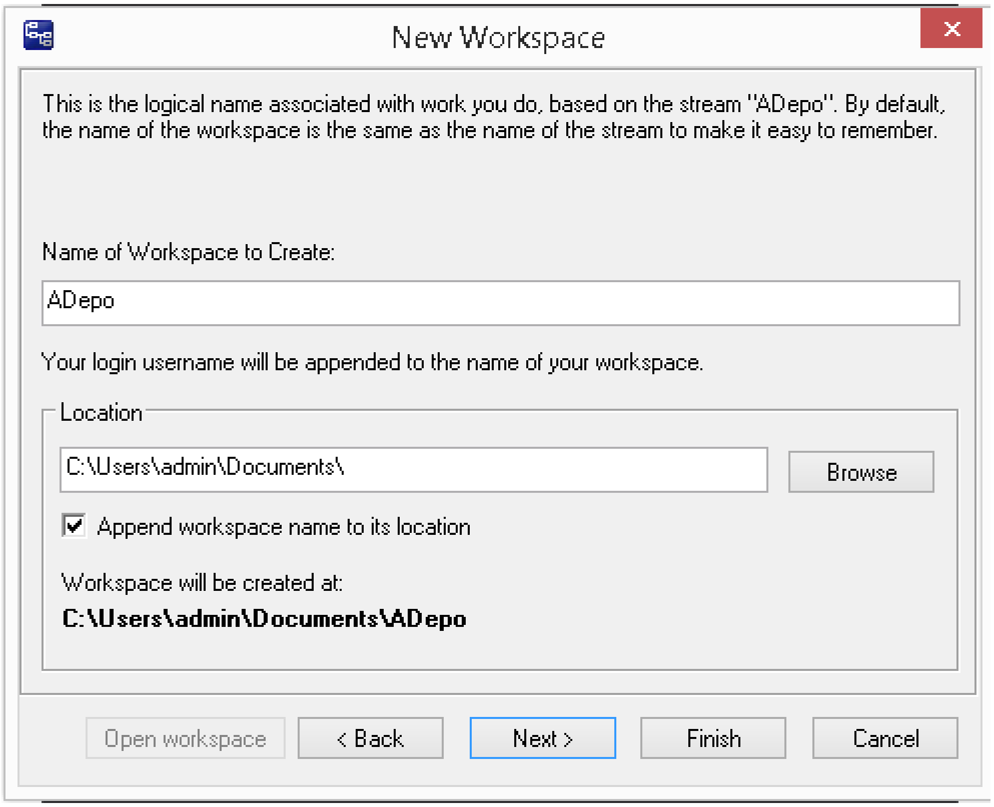
Workspace:

A_admin
ADepo_admin
ADepo1_admin
ADepo2_admin
ADepo3_admin

□ Include snapshots

2. In the **Basis Stream** list, select the AccuRev stream you want to use as the basis stream for the workspace you are creating and click **Next >**.

   The next page of the **New Workspace** wizard appears.



This is the logical name associated with work you do, based on the stream "ADepo". 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: ADepo

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

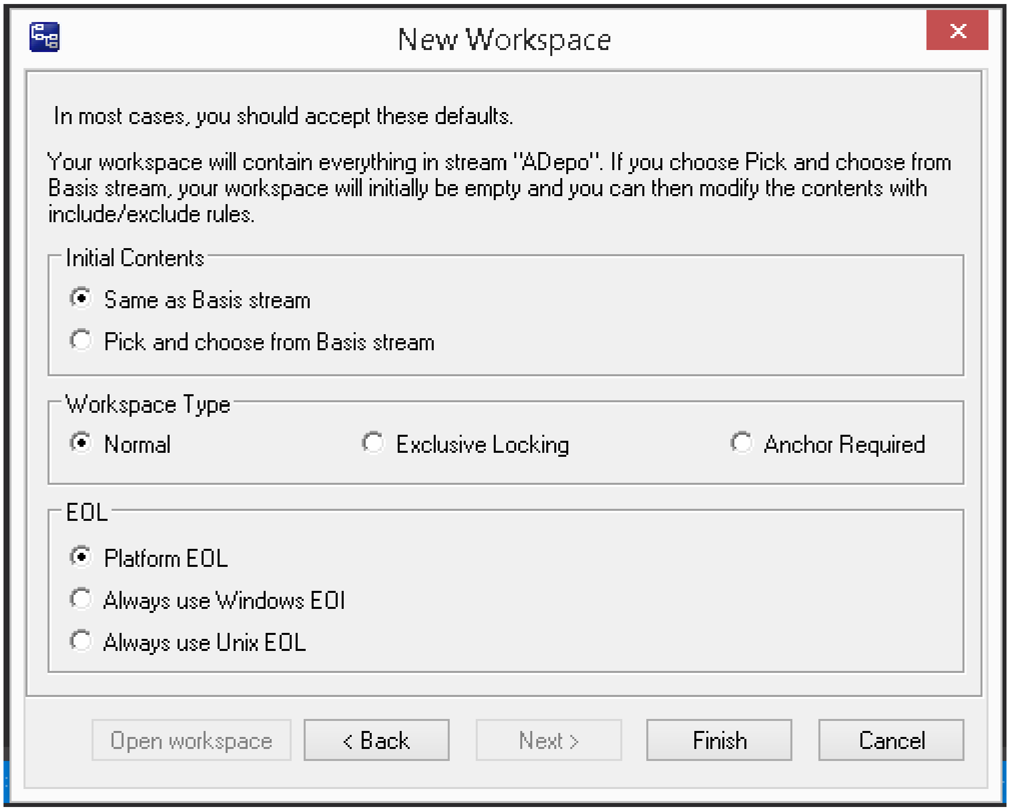
Location: C:\Users\admin\Documents\

☑ Append workspace name to its location

Workspace will be created at:
C:\Users\admin\Documents\ADepo

3. In the **Name of Workspace to Create** field, enter a name for the workspace. By default, AccuRev Plug-In for Visual Studio PE uses the name of the basis stream. Your AccuRev username is appended to the name you specify in this field. For example, if your username is `dfoster` and you specify `bluefish` as the workspace name, your workspace name is `bluefish_dfoster`.
4. In the **Location** field, specify where you want to create the workspace. You can use the default value provided, if any, or change the location as needed.
5. Optionally, change preference for appending the workspace name to its location.
6. If you want AccuRev Plug-In for Visual Studio PE to create the workspace using standard settings for contents, type, and end-of-line characters (EOL), click **Finish**. You are done. The new AccuRev workspace is ready for use. Otherwise, click **Next>**.

   The next page of the **New Workspace** wizard appears.



New Workspace

In most cases, you should accept these defaults.

Your workspace will contain everything in stream "ADepo". 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.

Initial Contents

● Same as Basis stream

○ Pick and choose from Basis stream

Workspace Type

● Normal ○ Exclusive Locking ○ Anchor Required

EOL

● Platform EOL

○ Always use Windows EOl

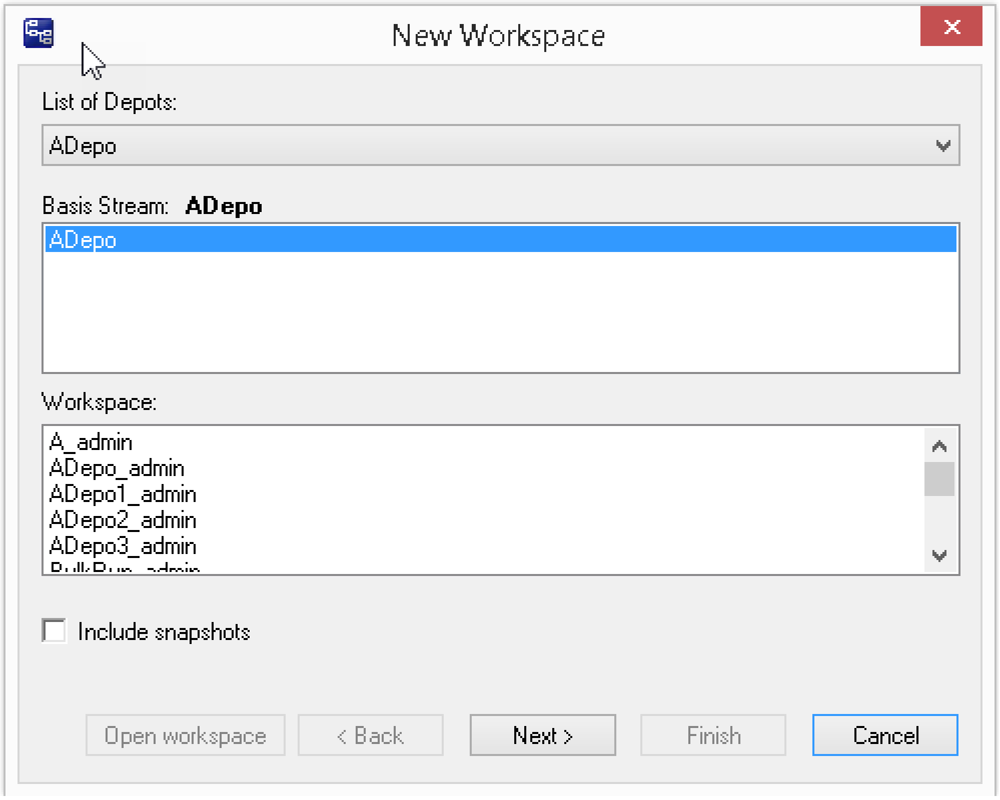
○ Always use Unix EOL

7. In the **Initial Contents** field, select **Pick and choose from Basis stream** if you want to add only specific directories from the basis stream to your new workspace. In the **Workspace Type** field, optionally specify whether you want to create the workspace with locking restrictions or anchoring requirements. If you choose either of these options, you will need to perform an `Anchor` command on a file before editing it. In the **EOL** field, specify the type of **End of Line (EOL)** symbol you want to use.
8. Click **Finish**.

Opening a Project from an AccuRev Workspace

Use this procedure to open a project under AccuRev control.

1. Select **AccuRev** > **Open from AccuRev** from the menu. Alternative: Select **File** > **AccuRev** > **Open from AccuRev** from the menu. The first page of the **New Workspace** wizard appears.



List of Depots: ADepo

Basis Stream: **ADepo**

ADepo

Workspace:

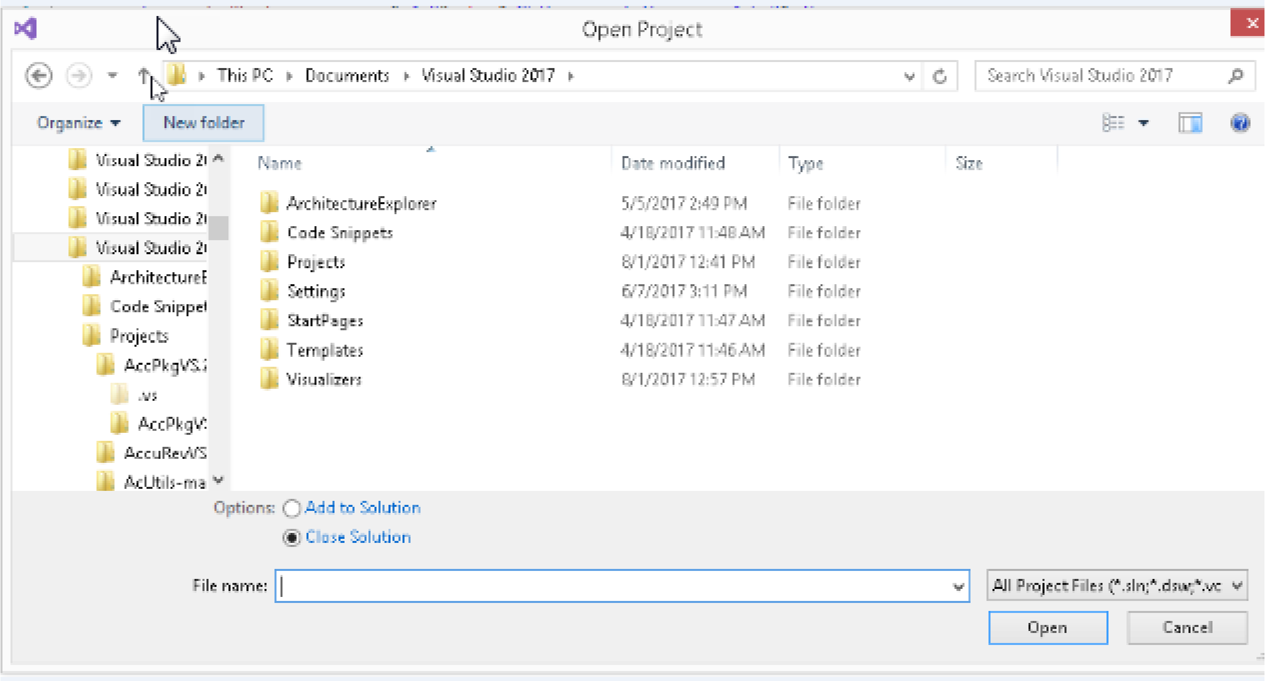
A_admin
ADepo_admin
ADepo1_admin
ADepo2_admin
ADepo3_admin
BulkRun_admin

□ Include snapshots

2. In the **Basis Stream** list, select the stream associated with the workspace that contains the project you want to open. Then, in the **Workspace** list, select the appropriate workspace.

Tip: If you want the **Basis Stream** list to also display snapshot streams, select the **Include snapshots** check box.

3. Click the **Open workspace** button. The **Open Project** dialog box appears.



Name	Date modified	Type	Size
ArchitectureExplorer	5/5/2017 2:49 PM	File folder	
Code Snippets	4/18/2017 11:48 AM	File folder	
Projects	8/1/2017 12:41 PM	File folder	
Settings	6/7/2017 3:11 PM	File folder	
StartPages	4/18/2017 11:47 AM	File folder	
Templates	4/18/2017 11:46 AM	File folder	
Visualizers	8/1/2017 12:57 PM	File folder	

Options: ○ Add to Solution
● Close Solution

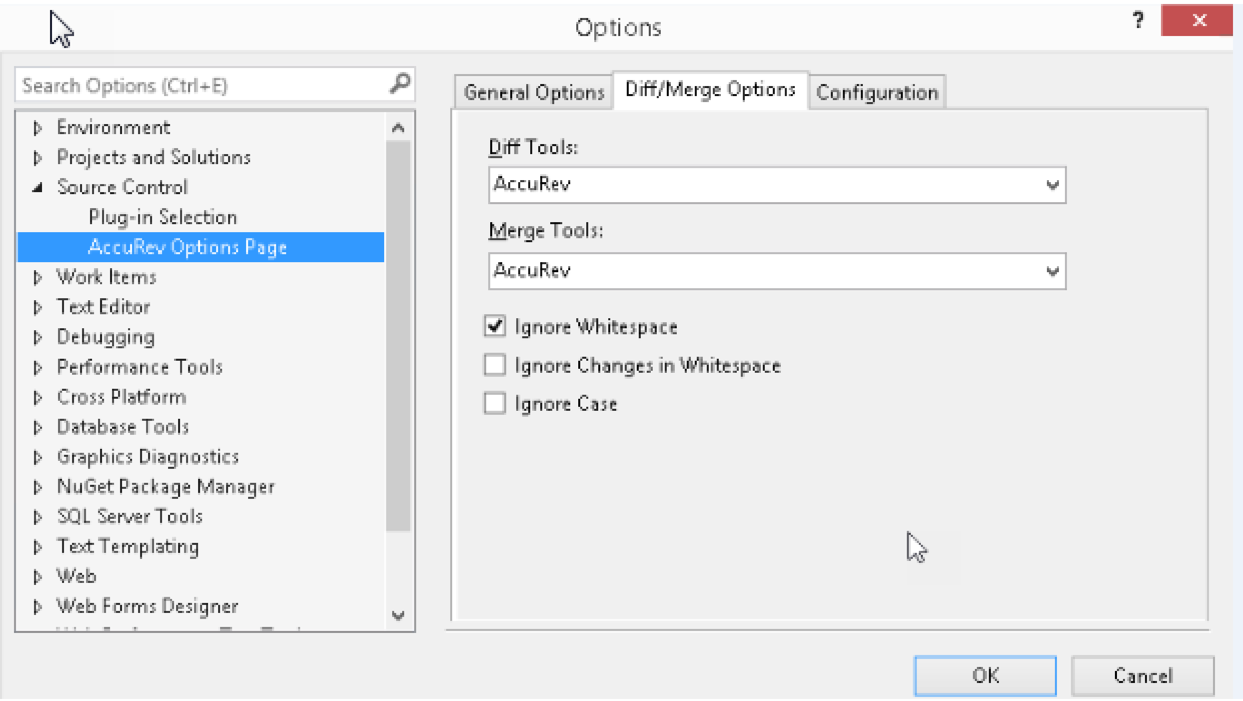
File name:

All Project Files (*.sln;*.dsw;*.vc

4. Navigate the workspace directory to locate the solution you want to open, and then click the **Open** button.

Setting AccuRev Options

Use the **Tools** > **Options** command to display the **Options** dialog. Then choose **Source Control** > **AccuRev Options Page** from the tree of option types. (If this page does not appear in the tree, select the **Show all settings** check box at the bottom of the dialog.) You can set user preferences for AccuRev's Diff/ Merge functionality, as well as more general options.



Diff Tools: AccuRev

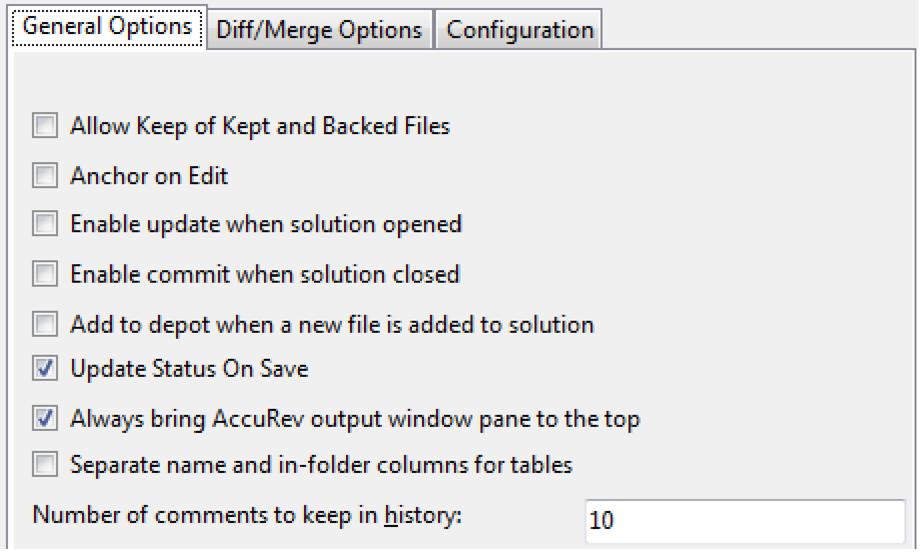
Merge Tools: AccuRev

☑ Ignore Whitespace

□ Ignore Changes in Whitespace

□ Ignore Case

General Options



□ Allow Keep of Kept and Backed Files

□ Anchor on Edit

□ Enable update when solution opened

□ Enable commit when solution closed

□ Add to depot when a new file is added to solution

☑ Update Status On Save

☑ Always bring AccuRev output window pane to the top

□ Separate name and in-folder columns for tables

Number of comments to keep in history: 10

Allow Keep of Kept and Backed Files Controls whether files with statuses of `(kept)` and `(backed)` appear in the **Keep** dialog when you invoke the *Keep* command.

Anchor on Edit Causes an AccuRev `Anchor` command to be performed automatically when you start to edit a file element in an anchor-required or exclusive-file-locking workspace. If the option is disabled, an `Anchor` command is performed when you click **File** > **Save**.

Enable update when solution opened Causes an `Update` command (from the **Synchronize** tab of the **AccuRev** window) to be performed automatically when you open a Visual Studio solution.

Enable commit when solution closed Causes a Commit command (from the **Synchronize** tab of the **AccuRev** window) to be performed automatically when you close a Visual Studio solution.

Add to depot when a new file is added to solution Causes an AccuRev Add command to be invoked automatically when you add a new object to a Visual Studio project (**Add** > **New Item**, **Add** > **Existing Item**, or **Add** > **New Folder**).

Update Status on Save Causes an automatic call to the AccuRev Server whenever you execute the Visual Studio command **File** > **Save** on a file. This determines the new AccuRev status of the saved file and updates its icon decoration. This option is turned on by default. Turning it off improves performance by eliminating the call to the AccuRev Server. To display the correct icon decoration for the saved file when the option is turned off, invoke the command **AccuRev** > **Refresh**.

Always bring AccuRev output window pane to the top Ensures that the **AccuRev** pane in the **Output** window is always brought to the top when you have more than one tab displayed.

Separate name and in-folder columns The pathname is split over two columns: a **Name** column with the “leaf” name, and an **In Folder** column with the folder/directory pathname to the element. If this option is not selected, entire pathname appears in a single column, labeled Name.

Number of comments to keep in history The number of choices to appear in the **Select Previous Comment** list in the dialog boxes for the `Add to Depot`, `Keep`, and `Promote` commands.

Diff/Merge Options

This set of options allows you to specify the text file comparison and merge tools that AccuRev Plug-In for Visual Studio PE will use, as well as how certain characters are treated if you use AccuRev’s tools.

When specifying values for the **Diff Tools** and **Merge Tools** fields, you have several options:

- Use AccuRev’s own tools — By default, the setting is `AccuRev`, indicating the graphical diff and merge tools that are part of the AccuRev product.
- Choose another tool from the list — The change affects AccuRev Plug-In for Visual Studio PE only; it does not affect the AccuRev GUI (which can also be configured to use third-party diff and merge tools).
- Specify a command line to be executed when the `Diff` or `Merge` command is invoked. This command line must contain the name of a text file comparison or merge tool, and a series of substitution patterns which indicate the files being compared or merged. The tool must either be in a directory on your search path, or be specified with an absolute path.

Note: These patterns look like Microsoft Windows batch file variables, but they are not. They are merely placeholders that AccuRev Plug-In for Visual Studio PE will replace with character strings.

Diff Substitution Patterns

%1% When the command line is executed, this pattern will be replaced by the quoted pathname of the first version to be compared.

%2% Will be replaced by the quoted pathname of the second version to be compared.

%3% Will be replaced by a displayable title string (unquoted) for the first version.

%4% Will be replaced by a displayable title string (unquoted) for second version.

Merge Substitution Patterns

%a% When the command line is executed, this pattern will be replaced by the quoted pathname of the closest common ancestor version.

%1% Will be replaced by the quoted pathname of the non-workspace version.

%2% Will be replaced by the quoted pathname of the version in the workspace.

%3% Will be replaced by a displayable title string (unquoted) for the closest common ancestor version.

%4% Will be replaced by a displayable title string (unquoted) for the non-workspace version.

%5% Will be replaced by a displayable title string (unquoted) for the workspace version.

%o% Will be replaced by the quoted pathname of a temporary file, in which the results of the merge can be stored.

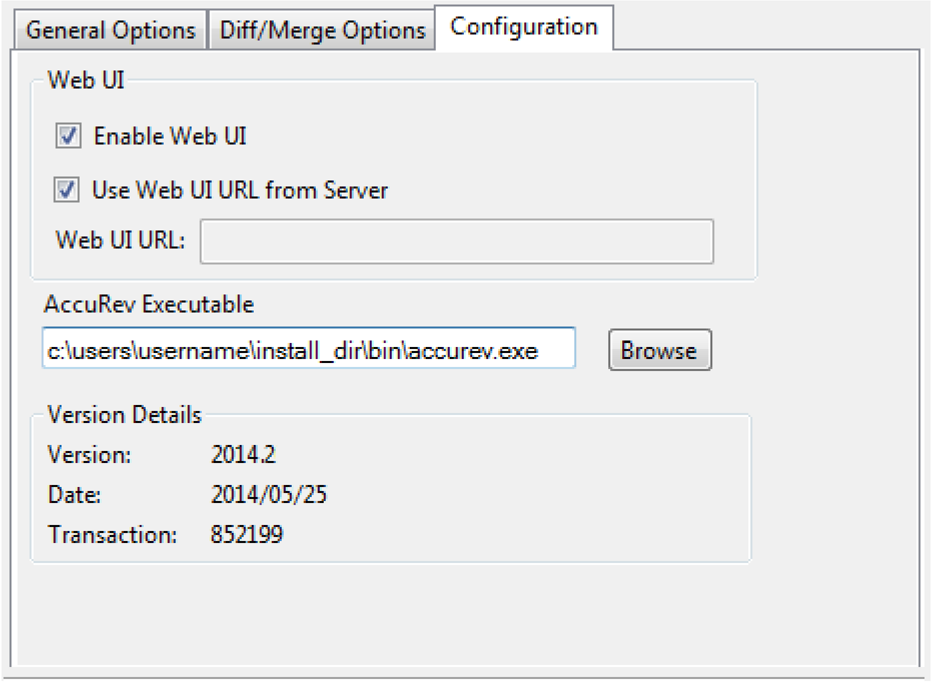
The next three options affect AccuRev's graphical `Diff` and `Merge` tools only. If you configure other text file comparison or merge tools, these settings are ignored.

Ignore Whitespace Controls whether whitespace is taken into account when comparing text lines. Overrides the **Ignore Changes in Whitespace** option.

Ignore Changes in Whitespace Controls whether a change in the amount of whitespace in a text line is considered to be a change.

Ignore Case Controls whether uppercase and lowercase characters are considered to be the same when comparing text lines.

Configuration Options



☑ Enable Web UI

☑ Use Web UI URL from Server

Web UI URL:

AccuRev Executable

c:\users\username\install_dir\bin\accurev.exe

Version Details

Version: 2014.2

Date: 2014/05/25

Transaction: 852199

Enable Web UI Enables or disables the AccuRev Web UI inside Visual Studio.

Use Web UI URL from Server

Use the URL specified in the AccuRev Server settings rather than the one specified in the **Web UI URL** field on this tab. The URL is stored in the `storage\site_slice\dispatch\config\settings.xml` file on the AccuRev Server:

```
<settings>
<webui url="https://<server>.web.<domain>:<port>/accurev" />
</settings>
```

Web UI URL

Specifies a URL to use to access the AccuRev Web UI. If the **Use Web UI URL from Server** check box is selected, the URL stored on the AccuRev Server is used instead of the one specified here.

AccuRev Executable

Specifies the path of the AccuRev executable. You can enter the path manually, or use the **Browse...** button to navigate to the executable's location. If this field is left blank, AccuRev Plug-In for Visual Studio PE uses the value in your system `PATH`.

Version Details

Information about the current AccuRev Plug-In for Visual Studio PE version:

Version	The currently installed version.
Date	The date this version was built.
Transaction	The AccuRev transaction number associated with the version's build.

Error Logs

Log files record the reason for errors during any AccuRev operation. The log generated can be configured to the following levels:

- `DEBUG`
- `INFO`
- `WARN`
- `ERROR`
- `FATAL`

Configuring Error Log

To configure the level of the log files:

1. Navigate to `C:\Users\<`*`username`*`>\AppData\Local\Microsoft\VisualStudio\<15.0_8ec0ccb7>\Extensions\<yb05aubz.rce>`.

   Note: `<15.0_8ec0ccb7>` and `<yb05aubz.rce>` may vary in the above path based on the version of Visual Studio installed. The remaining path remains constant, *username* is the account name with which the user logs in to the system.
2. Open the `App.config` file.
3. Replace OFF in `<level value=OFF>` with the required value from(`DEBUG`, `INFO`, `ERROR`, `WARN` and `FATAL`).

   Note: Error logs are not generated if `<level value>` is set to `"OFF"/`.
4. Save the `App.config` file.

Error logs are generated based on the value that you set.

Error Log Location

The error log file(s) are located in `C:\Users\<`*`username`*`>\AppData\Local\AccuRevVS`. They have a `.log` extension.

Note: *username* is the account name with which the user logs in to the system.

For example: `AccurevLog.log`.

AccuRev-Specific Windows

This section describes AccuRev Plug-In for Visual Studio PE's use of windows within the Visual Studio interface:

- A special **AccuRev** window, containing several tabs for executing, and displaying the results of, AccuRev commands.
- An **AccuRev Depot Explorer** window.
- An **AccuRev Workspace Explorer** window.
- The **AccuRev Query Editor**.
- An **AccuRev** pane in Visual Studio's standard **Output** window.

AccuRev Window

AccuRev Plug-In for Visual Studio PE provides an **AccuRev** window, in which you can view the results of AccuRev commands and run AccuRev commands. To display this window, use the **View** > **AccuRev Window** command. It appears automatically when you invoke AccuRev's `History` command in the **Solution Explorer**.

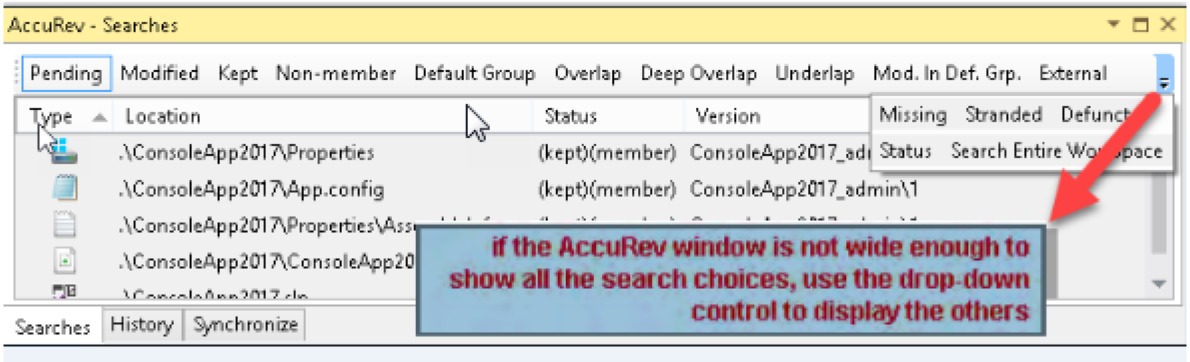
As long as the current solution remains open, you can close and reopen the AccuRev window without losing information.

The **AccuRev** window contains three tabs: *Searches*, the *History*, and the *Synchronize*. Each tab contains a table which you can manipulate in the following ways:

- Clicking a column header sorts the table's row on that column's contents. Clicking again reverses the sort order, indicated by a triangle.
- Drag column headers to the left or right to change the column order.
- Drag column separators to change column widths.

Searches Tab

In the **Searches** tab, you can search for items based on their AccuRev status, and execute commands on search results items.



Location	Status	Version
.\ConsoleApp2017\Properties	(kept)(member)	ConsoleApp2017_adı
.\ConsoleApp2017\App.config	(kept)(member)	ConsoleApp2017_admin\1
.\ConsoleApp2017\Properties\Ass		
.\ConsoleApp2017\ConsoleApp20		
.\ConsoleApp2017.sln		

if the AccuRev window is not wide enough to show all the search choices, use the drop-down control to display the others

To perform a search, click a button on the tab's toolbar. The items that satisfy the search criterion are displayed in a table, with these columns:

Type	Visual Studio file type.
Location	Location within the workspace.

Status AccuRev status indicator(s).

Version AccuRev version-ID.

Overlap Stream AccuRev stream containing the overlapped version.

The tab’s toolbar includes these additional buttons:

- Click **Status** to display the status of all items in the workspace.
- The **Search Entire Workspace** button is a toggle switch that controls the scope of the search. If the toggle is on (highlighted), the results table includes matching items in all the workspaces that contain projects in the current solution. If the toggle is off (not highlighted), the results table only includes matching items in the current solution.

Updating Search Results

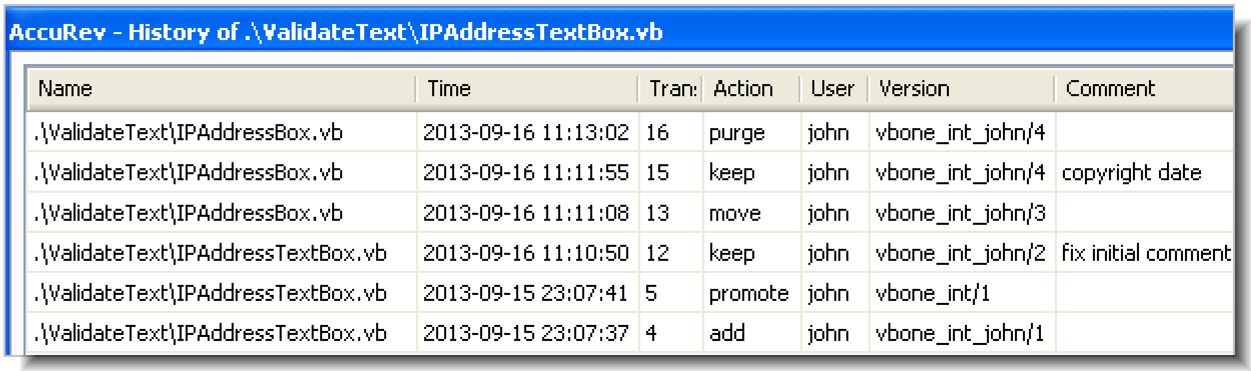
Clicking a button again reruns the search. If you change an element’s status within Visual Studio (for example, by editing a file) while the **Searches** tab is loaded with search results, the results table is updated automatically.

Running AccuRev Commands

You can run AccuRev commands in the **Searches** tab by selecting one or more items, then right-clicking to bring up a context menu. A command in enabled only if it applies to at least one of the selected items. The **Searches** tab commands are the same as those available in the **Solution Explorer**, with the addition of *Defunct* and *Locate in Solution Explorer*. See *AccuRev Command Reference* for full descriptions of the commands.

History Tab

The **History** tab displays a table containing the results of a *History* command. Each row lists an AccuRev transaction that created a new version of the element.



AccuRev - History of .\ValidateText\IPAddressTextBox.vb

Name	Time	Tran:	Action	User	Version	Comment
.\ValidateText\IPAddressBox.vb	2013-09-16 11:13:02	16	purge	john	vbone_int_john/4	
.\ValidateText\IPAddressBox.vb	2013-09-16 11:11:55	15	keep	john	vbone_int_john/4	copyright date
.\ValidateText\IPAddressBox.vb	2013-09-16 11:11:08	13	move	john	vbone_int_john/3	
.\ValidateText\IPAddressTextBox.vb	2013-09-16 11:10:50	12	keep	john	vbone_int_john/2	fix initial comment
.\ValidateText\IPAddressTextBox.vb	2013-09-15 23:07:41	5	promote	john	vbone_int/1	
.\ValidateText\IPAddressTextBox.vb	2013-09-15 23:07:37	4	add	john	vbone_int_john/1	

The table contains these columns:

Name Element's pathname within the workspace.

Time A timestamp indicating when the transaction took place.

Transaction The unique number (within this AccuRev depot) of the transaction.

Action The kind of AccuRev transaction: keep, promote, etc.

User AccuRev username of the user who performed the transaction.

Version Unique version-ID of the element version that was created in this transaction.

Comment User-supplied comment for this transaction.

Running AccuRev Commands

You can select one or two transactions from an item's history, then invoke an AccuRev command from the selection's context menu:

File in the Workspace Compares the version in the selected transaction with the version currently in your project.

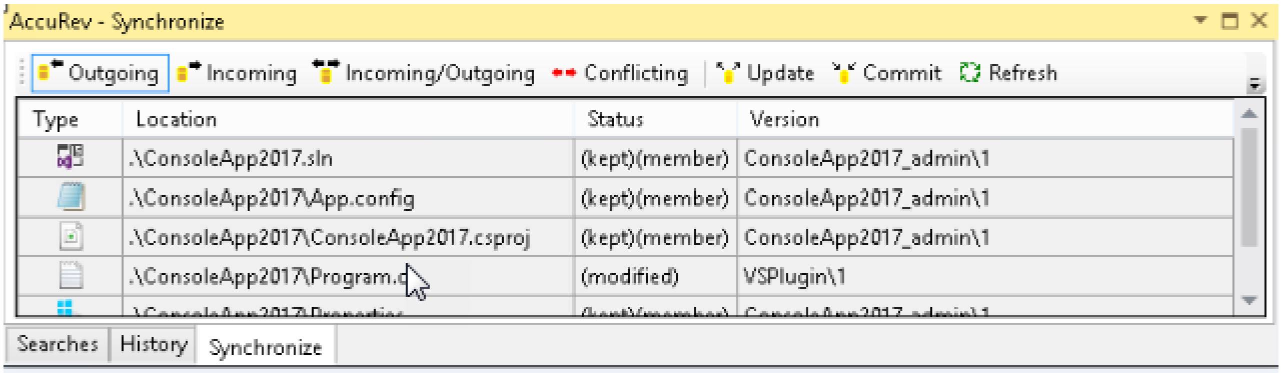
Other Version Compares the versions created in the two selected transactions.

Checkout Version Replaces the version in your workspace with the version in the selected transaction.

Show Transaction Pops up another window , **Transaction History**, showing the complete contents of the selected transaction.

Synchronize Tab

The **Synchronize** tab shows an element-by-element listing that answers the question: What are the current differences between my project(s) and the backing stream(s) in the AccuRev repository? The **Synchronize** tab's toolbar provides command buttons to filter project files by how they differ from the backing-stream versions in the AccuRev repository:



Type	Location	Status	Version
	.\ConsoleApp2017.sln	(kept)(member)	ConsoleApp2017_admin\1
	.\ConsoleApp2017\App.config	(kept)(member)	ConsoleApp2017_admin\1
	.\ConsoleApp2017\ConsoleApp2017.csproj	(kept)(member)	ConsoleApp2017_admin\1
	.\ConsoleApp2017\Program.c	(modified)	VSPlugin\1
	.\ConsoleApp2017\Properties	(kept)(member)	ConsoleApp2017_admin\1

Incoming The element changed in the repository, but not in the project. Objects with AccuRev status `(stale)`, `(overlap)`, `(underlap)`, or `(missing)` are selected by this filter.

Outgoing The element changed in the project, but not in the repository. Objects with AccuRev status `(modified)`, (kept), (member), `(external)`, `(overlap)`, `(underlap)`, or `(missing)` are selected by this filter.

> **Note:** The AccuRev command *Anchor* changes a file's AccuRev status to `(member)`, but does not affect the file's contents. Such a file is classified as `Outgoing` in the **Synchronize** tab, even though there might be no difference between the file's contents in the project and in the repository.

Incoming/ Outgoing Shows elements from both the Incoming and Outgoing categories.

Conflicting Shows elements that have changed in both the project and the repository. Objects with AccuRev status `(overlap)` or `(underlap)` are selected by this filter.

Two additional buttons affect the display of elements in the Synchronize tab:

Refresh Retrieves the AccuRev status of all the files in the project, and updates the current display.

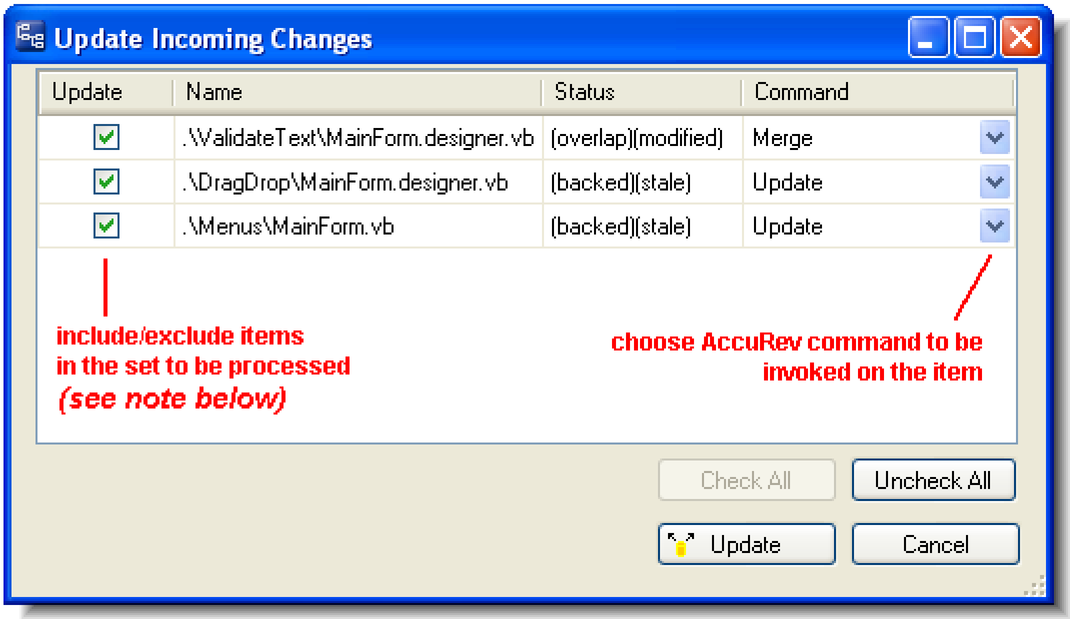
Sync Entire Workspace Toggles the scope of the **Synchronize** tab. If the toggle is on (highlighted), the tab includes items from the entire AccuRev workspace containing the current solution. And if the solution contains multiple projects that are contained in multiple workspaces, the tab includes items from the entirety of all those workspaces.

If the toggle is off (not highlighted), only items that are contained in the current solution are displayed.

Running AccuRev Commands to Synchronize the Project with the Reposit

The **Synchronize** tab’s toolbar includes two command buttons that initiate AccuRev commands to bring your project into closer (or complete) synchronization with your workspace’s backing stream.

Update Opens an **Update Incoming Changes** window, in which you can bring changes into your project, with the AccuRev commands *Merge*, *Revert to Backed Version*, and *Update*.



Update Incoming Changes

Update	Name	Status	Command
☑	.\ValidateText\MainForm.designer.vb	(overlap)(modified)	Merge
☑	.\DragDrop\MainForm.designer.vb	(backed)(stale)	Update
☑	.\Menus\MainForm.vb	(backed)(stale)	Update

include/exclude items in the set to be processed *(see note below)*

choose AccuRev command to be invoked on the item

You can process an (overlap)-status item with either *Merge* or *Revert to Backed Version*. The only command available for a (stale)-status item is Update.

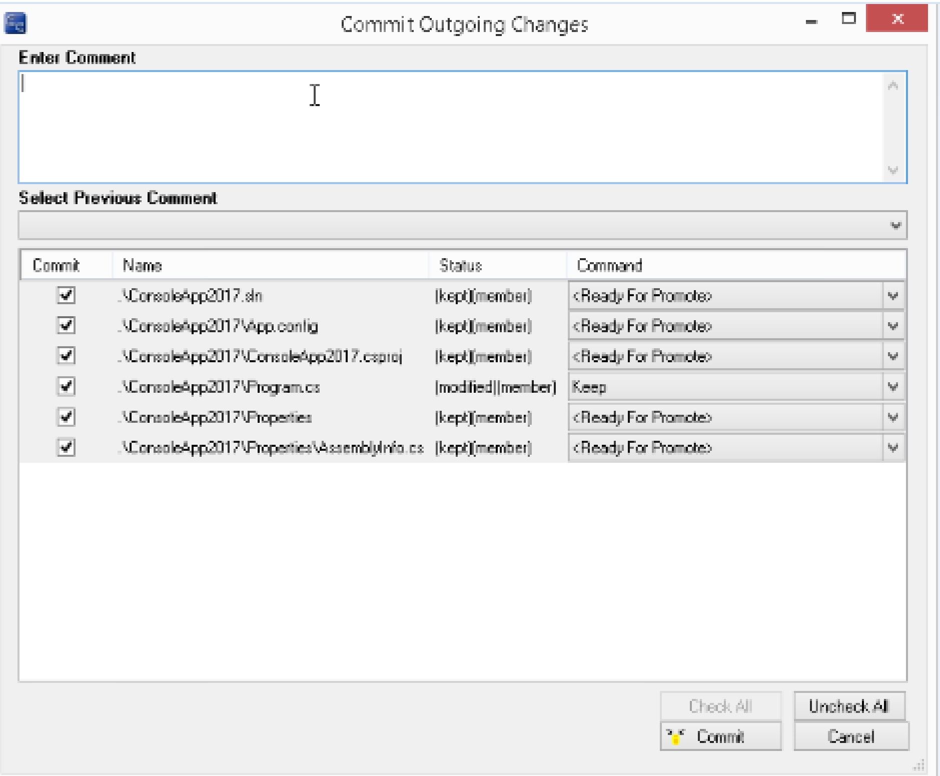
Clicking the **Update** button at the bottom of this window launches a series of AccuRev commands to perform the operations you’ve configured.

Note: The AccuRev *Update* command operates on the entire workspace, not on individual elements within it. Accordingly, clearing the checkbox for an item to be processed with the `Update` command has no effect.

Commit Pops up a **Commit Outgoing Changes** window, in which you can send your changes to the repository, using the AccuRev commands, *Add to Depot*, *Merge*, and *Promote*. (You can also choose to undo changes to files, using the *Revert to Backed Version* and *Revert to Most Recent Version* commands.)

Note: For items with `(member)` status, the **Keep** choice in the **Command** column performs a *Keep* and *Promote* sequence.

Clicking the **Commit** button at the bottom of this window launches a series of AccuRev commands to perform the operations you’ve configured.



Commit	Name	Status	Command
☑	.\ConsoleApp2017.sln	(kept)(member)	<Ready For Promote>
☑	.\ConsoleApp2017\App.config	(kept)(member)	<Ready For Promote>
☑	.\ConsoleApp2017\ConsoleApp2017.csproj	(kept)(member)	<Ready For Promote>
☑	.\ConsoleApp2017\Program.cs	(modified)(member)	Keep
☑	.\ConsoleApp2017\Properties	(kept)(member)	<Ready For Promote>
☑	.\ConsoleApp2017\Properties\AssemblyInfo.cs	(kept)(member)	<Ready For Promote>

Manual Synchronization

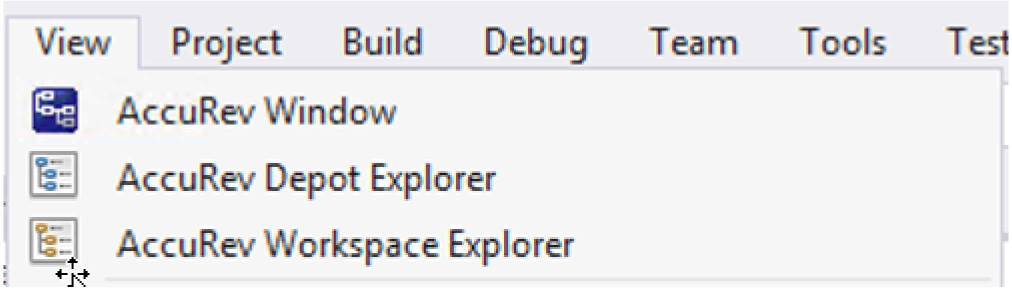
Instead of clicking the **Commit** or **Update** button in the **Synchronize** tab, you can invoke AccuRev commands on specific items: select one or more items, then right-click to bring up a context menu. A command is enabled only if it applies to at least one of the selected items. See *AccuRev Command Reference* for full descriptions of the AccuRev commands.

AccuRev Depot Explorer

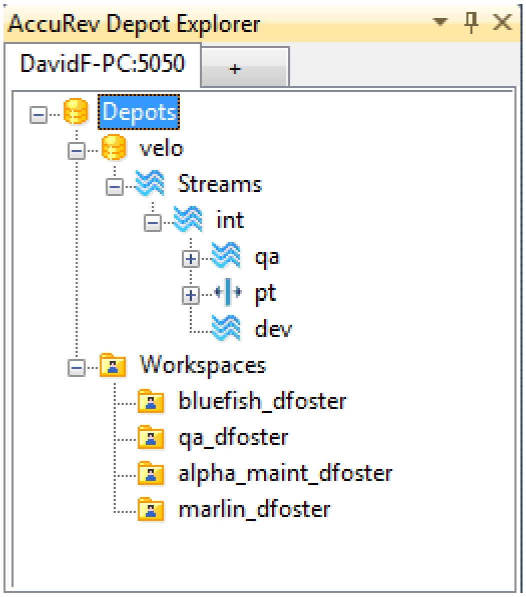
Use the AccuRev **Depot Explorer** to perform the following operations:

- Create a read-only navigable list of depots.
- Create a navigable tree of the streams and workspaces in an AccuRev depot.
- Create a new dynamic stream, snapshot stream, or pass-through stream.
- Create a new workspace.
- Change a stream’s name, its basis stream, or its basis time.

To display this window, use the **View** > **AccuRev Depot Explorer** command.



The AccuRev **Depot Explorer** is a dockable window that allows you to create a read-only navigable list of depots, as shown in the following illustration:



To create a navigable tree of the streams and workspaces in an AccuRev depot:

1. Open a solution that is known to AccuRev.

   Tip: This solution can be on any AccuRev Server, not necessarily the one for which you want to create the depot explorer.

2. Choose **View** > **AccuRev Depot Explorer** from the Visual Studio menu. An empty **Depot Explorer** window appears.
3. Click the plus symbol at the top of the tab. The **Login to AccuRev Server** dialog box appears.
4. Log in to the AccuRev Server for which you want to create the depot explorer tree and click **Ok**.

   Tip: This login is used solely to create the **Depot Explorer** tree. It does not interfere or change the AccuRev Server you are currently logged in to through Visual Studio. For example, you can currently be logged in to the AccuRev Server at `falcon:5050` and create a depot explorer tree for the AccuRev Server at `kestrel:5050`.

   A Depots symbol appears in the **Depot Explorer** window.
5. Expand the nodes you wish to explore by clicking that node's plus symbol.

To close a depot explorer tree, right-click the tab and choose **Close**.

To refresh the **Depot Explorer** tree, right-click the depot or stream and choose **Refresh**.

Depot Explorer trees are not saved when you close Visual Studio.

Creating Streams

Use this procedure to create a new dynamic stream, snapshot stream, or pass-through stream from the **Depot Explorer**.

1. Select the depot or stream off which you want to create the new stream.
2. Right-click and choose **New Stream** from the context menu.

   Tip: If you want to create a snapshot stream, choose **New Snapshot**.

3. In the **Name** field of the dialog box that appears, enter a name for the new stream.
4. In the **Stream Type** field, choose the type of stream you want to create.
5. In the **Basis Time** field, specify the watermark you want to use when creating this stream. Note that you cannot specify a basis time for pass-through streams.

6. Click **Ok**.

Creating Workspaces

To create a new workspace from the **Depot Explorer**:

1. Select the stream.
2. Right-click and choose **New Workspace**.
3. Follow instructions in the **New Workspace** dialog box to create the new workspace.
4. Click **Finish** to create the new workspace using default settings for locking, workspace contents, and end-of-line characters.

   Tip: The **Next** button is active only if you customize the new workspace by changing one of the default settings.
5. Right-click and choose **Refresh** to view the new workspace.

Changing Streams

To change a stream's name, its basis stream, or (for dynamic and time basis streams only) its basis time from the **Depot Explorer**:

Tip: Changing the basis stream has the same effect as reparenting.

Note: This procedure does not apply to snapshot streams.

1. Select the stream.
2. Right-click the stream and choose **Change Stream**.
3. Optionally change one or more of the following: **Name**, **Basis stream**, **Basis time** (dynamic and time basis streams only).
4. Click **Ok** to complete the change.

Reparenting Streams and Workspaces

You can reparent a dynamic, pass-through, or time-basis stream or workspace using drag-and-drop or the context menu from the **Depot Explorer**.

To reparent a stream using drag-and-drop:

1. Press and hold mouse button 1 on the stream or workspace you want to reparent.
2. Drag the mouse and place the pointer over the stream you want to use as the new backing stream.
3. Release mouse button 1.
4. To reparent a stream or workspace using the context menu:
   a) Select the or workspace you want to reparent.
   b) Right-click the stream and choose **Reparent** from the context menu. The pointer changes shape.
   c) Click the stream you want to use as the stream's or workspace's new backing stream.
   d) Click **Ok** to dismiss the informational message and complete the reparenting operation.

File View

The **File** view in the **Depot Explorer** displays the files associated with a selected stream or workspace.

To display the **File** view:

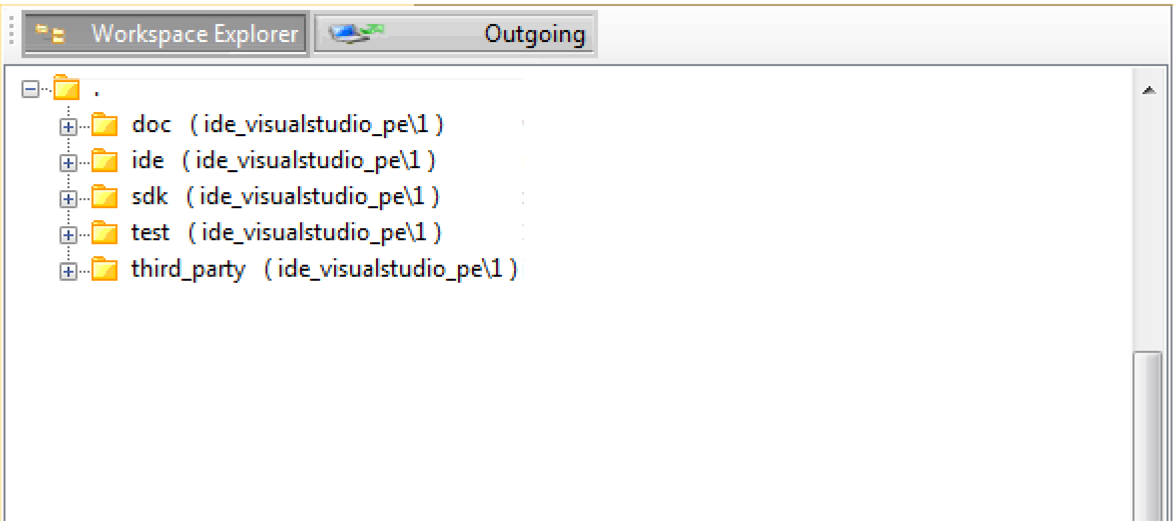
1. In the **Depot Explorer**, navigate to the stream or workspace whose files you wish to view.
2. Right click the stream or workspace, and choose **Open** from the shortcut menu.

   The **File** view appears on a new tab in the **Depot Explorer**.

AccuRev Workspace Explorer

You can use the **Workspace Explorer** option to view the folders and files in the current workspace in a tree view. To display this window, use the **View** > **AccuRev Workspace Explorer** command.

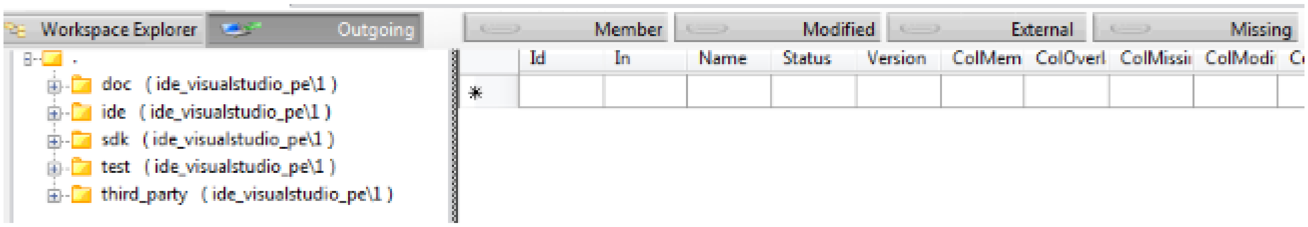
The **AccuRev Workspace Explorer** opens with the **Workspace Explorer** option selected:



.
doc (ide_visualstudio_pe\1)
ide (ide_visualstudio_pe\1)
sdk (ide_visualstudio_pe\1)
test (ide_visualstudio_pe\1)
third_party (ide_visualstudio_pe\1)

If you want to view the elements that have changed in the workspace, select the **Outgoing** option to display only those files and directories that have been added to, changed in, or removed from the current workspace. You can also keep and promote files from the **Outgoing** view.

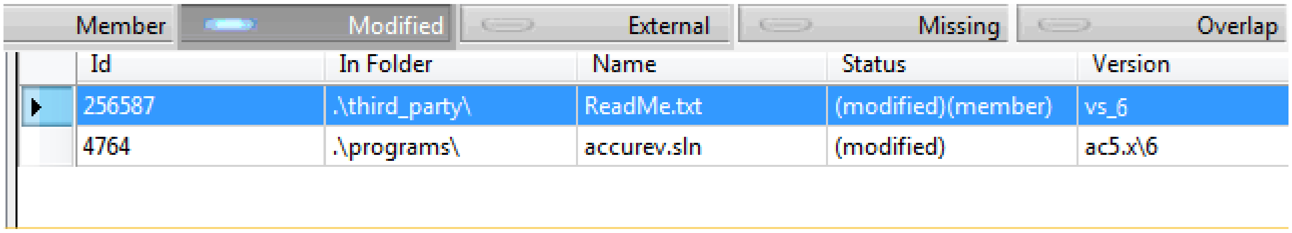
When you select the **Outgoing** option, the **AccuRev Workspace Explorer** expands to display another panel on the right that contains the **Member**, **Modified**, **External**, **Missing**, and **Overlap** buttons.



.
doc (ide_visualstudio_pe\1)
ide (ide_visualstudio_pe\1)
sdk (ide_visualstudio_pe\1)
test (ide_visualstudio_pe\1)
third_party (ide_visualstudio_pe\1)

	Id	In	Name	Status	Version	ColMem	ColOverl	ColMissi	ColModi	C
*										

When you select one of these buttons, the **AccuRev Workspace Explorer** displays the elements with that AccuRev status in the current workspace. The following example shows the files in the workspace with a **Modified** status:



	Id	In Folder	Name	Status	Version
▶	256587	.\third_party\	ReadMe.txt	(modified)(member)	vs_6
	4764	.\programs\	accurev.sln	(modified)	ac5.x\6

To keep or promote elements, right-click the elements in the **Outgoing** view and select **Keep** or **Promote**.

Query Editor

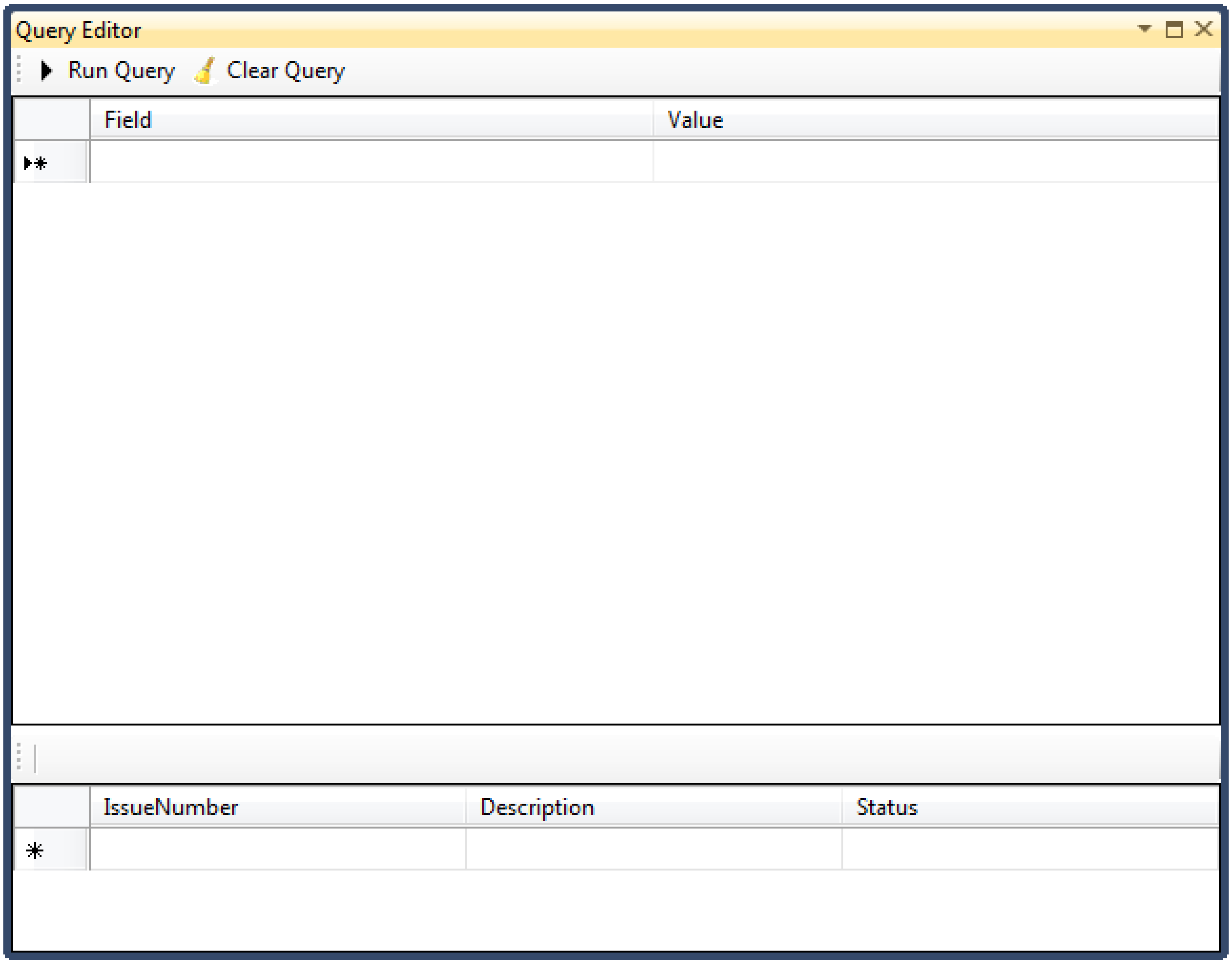
You can use the **Query Editor** to construct and run queries of AccuWork issues. You can perform the following operations using the **Query Editor**:

- *Creating Queries of AccuWork Issues*.
- *Editing AccuWork Issues*.
- *Viewing Files and Versions associated with AccuWork Issues*.
- *Shelving AccuWork Issues*.

Creating Queries of AccuWork Issues

To create a query of AccuWork issues:

1. When logged into AccuRev, select the **New Query** icon in the **AccuRev** tool bar. The **Query Editor** dialog box appears:



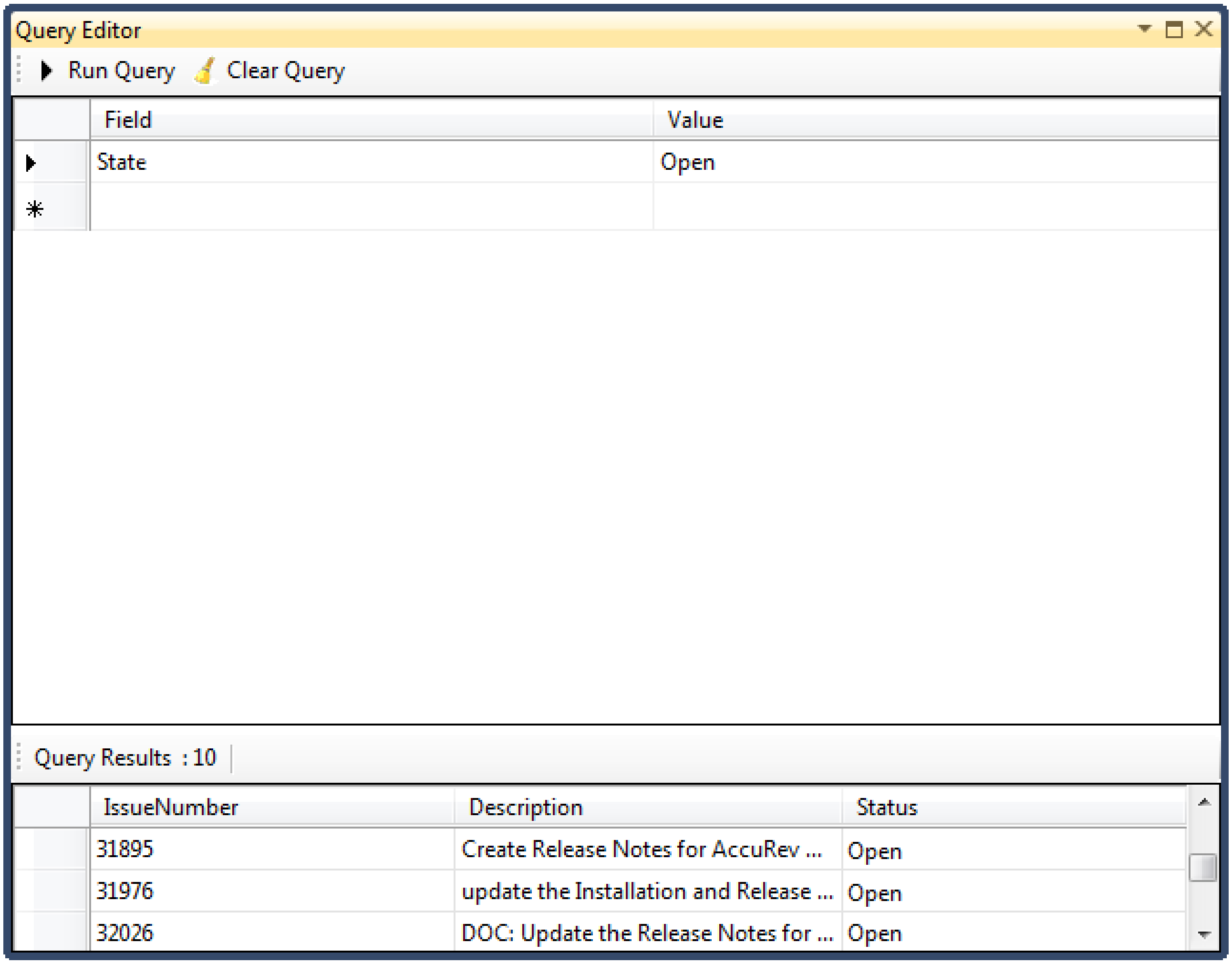
2. Select the **Field** field and select a setting from the list.
3. Select the **Value** field and select a setting from the list.

   Note the following:

   - The implied condition for the relationship between the **Field** and **Value** settings in the **Query Editor** is an equivalency =.

- You can have multiple rows of criteria. The implied join between the rows of criteria is an `AND` condition.

4. After you have specified the criteria for your query, select **Run Query**. The query results appear in the bottom portion of the **Query Editor** dialog box. In the following example, the **Field** is `State` and the **Value** is `Open`:



Field	Value
State	Open

Query Results : 10

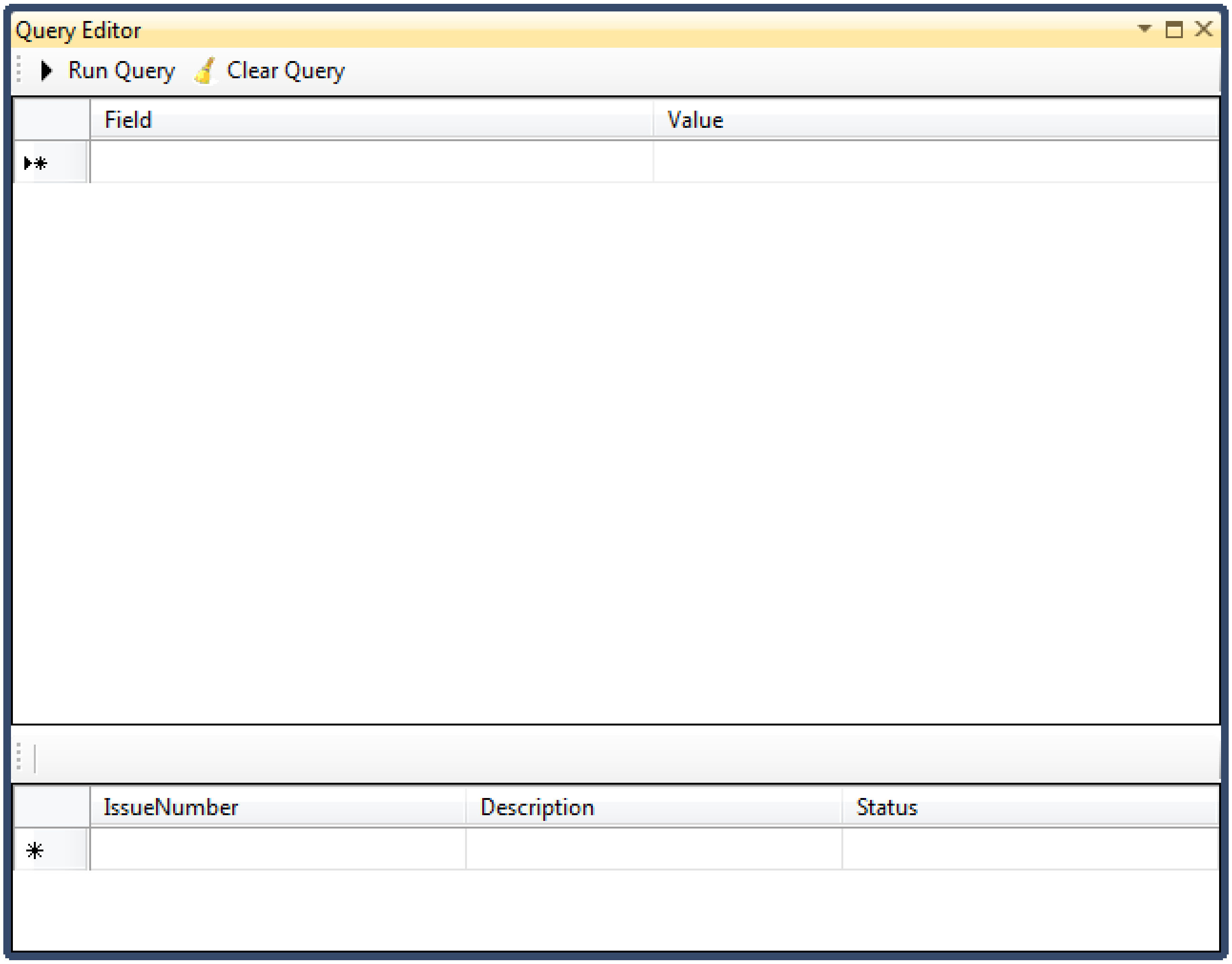
IssueNumber	Description	Status
31895	Create Release Notes for AccuRev ...	Open
31976	update the Installation and Release ...	Open
32026	DOC: Update the Release Notes for ...	Open

5. If you want to remove the current criteria at any time, select **Clear Query**.

Editing AccuWork Issues

To edit an AccuWork issue returned by a query:

1. When logged into AccuRev, select the **New Query** icon in the **AccuRev** tool bar. The **Query Editor** dialog box appears:



Field	Value

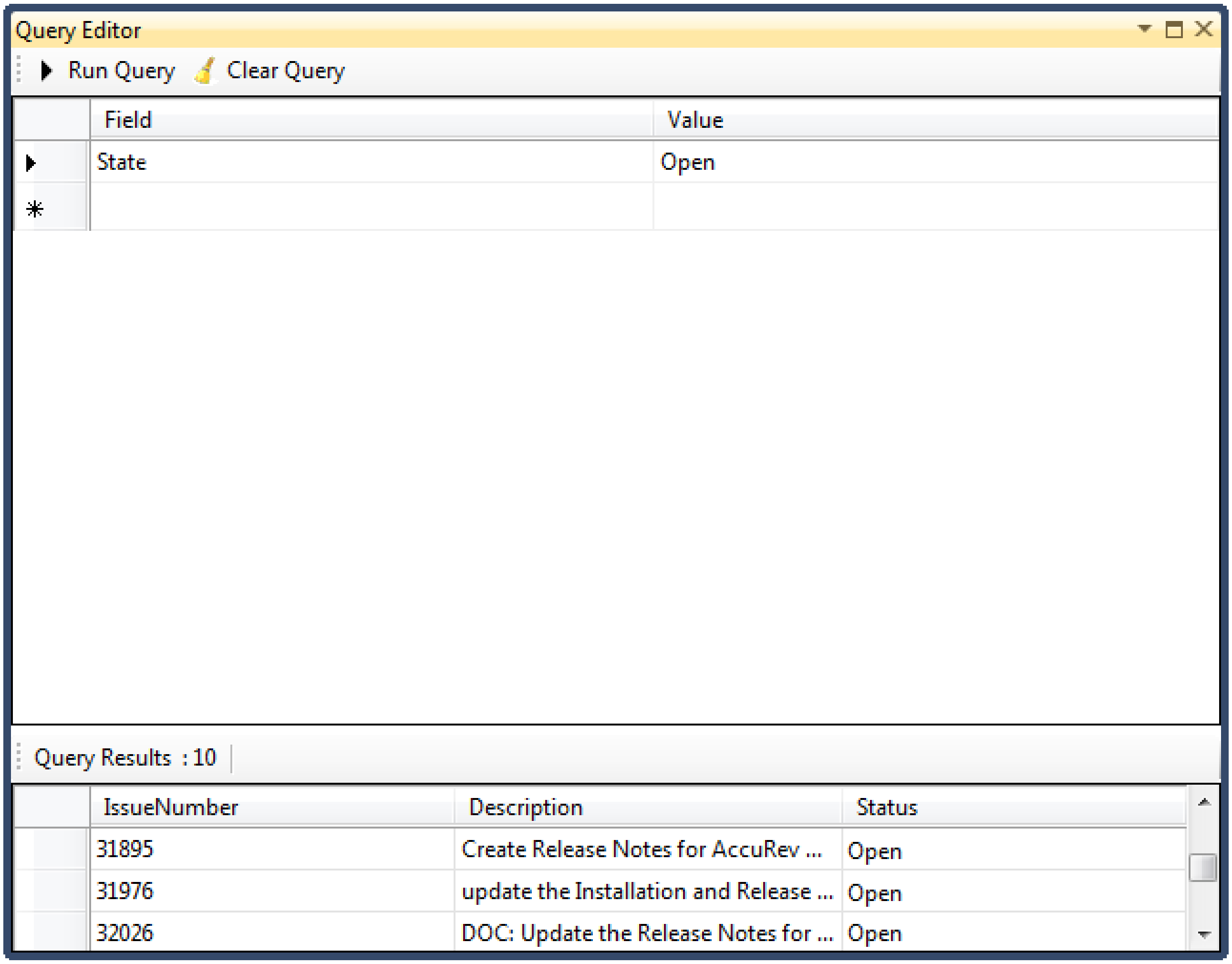
IssueNumber	Description	Status

2. Select the **Field** field and select a setting from the list.
3. Select the **Value** field and select a setting from the list.

   Note the following:

   - The implied condition for the relationship between the **Field** and **Value** settings in the **Query Editor** is an equivalency =.
   - You can have multiple rows of criteria. The implied join between the rows of criteria is an `AND` condition.

4. After you have specified the criteria for your query, select **Run Query**. The query results appear in the bottom portion of the **Query Editor** dialog box. In the following example, the **Field** is `State` and the **Value** is `Open`:



Field	Value
State	Open

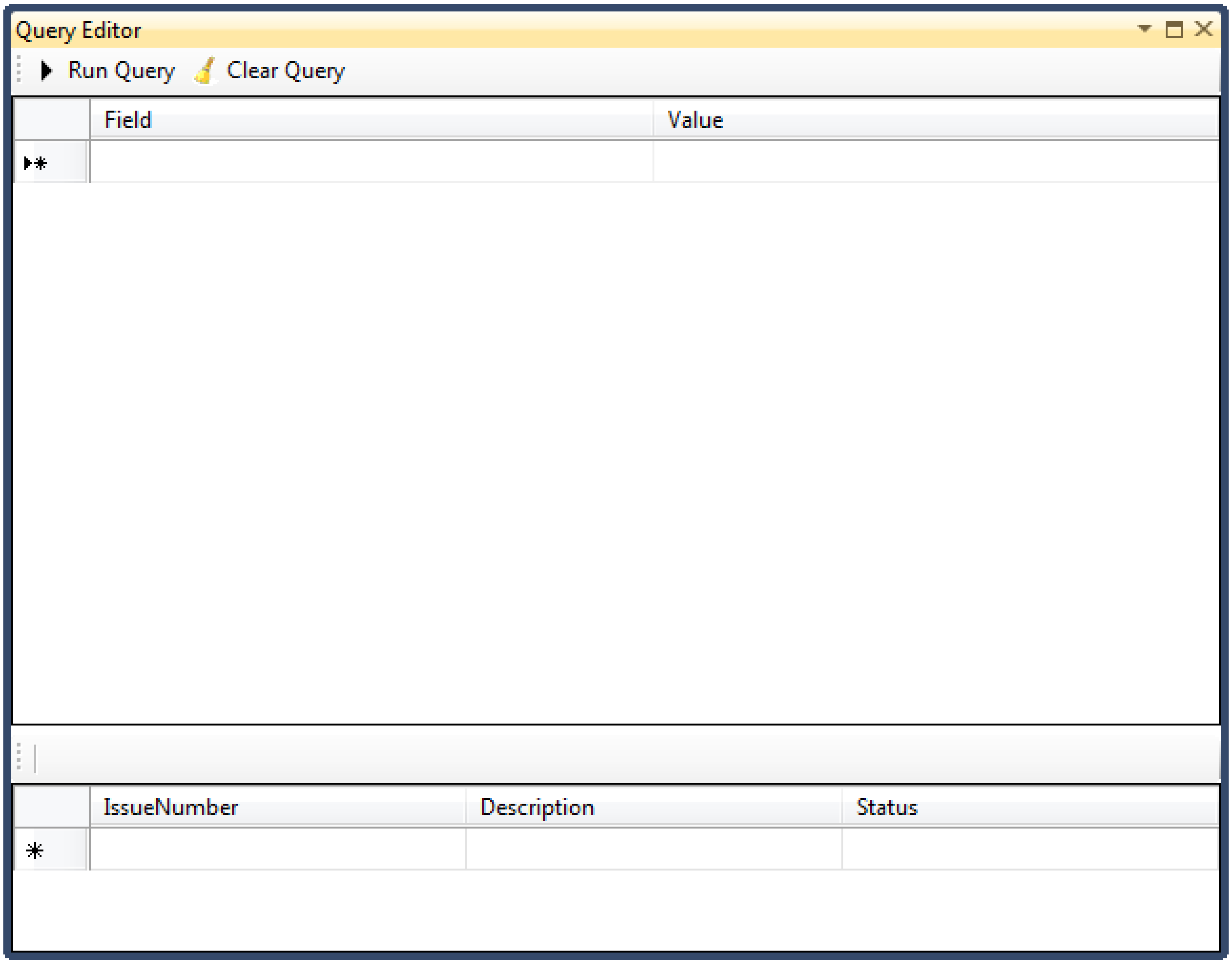
IssueNumber	Description	Status
31895	Create Release Notes for AccuRev ...	Open
31976	update the Installation and Release ...	Open
32026	DOC: Update the Release Notes for ...	Open

5. In the query results in the bottom portion of the **Query Editor** dialog box, double-click the issue you want to edit.
6. Select **Edit** to make the issue fields active. You can now make changes to the issue fields.

Viewing Files and Versions associated with AccuWork Issues

To view files and versions associated with specific AccuWork issues that are returned by AccuWork queries:

1. When logged into AccuRev, select the **New Query** icon in the **AccuRev** tool bar. The **Query Editor** dialog box appears:

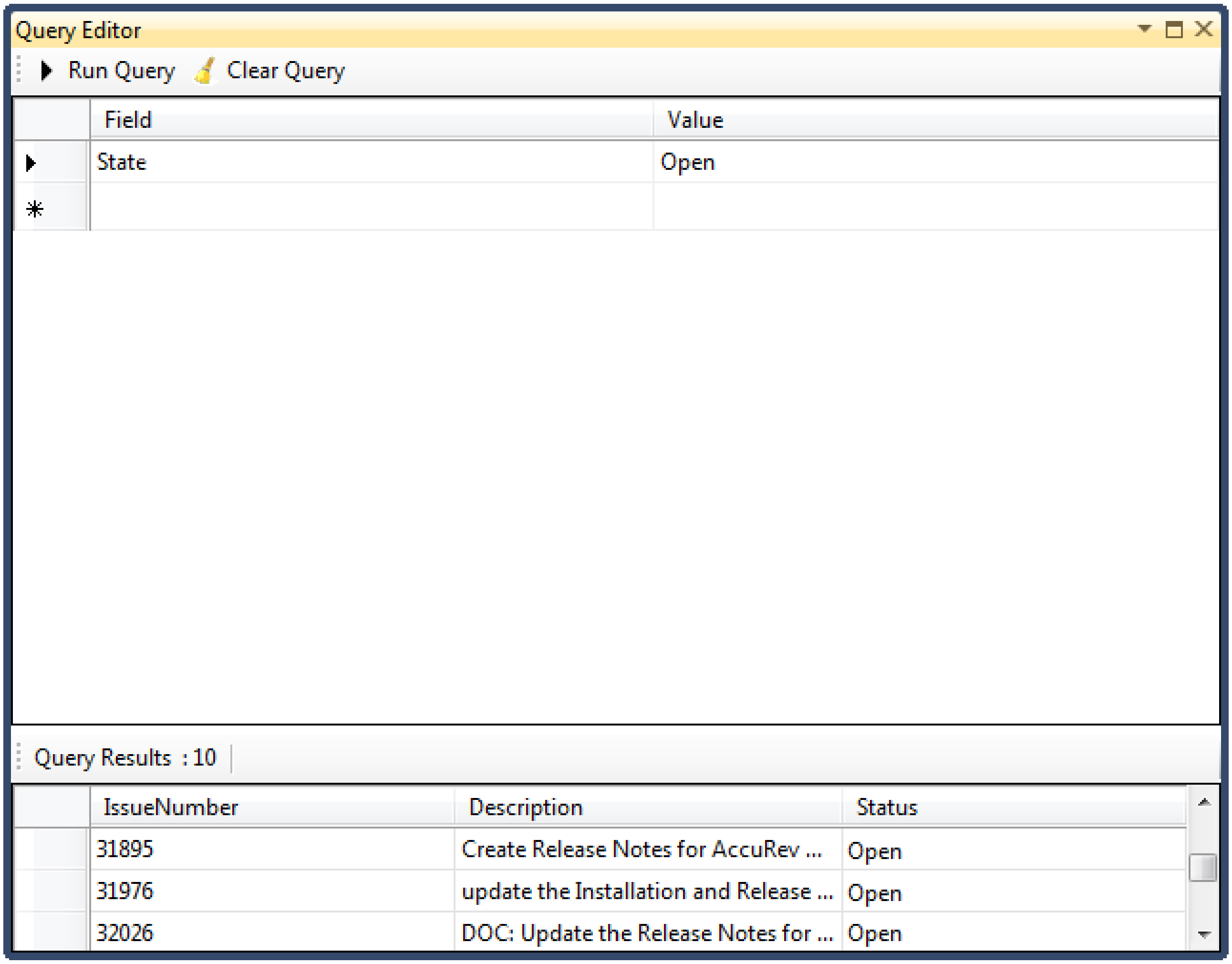


2. Select the **Field** field and select a setting from the list.
3. Select the **Value** field and select a setting from the list.

   Note the following:

   - The implied condition for the relationship between the **Field** and **Value** settings in the **Query Editor** is an equivalency =.
   - You can have multiple rows of criteria. The implied join between the rows of criteria is an `AND` condition.

4. After you have specified the criteria for your query, select **Run Query**. The query results appear in the bottom portion of the **Query Editor** dialog box. In the following example, the **Field** is `State` and the **Value** is `Open`:

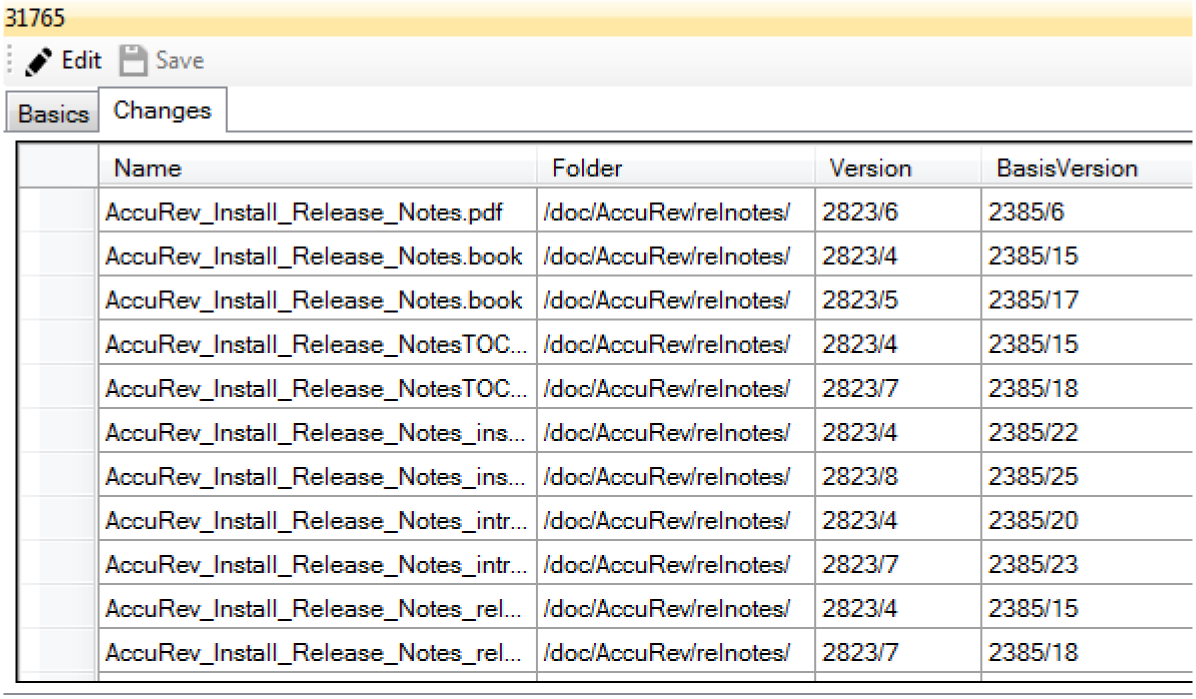


Field	Value
State	Open

Query Results : 10

IssueNumber	Description	Status
31895	Create Release Notes for AccuRev ...	Open
31976	update the Installation and Release ...	Open
32026	DOC: Update the Release Notes for ...	Open

5. In the query results in the bottom portion of the **Query Editor** dialog box, right-click the issue you want to view the associated files for and select **View File Versions**. The files associated with the currently selected issue appear on the **Changes** tab of the **<issue_number>** dialog box:



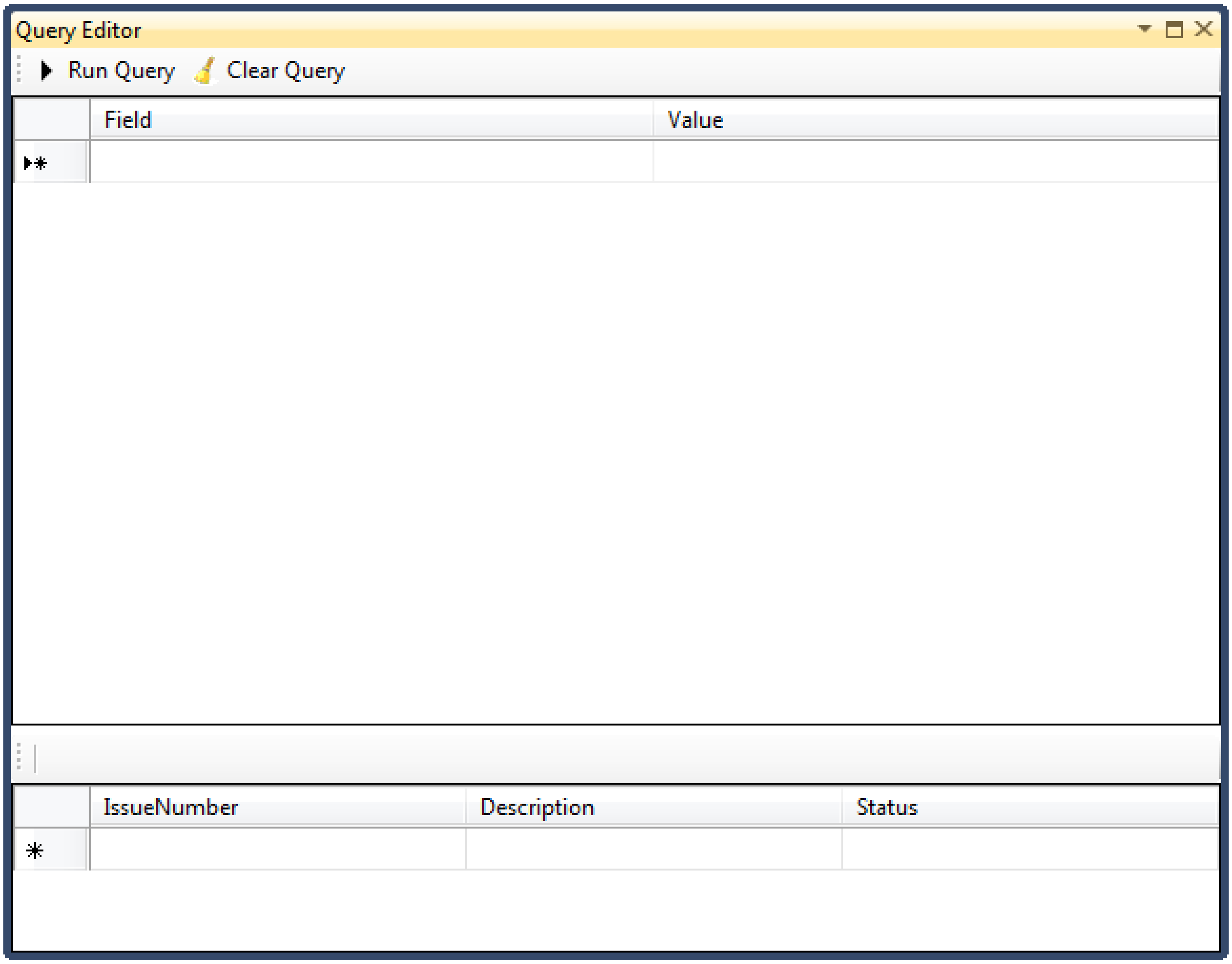
Name	Folder	Version	BasisVersion
AccuRev_Install_Release_Notes.pdf	/doc/AccuRev/relnotes/	2823/6	2385/6
AccuRev_Install_Release_Notes.book	/doc/AccuRev/relnotes/	2823/4	2385/15
AccuRev_Install_Release_Notes.book	/doc/AccuRev/relnotes/	2823/5	2385/17
AccuRev_Install_Release_NotesTOC...	/doc/AccuRev/relnotes/	2823/4	2385/15
AccuRev_Install_Release_NotesTOC...	/doc/AccuRev/relnotes/	2823/7	2385/18
AccuRev_Install_Release_Notes_ins...	/doc/AccuRev/relnotes/	2823/4	2385/22
AccuRev_Install_Release_Notes_ins...	/doc/AccuRev/relnotes/	2823/8	2385/25
AccuRev_Install_Release_Notes_intr...	/doc/AccuRev/relnotes/	2823/4	2385/20
AccuRev_Install_Release_Notes_intr...	/doc/AccuRev/relnotes/	2823/7	2385/23
AccuRev_Install_Release_Notes_rel...	/doc/AccuRev/relnotes/	2823/4	2385/15
AccuRev_Install_Release_Notes_rel...	/doc/AccuRev/relnotes/	2823/7	2385/18

Shelving AccuWork Issues

In some cases, you may want to stop working on a project for the time being and to remove elements related to that issue from your workspace. This process is known as shelving an issue. To shelve an issue means that changes you have made to files that are associated with the issue are purged from your workspace. For example, suppose that you are working on issue 1 and you change `file1.xml` and `file3.java` and you then perform a keep operation on the files and add them to the issue. If you then shelve the issue, the AccuRev Plug-In for Visual Studio PE purges those files from your workspace. The files are still associated with the issue, but they are no longer in your workspace.

To shelve an AccuWork issue returned by a query:

1. When logged into AccuRev, select the **New Query** icon in the **AccuRev** tool bar. The **Query Editor** dialog box appears:

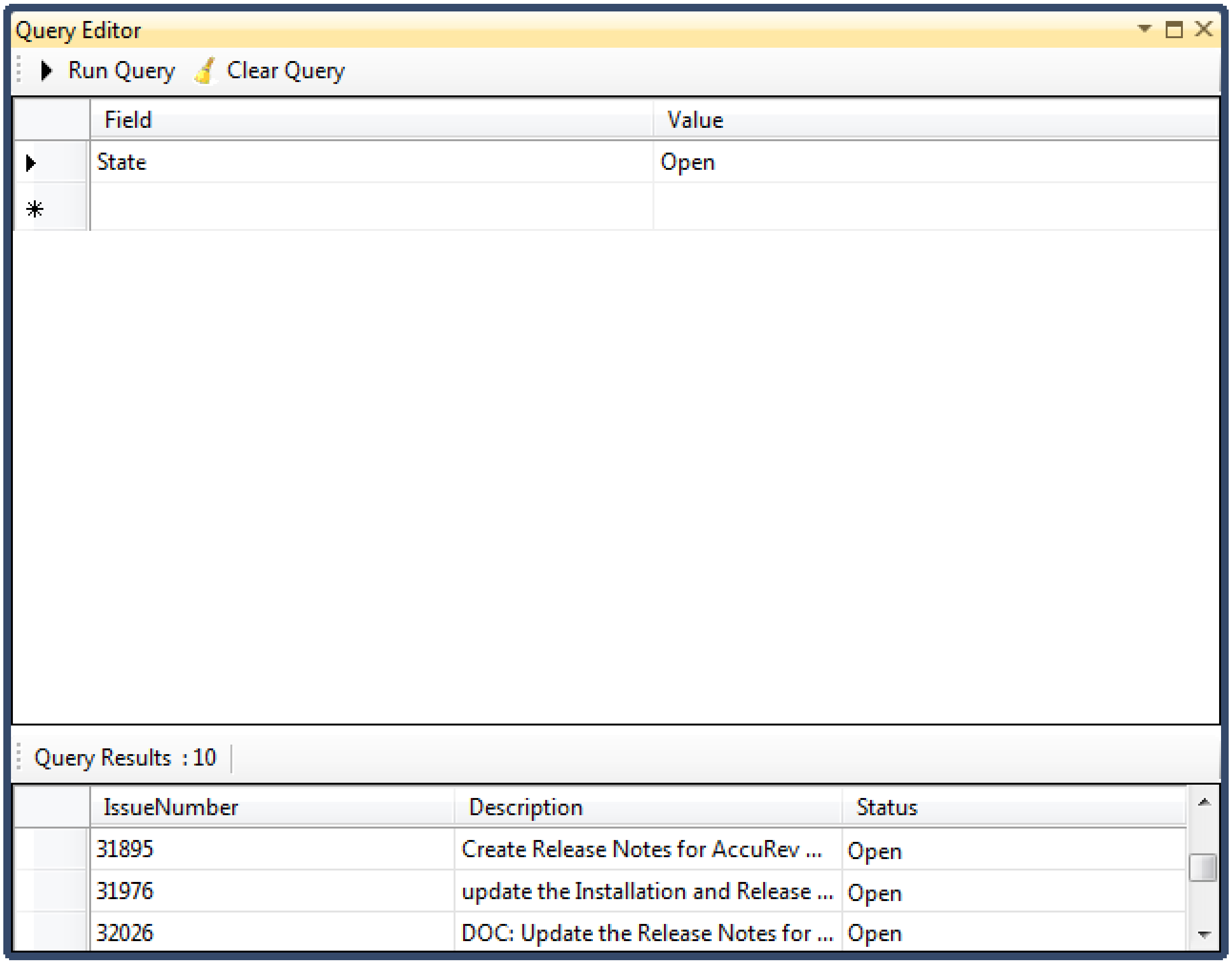


2. Select the **Field** field and select a setting from the list.
3. Select the **Value** field and select a setting from the list.

   Note the following:

   - The implied condition for the relationship between the **Field** and **Value** settings in the **Query Editor** is an equivalency =.
   - You can have multiple rows of criteria. The implied join between the rows of criteria is an `AND` condition.

4. After you have specified the criteria for your query, select **Run Query**. The query results appear in the bottom portion of the **Query Editor** dialog box. In the following example, the **Field** is `State` and the **Value** is `Open`:



Field	Value
State	Open

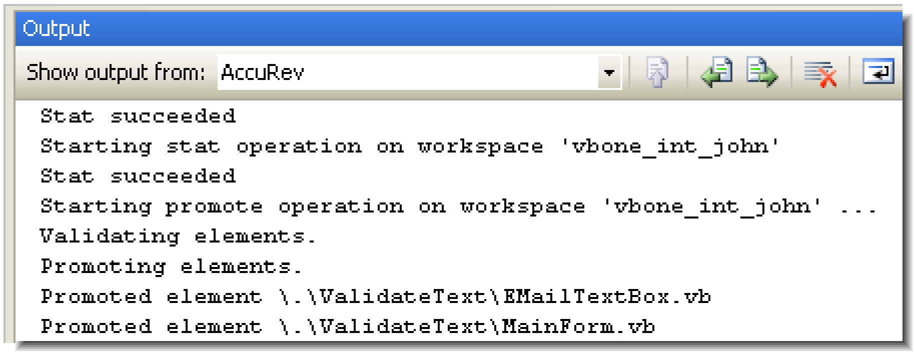
Query Results : 10

IssueNumber	Description	Status
31895	Create Release Notes for AccuRev ...	Open
31976	update the Installation and Release ...	Open
32026	DOC: Update the Release Notes for ...	Open

5. In the query results in the bottom portion of the **Query Editor** dialog box, right-click the issue you want to shelve and select **Shelve**.

AccuRev Pane in the Output Window

Messages generated by AccuRev commands are directed to the **AccuRev** pane of Visual Studio's **Output** window.

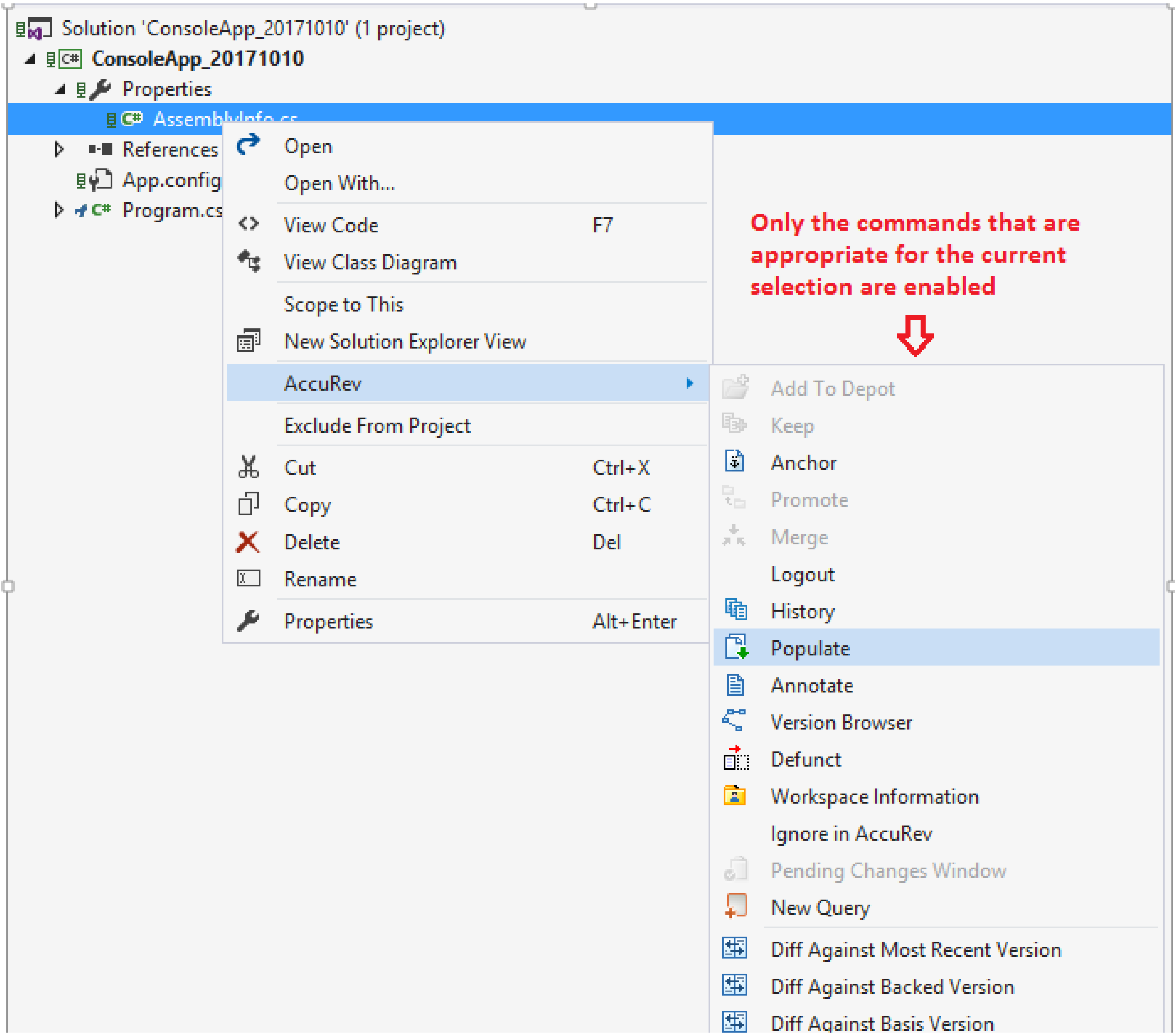


```
Stat succeeded
Starting stat operation on workspace 'vbone_int_john'
Stat succeeded
Starting promote operation on workspace 'vbone_int_john' ...
Validating elements.
Promoting elements.
Promoted element \.\ValidateText\EMailTextBox.vb
Promoted element \.\ValidateText\MainForm.vb
```

AccuRev Command Reference

To execute AccuRev commands in the **Solution Explorer**:

1. Select one or more files or directories.
2. Right-click to display the selection’s context menu, and choose a command from the **AccuRev** submenu. Alternatively, click a command button on the **AccuRev** toolbar.



Only the commands that are appropriate for the current selection are enabled

In other situations, for example, in the **AccuRev** window, AccuRev commands appear at the top level of a selection’s context menu.

Commands that operate on files and directories are enabled if they apply to one or more items in the current selection. Certain commands, such as *History* on page 44, are enabled only if a single item is selected.

Command enabling and execution takes into account the fact that a project consists of a hierarchy of items. In particular, invoking a command on an item sometimes causes the item's sub-hierarchy to be processed, also.

Examples:

- Running the *Keep* command on a directory operates on files within that directory. If no files are candidates for `Keep`, the command is disabled.
- Running the *Keep* command on a project or solution item provides an easy way to invoke the command on all the project's or solution's (modified)-status elements.
- A *Keep* command run on code file `MainForm.vb` can also include its sub-items: the form-design file `MainForm.designer.vb` and the resources file `MainForm.resx`.

The following sections describe the AccuRev commands that are available within Visual Studio:

- *Commands that Operate on Files and Directories* describes most of the commands that you'll use on a day-to-day basis. These commands are available in the **Solution Explorer** and also on the **Searches** tab and the **Synchronize** tab of the **AccuRev** window.
- A few additional commands, which operate on historical versions of elements, are described in *Commands Available on the History Tab*.
- Commands that don't operate on individual files and directories are described in *Commands Available on the AccuRev Menu*.
- Certain Visual Studio operations cause AccuRev commands to be executed automatically. This is discussed in *Automatic Invocation of AccuRev Commands*.

Commands that Operate on Files and Directories

The following commands operate on the files and directories within Visual Studio projects and solutions. Most are available in the **Solution Explorer**, and are also available in the *Searches* tab and the *Synchronize* tab of the AccuRev window.

Add to Depot

Converts the selected items into AccuRev version-controlled elements. The directory containing the files is also converted to an element, if necessary. You are prompted to enter a comment, just as with the *Keep* command. This command applies only to items not already under AccuRev version control, that is, to items whose AccuRev status is `(external)`.

Anchor

Activates the selected file in your workspace, adding it to the workspace's default group. Doing so ensures that:

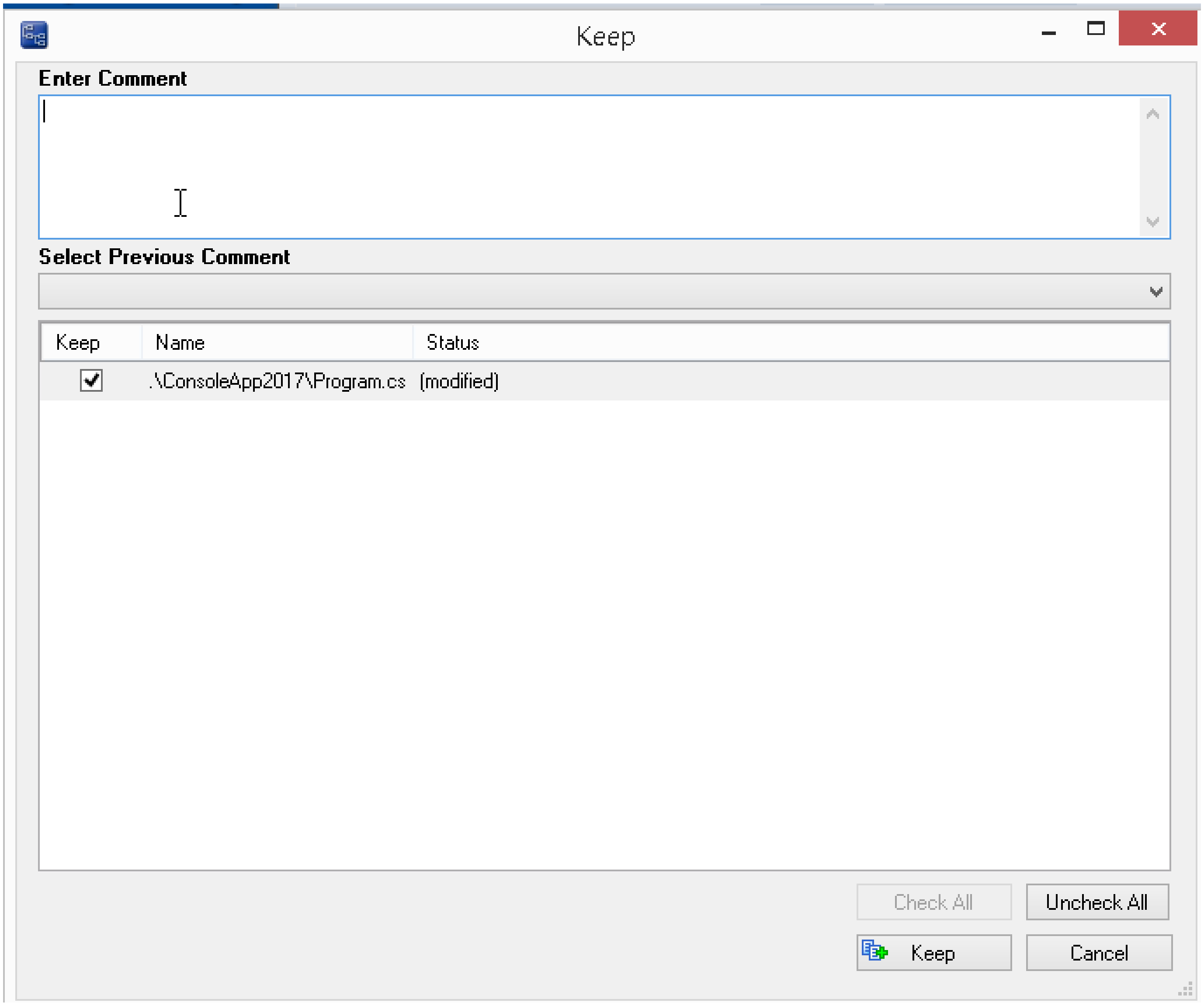
- The file is not read-only.
- The file won't be overwritten by an `Update` command.

If your workspace uses the exclusive file locking or anchor-required feature, its files are initially read-only. In that case, you can `Anchor` the file before editing it, or AccuRev Plug-In for Visual Studio PE will invoke `Anchor` automatically when you save the file.

Keep

Saves the changes you've made to one or more elements as *private* versions in the AccuRev repository. These versions are visible only in your workspace, not in the backing stream or other users' workspaces. Use the `Promote` command to send versions created with `Keep` to the backing stream, making them public.

You are prompted to enter a comment, which can span multiple lines. You can reuse a comment that you have entered previously for a `Keep`, `Add to Depot`, `Promote`, or `Defunct` command, in this Visual Studio session or in a previous one.



Keep	Name	Status
☑	.\ConsoleApp2017\Program.cs	(modified)

After selecting an old comment, you can revise it before submitting it as the comment for the current command.

Anchor

Activates the selected file in your workspace, adding it to the workspace's default group. Doing so ensures that:

- The file is not read-only.
- The file won't be overwritten by an `Update` command.

If your workspace uses the exclusive file locking or anchor-required feature, its files are initially read-only. In that case, you can `Anchor` the file before editing it, or AccuRev Plug-In for Visual Studio PE will invoke `Anchor` automatically when you save the file.

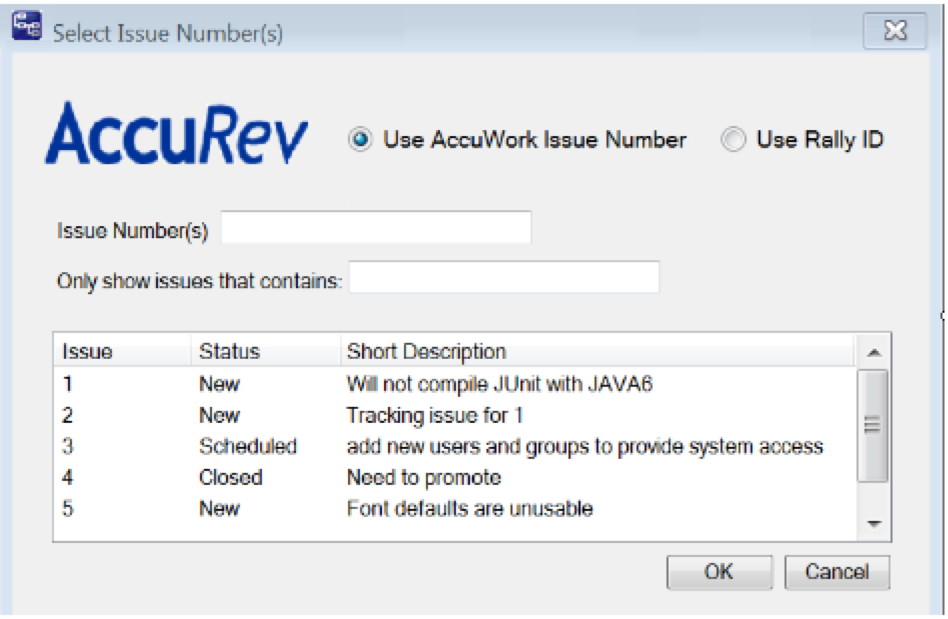
Promote

Converts the *private* versions of one or more file elements into *public* versions. It takes versions that you previously created in your workspace with `Keep`, and sends them to the backing stream shared by you and other members of your development team. You are prompted to enter a comment, just as with the *Keep* command.

If any of the selected files have been changed since their last `Keep`, and so have (`modified`) status, this command first `Keeps` the changed files, then `Promotes` all files in the selection with (`kept`) status.

Change Package Trigger

If a change package trigger is defined for the current depot's issue database, a **Select Issue Number(s)** dialog box appears, displaying a set of AccuWorkissue records. The change package trigger definition includes a query, which dynamically selects the set of issue records to display. (If you are using a third-party issue tracking system with AccuRev, this dialog box gives you the option of displaying issues using either the AccuWork issue number or the third-party system key, as shown below.)



Select Issue Number(s)

● Use AccuWork Issue Number ○ Use Rally ID

Issue Number(s)

Only show issues that contains:

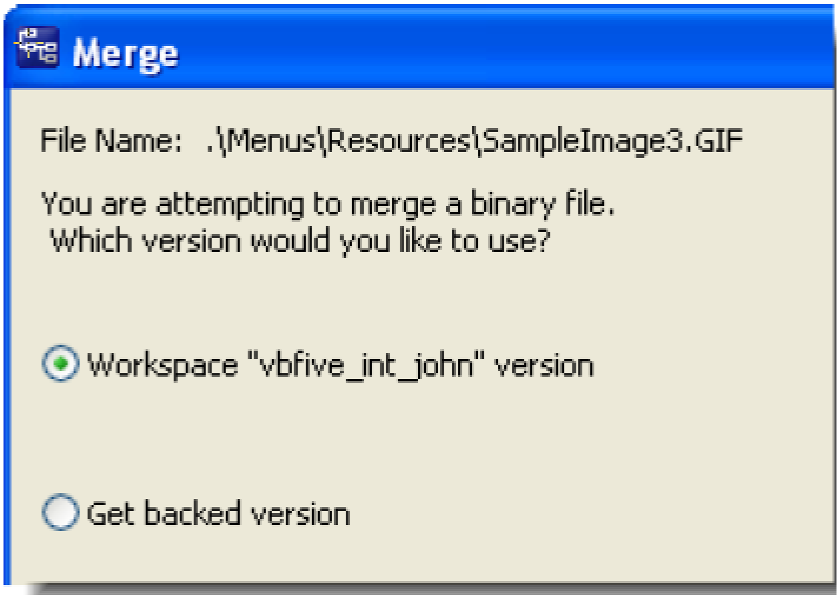
Issue	Status	Short Description
1	New	Will not compile JUnit with JAVA6
2	New	Tracking issue for 1
3	Scheduled	add new users and groups to provide system access
4	Closed	Need to promote
5	New	Font defaults are unusable

You must select one or more of these issue records by point-and-click, or by typing in the **Issue Number(s)** input field. Use SPACE characters to separate multiple issue numbers. When you click **OK**, the versions being promoted are added to the change packages of the selected issue record(s).

Merge

Combines two versions of an element, producing a new version in your workspace (recorded as a *Keep* transaction). Typically, you need to merge when you and another team member both modify the same element, and your colleague *Promotes* her changes to the common backing stream before you do. (This command is the equivalent of *Resolve* in other source code management terminologies.)

If the element type is binary, merging amounts to picking your own version or the backing-stream version. AccuRev does not support combining the contents of binary files.



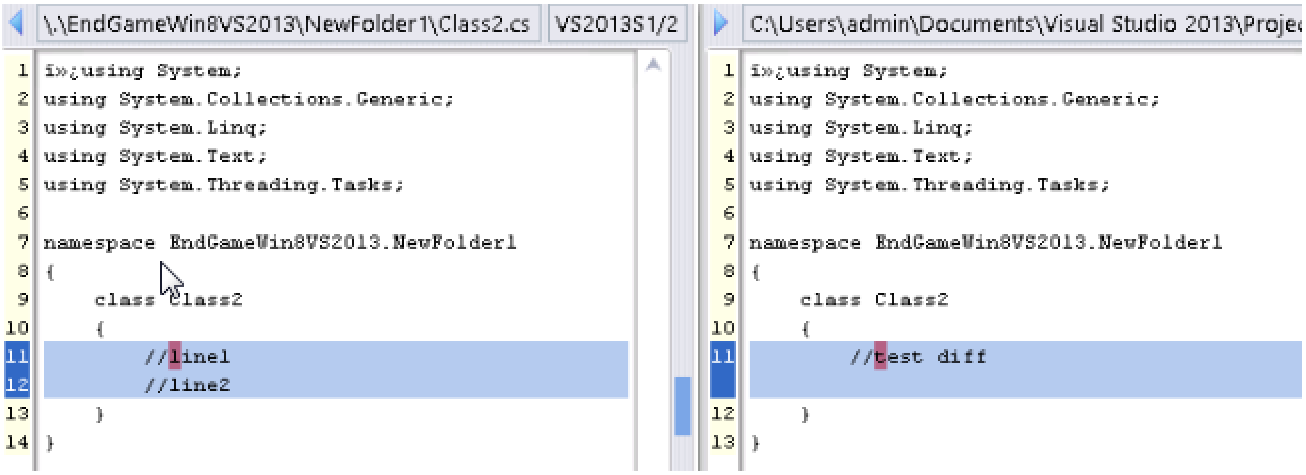
File Name: .\Menus\Resources\SampleImage3.GIF

You are attempting to merge a binary file.
Which version would you like to use?

● Workspace "vbfive_int_john" version

○ Get backed version

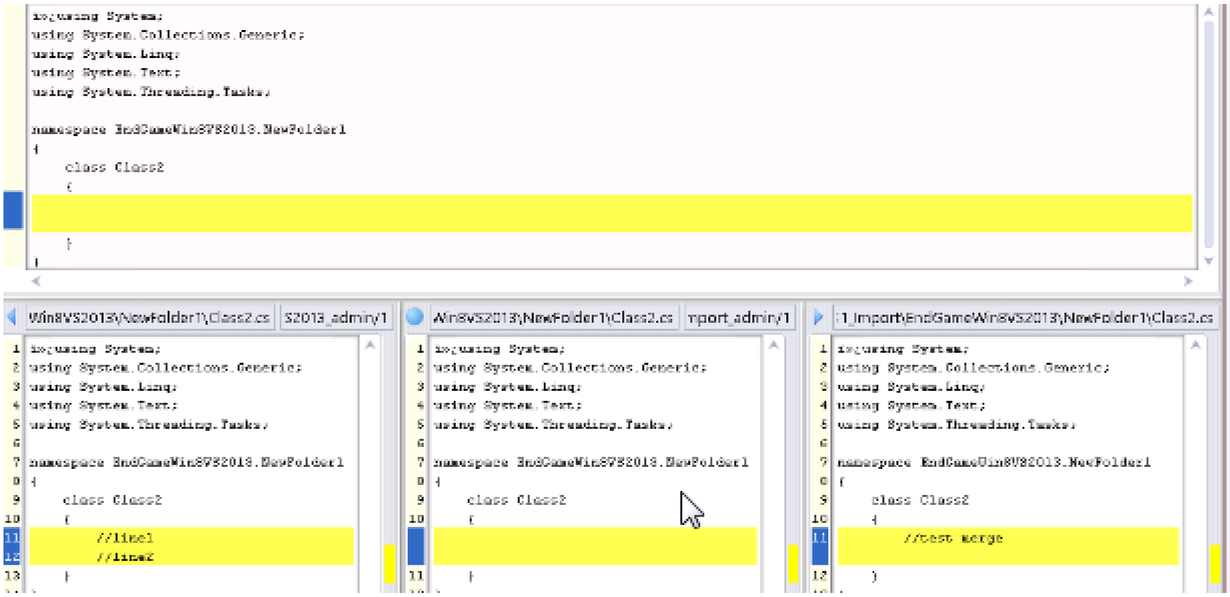
If the element type is text, the versions' contents are merged with an external program, by default, AccuRev's own Merge tool, which uses a three-way merge algorithm. You can configure another merge tool, using the **Tools** > **Options** command . See *Setting AccuRev Options*


\.\EndGameWin8VS2013\NewFolder1\Class2.cs
VS2013S1/2
C:\Users\admin\Documents\Visual Studio 2013\Projec

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {
        //line1
        //line2
    }
}
```

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {
        //test diff
    }
}
```

A merge might be required on the `namespace` level instead of, or in addition to, the content level. This occurs when the file or directory has been renamed or moved, both in your AccuRev workspace / Visual Studio project and in the backing stream version. The `Merge` command has you choose which namespace change to use. You can also choose to reject both changes, and use the element's "original" name in the common ancestor version.



```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {

    }
}
```

Win8VS2013\NewFolder1\Class2.cs
S2013_admin/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {
        //line1
        //line2
    }
```

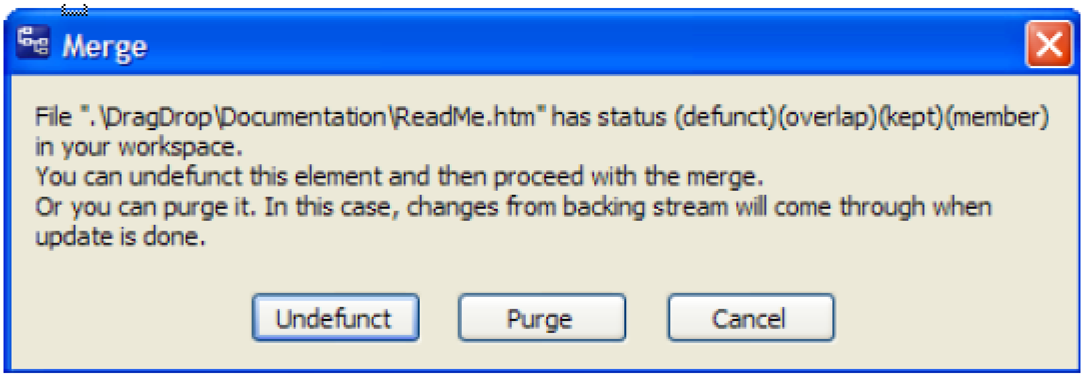
Win8VS2013\NewFolder1\Class2.cs
mport_admin/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {

    }
```

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EndGameWin8VS2013.NewFolder1
{
    class Class2
    {
        //test merge
    }
```

Note: After a namespace-level merge in which you choose `not` to retain the name in your workspace, the project file must be updated to reflect the name change. In many cases, you can Update your workspace to bring in a new version of the project file that contains the name change. If not, use Visual Studio's `Delete` command to remove the old name from the project file, then use **Add** > **Existing Item** to include the renamed item in the project.

Another special case occurs if an item has `(defunct)` status in your workspace. (You performed a Visual Studio `Delete`, which triggered an AccuRev `Defunct`.) A dialog box appears, in which you can reinstate `undefunct` the item:



File ".\DragDrop\Documentation\ReadMe.htm" has status (defunct)(overlap)(kept)(member) in your workspace.
You can undefunct this element and then proceed with the merge.
Or you can purge it. In this case, changes from backing stream will come through when update is done.

This enables you to change your mind about deleting it, and incorporate the changes in the backing-stream version instead. Choosing **Purge** performs a `Revert to Backed Version` command, which discards all your changes to the item, including the **Delete/Defunct**. If you really want the item to be removed from the project:

1. Choose **Purge** in this dialog.
2. *Update* your workspace.
3. `Delete` the item again and allow AccuRev Plug-In for Visual Studio PE to `Defunct it`.
4. Promote the `(defunct)` item to the backing stream.

Login (Logout)

AccuRev is configured to use the `accurev_login` user-authentication scheme, you must execute this command before you can perform any AccuRev operations within Visual Studio.

History

Displays the AccuRev transactions that created versions of the selected element in the **History** tab of the **AccuRev** window. See *History Tab*.

Locate in Solution Explorer

Highlights the selected item in the **AccuRev Solution Explorer** pane (available in the AccuRev window only), expanding the item's project hierarchy, if necessary to make it visible.

Open File

Opens the selected file in Microsoft Visual Studio's tabbed document window. (Available in the AccuRev window only).

Populate

Restores a file with `(missing)` status to your project. To locate such files, use the **Missing element** search, see the documentation for the *Searches Tab*.

The `Populate` command copies the *version* currently in the *workspace stream* (in the AccuRev repository) to the workspace. For more information, see `What's the Difference between Populate and Update?` in *AccuRev Technical Notes*.

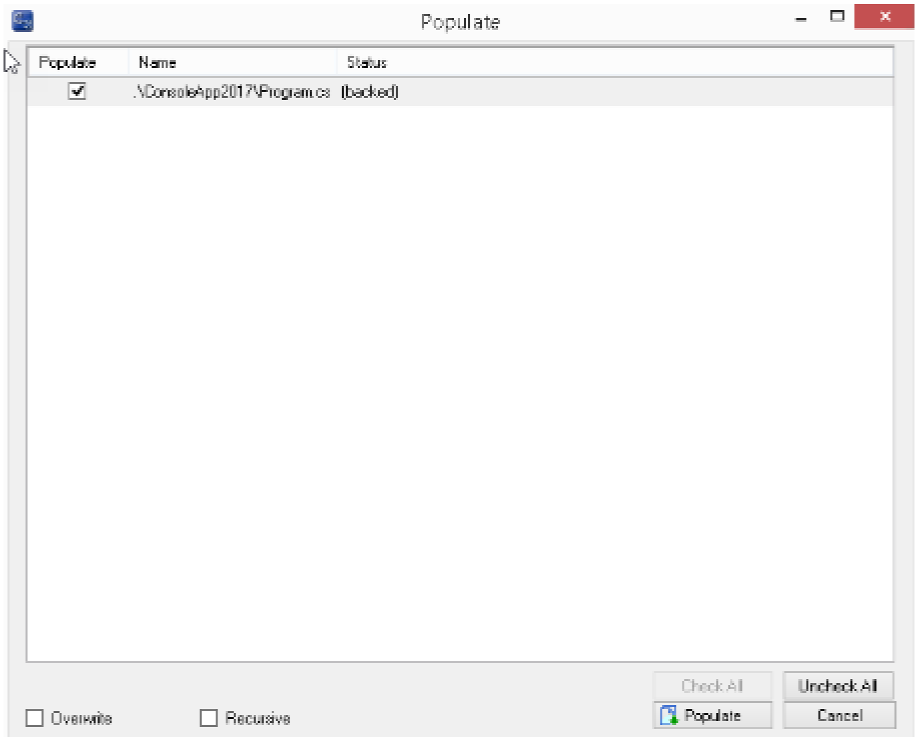
Invoking the `Populate` command displays a dialog listing the files and directories to be populated. Use the checkboxes in the **Populate** column to choose which of the listed items will be populated.

There are two other controls at the bottom of the dialog:

Recursive For each item in the list that is a directory, perform a Populate command on that element and on all elements below it.

Overwrite For each item in the list that is a file, replace the file (if any) currently in your workspace.

If this checkbox is not selected, no files will be overwritten.



Populate

Populate	Name	Status
☑	.\ConsoleApp2017\Program.cs	(backed)

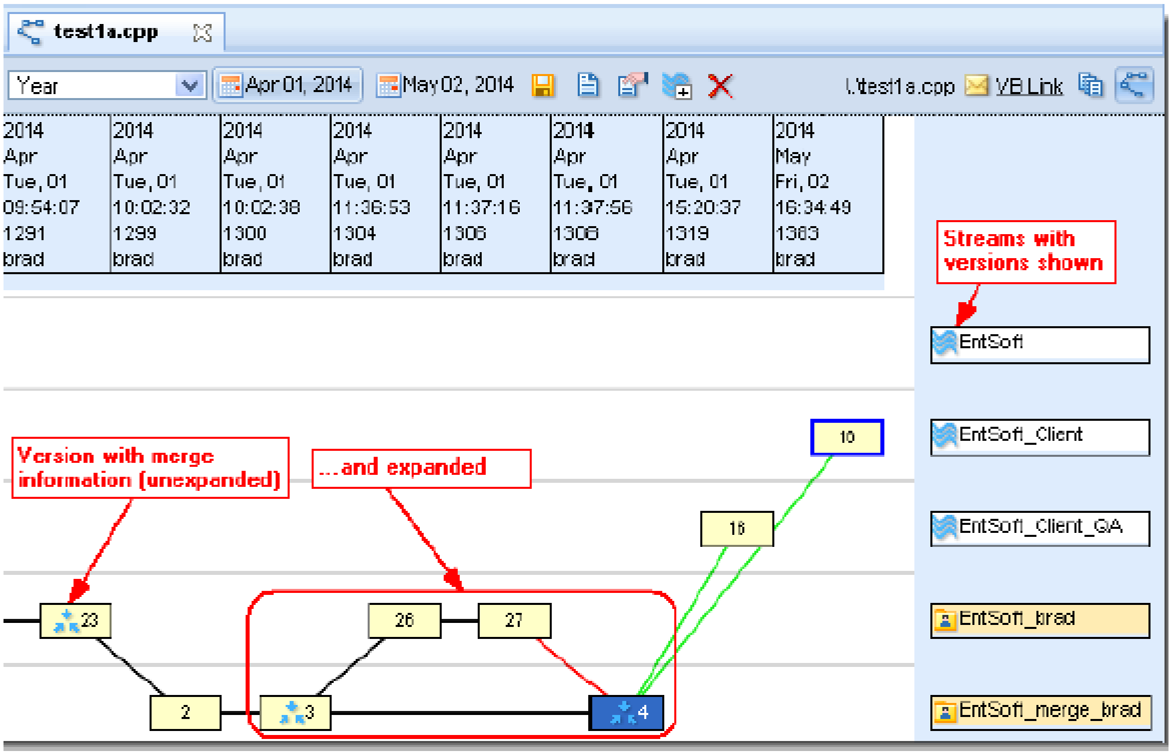
□ Overwrite □ Recursive

Annotate

Runs the AccuRev Web UI to display a read-only copy of the selected file, prefixing each line with one or more of the following: the user who created the line, the transaction in which the line was added or most recently modified, the timestamp of that transaction, the version-ID of the version created in that transaction. Also noted are lines that have been modified in the workspace but not yet kept or promoted.

Version Browser

Shows the ancestry for each element you select in the **Version Browser**, a graphical view that displays the current version of the selected element, the real ancestors of that version, and the virtual versions that have one of those ancestors as a parent. The **Show** field in the **Version Browser** toolbar specifies the number of transactions displayed in the **Version Browser**. The **Version Browser** time line shows when each transaction took place, as well as the number and user associated with the transaction.



2014	2014	2014	2014	2014	2014	2014	2014
Apr	Apr	Apr	Apr	Apr	Apr	Apr	May
Tue, 01	Tue, 01	Tue, 01	Tue, 01	Tue, 01	Tue, 01	Tue, 01	Fri, 02
09:54:07	10:02:32	10:02:38	11:36:53	11:37:16	11:37:56	15:20:37	16:34:49
1291	1299	1300	1304	1305	1306	1319	1383
brad	brad	brad	brad	brad	brad	brad	brad

Streams with versions shown

Version with merge information (unexpanded)

...and expanded

Note: The time line proceeds left-to-right, therefore the most recent versions will be visible on the right side of the **Version Browser**.

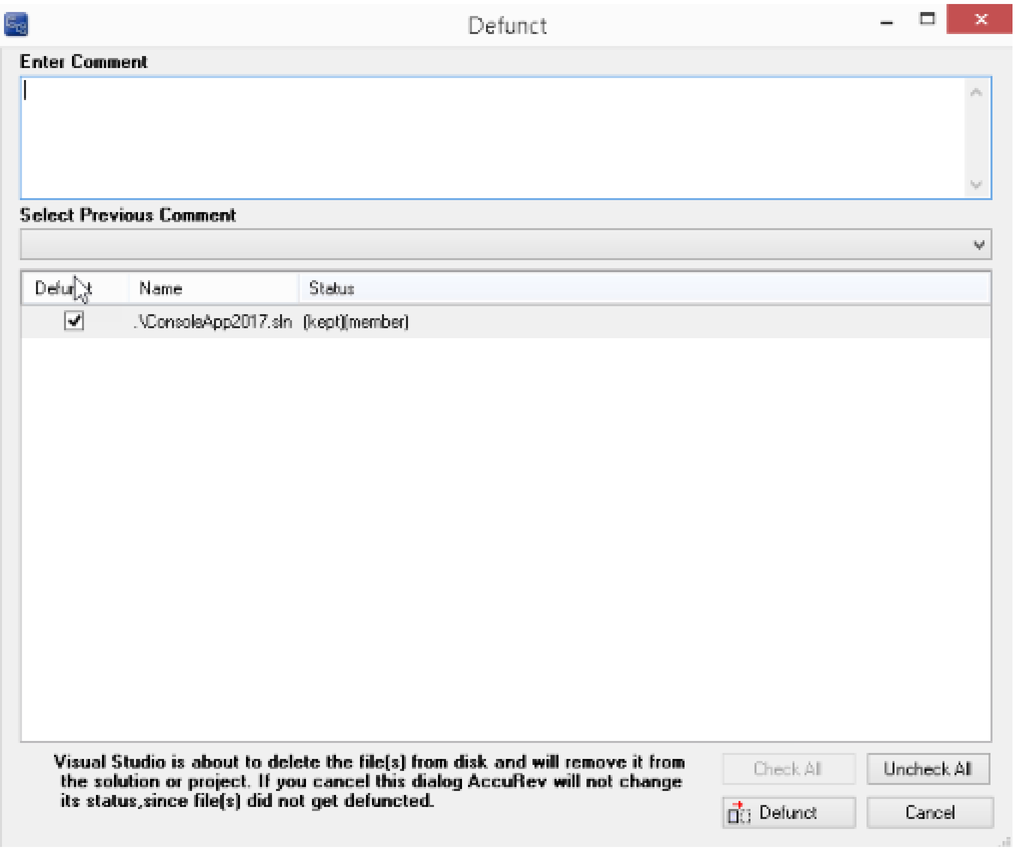
Defunct

Deletes the selected file from disk, and marks it with (defunct) status in the AccuRev workspace. You are prompted to enter a comment, just as with the *Keep* command.

A defunct file is no longer on your disk, but appears in the results of an AccuRev search for `Defunct`, `Pending`, or `Default Group` elements. It disappears entirely from the AccuRev workspace when you *Promote* the element to the backing stream.

Note: You can also defunct a directory element, but this can have far-reaching consequences. Before doing so, consult the description of the defunct command in the *AccuRev CLI User's Guide*.

Visual Studio's Delete command removes a file from your project (and local disk storage), then automatically offers to perform a `Defunct`, as well:



Defunct	Name	Status
☑	.\ConsoleApp2017.sln	(kept)(member)

Visual Studio is about to delete the file(s) from disk and will remove it from the solution or project. If you cancel this dialog AccuRev will not change its status,since file(s) did not get defuncted.

Be sure to proceed with the `Defunct`, to keep the Visual Studio project in sync with the AccuRev repository. If you don't, the element will have `(missing)` status in your workspace.

Workspace Information

Displays information regarding your current AccuRev work environment. See *Workspace Information*, for more information.

Ignore in AccuRev

Excludes external files and directories from various operations to enhance performance and reduce visual clutter. For example, if you never put program-generated executables under AccuRev control, you can tell AccuRev to ignore such files.

When you select this option, a `.acignore` file appears in the same directory as the file; if you selected a directory, the`.acignore` file appears in the parent directory.

Pending Changes

Lists modified (files which have been updated) and external files (new files). To see the list of modified and external files under a specific folder, you must select that particular folder. The **Pending Changes** window is similar to the **Keep** window, you can select one (or more) file(s) from the list and perform the Keep operation on it (them).

New Query

Opens the **Query Editor** which allows you to create and run queries of AccuWork issues. You can use the **Query Editor** to view files and versions associated with specific issues, perform **Diff Against Basis** operations on them, edit issues, and shelve issues. See *New Query*.

Revert to Most Recent Version

Similar to **Revert to Backed Version**, but rolls back a file only as far as the private version you recently created with **Keep**. This command is useful when you edit a file, save it with**File** > **Save**, then decide to discard the changes.

Revert to Backed Version

Discards the changes you've made to a file. It restores the version that was in the backing stream at the time of your most recent *Update* command. But if you promoted one or more versions of the element to the backing stream since your most recent `Update`, it restores the most recently promoted version.

Diff Against Most Recent Version

Compares your file with the private version you recently created with *Keep*. It answers the question `What have I changed since my last Keep?` (Text files only. Single-file selection only).

Diff Against Backed Version

Compares your file with the version currently in the workspace's backing *stream*. (Text files only. Single-file selection only).

Diff Against Basis Version

Compares your file with the more recent of these ancestor versions:

- A version that was originally created in another *workspace*, and brought into your *workspace* with an *Update*.
- A version that you created and *Promoted* to the backing *stream*.

It answers the question `What changes have I made recently to this file?`
(Text files only. Single-file selection only).

Diff Against File on Disk

Compares your file with another file you select, using a standard **Open** dialog box. (Text files only; single-file selection only).

Commands Available on the History Tab

These commands are available only through context menus on the **AccuRev** window's **History** tab.

File in the Workspace

Compares the version created by the selected transaction with the file that is currently in your project.

Other Version

Compares the versions created by the two selected transactions.

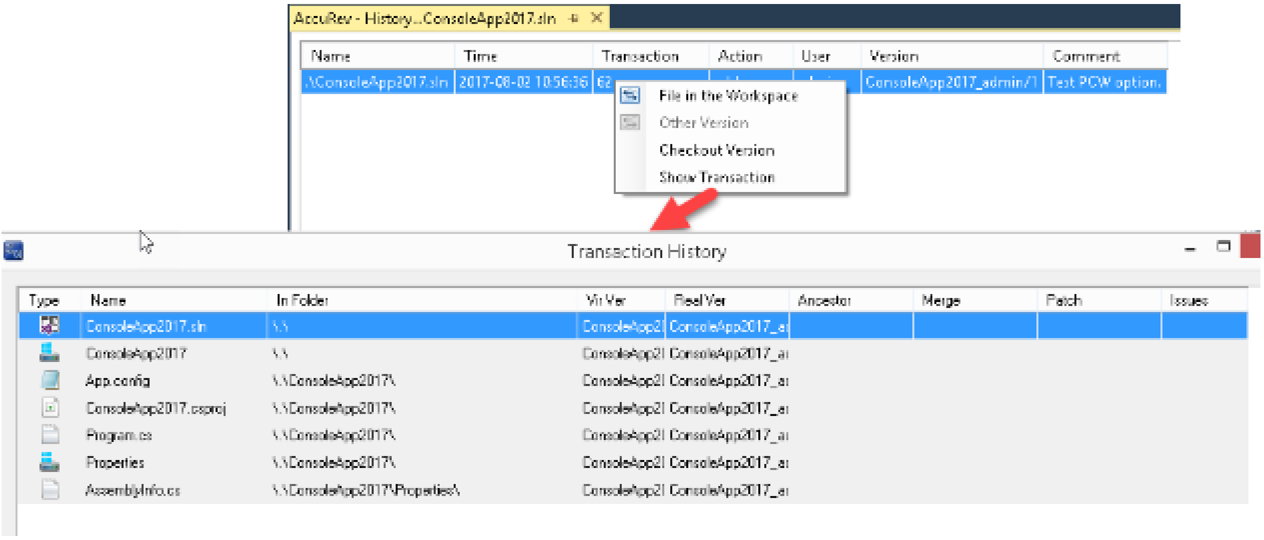
Checkout Version

Performs an AccuRev checkout (co) command, which replaces the version in your workspace with the selected version. This is similar to the *Anchor* command, but activates a historical version rather than the workspace's current version.

The element must currently be inactive in your workspace, that is, it must have (backed) status.

Show Transaction

Displays a **Transaction History** window, showing the complete contents of the selected transaction. Some kinds of transactions can create a new version of multiple elements.

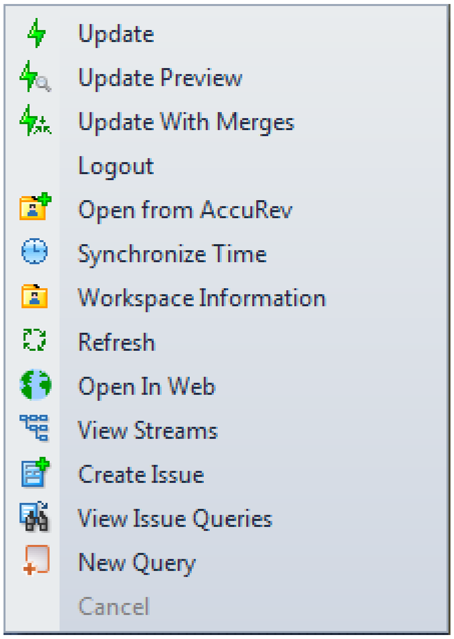


Name	Time	Transaction	Action	User	Version	Comment
.\ConsoleApp2017.sln	2017-08-02 10:56:36	62			ConsoleApp2017_admin/1	Test PCW option.

Type	Name	In Folder	Vir Ver	Real Ver	Ancestor	Merge	Patch	Issues
	ConsoleApp2017.sln	\.\	ConsoleApp2	ConsoleApp2017_a				
	ConsoleApp2017	\.\	ConsoleApp2	ConsoleApp2017_a				
	App.config	\.\ConsoleApp2017\	ConsoleApp2	ConsoleApp2017_a				
	ConsoleApp2017.csproj	\.\ConsoleApp2017\	ConsoleApp2	ConsoleApp2017_a				
	Program.cs	\.\ConsoleApp2017\	ConsoleApp2	ConsoleApp2017_a				
	Properties	\.\ConsoleApp2017\	ConsoleApp2	ConsoleApp2017_a				
	AssemblyInfo.cs	\.\ConsoleApp2017\Properties\	ConsoleApp2	ConsoleApp2017_a				

Commands Available on the AccuRev Menu

These commands are available under **AccuRev** on Visual Studio's main menu. They are also available under **File** > **AccuRev**. The **AccuRev** menu appears only if a solution is active in the **Solution Explorer**. The solution need not have any projects under AccuRev source control.



Update

Performs an AccuRev `Update` command on all workspaces that contain projects in the current solution. An update copies versions into your workspace from your workspace's backing stream. This has the effect of incorporating other people's changes, which they have promoted to the backing stream, into your workspace.

Update Preview

Lists the changes that would be made by an actual workspace update in the **Output** window's **AccuRev** pane.

Login (Logout)

If AccuRev is configured to use the `accurev_login user-authentication` scheme, you must execute this command before you can perform any AccuRev operations within Visual Studio.

Do not use this command if AccuRev is configured to use the traditional user-authentication scheme. See *Establishing Your Identity* for more information.

Open from AccuRev

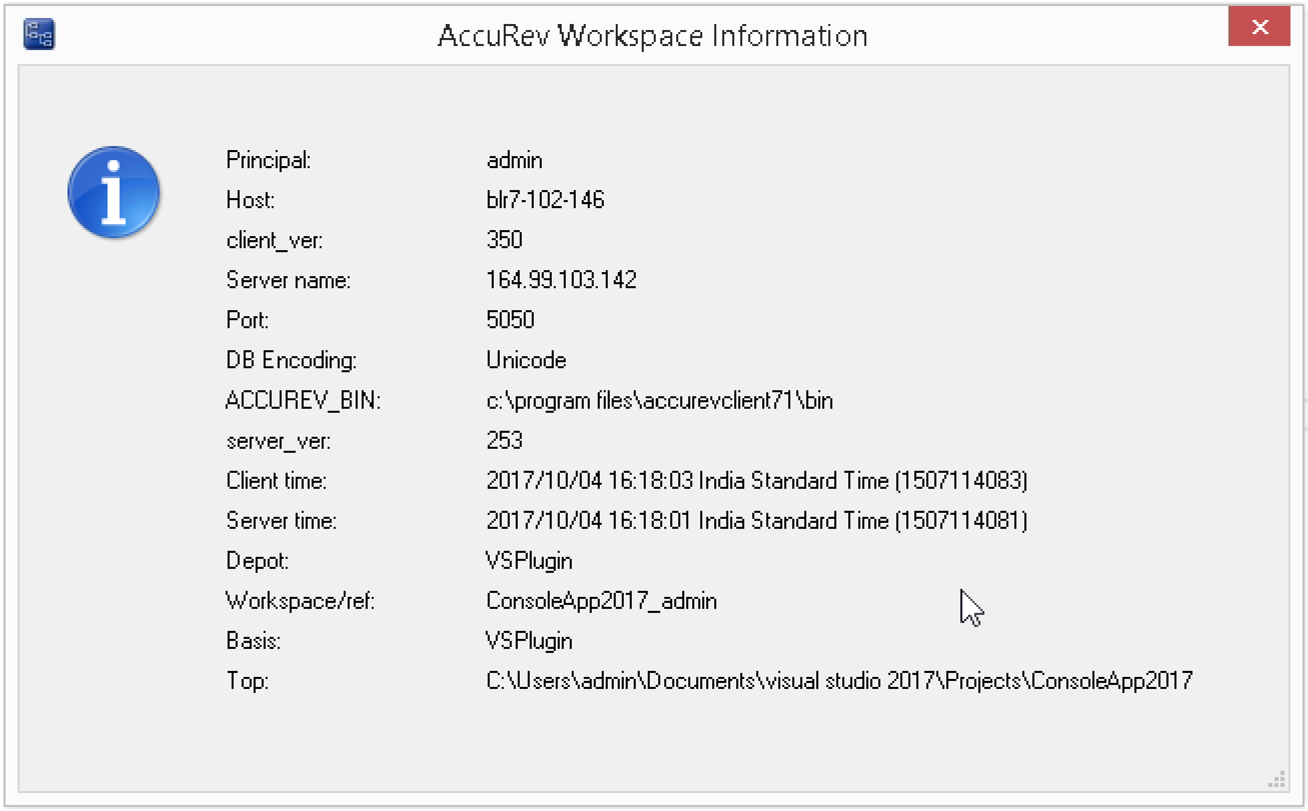
Runs a wizard that helps you create a new AccuRev workspace or open a solution that is in an AccuRev workspace. See *Creating an AccuRev Workspace*.

Synchronize Time

Adjusts the system clock on your machine to match the clock on the AccuRev Server machine. This requires `administrator-level` permissions at the operating system level.

Workspace Information

Displays information regarding your current AccuRev work environment.



Principal:	admin
Host:	blr7-102-146
client_ver:	350
Server name:	164.99.103.142
Port:	5050
DB Encoding:	Unicode
ACCUREV_BIN:	c:\program files\accurevclient71\bin
server_ver:	253
Client time:	2017/10/04 16:18:03 India Standard Time (1507114083)
Server time:	2017/10/04 16:18:01 India Standard Time (1507114081)
Depot:	VSPlugin
Workspace/ref:	ConsoleApp2017_admin
Basis:	VSPlugin
Top:	C:\Users\admin\Documents\visual studio 2017\Projects\ConsoleApp2017

Refresh

Contacts the AccuRev Server and updates the status of all items in the solution. Use this command if a process other than Visual Studio has modified development files.

Open in Web

Opens the AccuRev Web UI in a new tab in Visual Studio.

View Streams

Opens the AccuRev Web UI to the **StreamBrowser**.

Create Issue

Opens the **Issue Edit** form which allows you to create a new issue in AccuWork.

View Issue Queries

Opens the **Query Browser**, which allows you to create, manage, and run AccuWork queries.

New Query

Opens the **Query Editor** which allows you to create and run queries of AccuWork issues. You can use the **Query Editor** to view files and versions associated with specific issues, perform `Diff Against Basis` operations on them, edit issues, and shelve issues.

Automatic Running of AccuRev Commands

When you perform a Visual Studio operation that changes a project's "namespace", an AccuRev command is executed automatically, in order to keep the Visual Studio project namespace in sync with the AccuRev repository namespace.

Rename

If you perform a `Rename` command on an element, a corresponding AccuRev `Rename` (or `move`) command is invoked automatically. The change is logged to the **AccuRev** pane of the **Output** window:

```
Starting rename operation on '.\ValidateText\Documentation\ReadMe.htm' ...
Moving \.\ValidateText\Documentation\ReadMe.htm to
\.\ValidateText\Documentation\ReadMeFirst.htm
Operation completed successfully
```

Delete

If you perform a `Delete` command on an element, a corresponding AccuRev *Defunct* command is invoked automatically. Be sure to proceed, don't choose to cancel in the **Defunct** dialog box. If you cancel, the element's status becomes `(missing)`.

Moving an Element

You can change the location of an element by dragging it from one directory to another, perhaps to a location in another project, while holding down the `SHIFT` key. You can also move an element with the `Cut` and `Paste` commands. AccuRev Plug-In for Visual Studio PE handles such operations as follows:

- If the destination is within the same AccuRev workspace, though not necessarily within the same Visual Studio project, an AccuRev `Rename` (or `move`) command is invoked automatically. (Note that in a C++ project, the copy operation does not create a file system object, merely a reference in the project file.)
- If the destination is not within the same AccuRev workspace, the AccuRev `Rename` command fails.

Note: A plain drag-and-drop, without holding down the `SHIFT` key, implements a copy operation, as does a `Copy-Paste` sequence. No AccuRev command is invoked in this situation. If you copy the item to a location within an AccuRev workspace, the new copy becomes an `(external)` item.

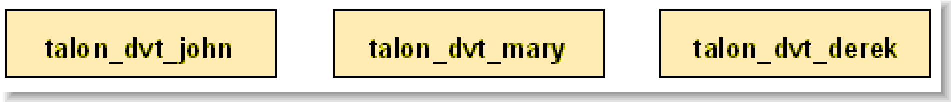
AccuRev Usage Model

This section 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.

AccuRev's flexibility makes it easy to use for a variety of development scenarios. But like every software system, it 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:



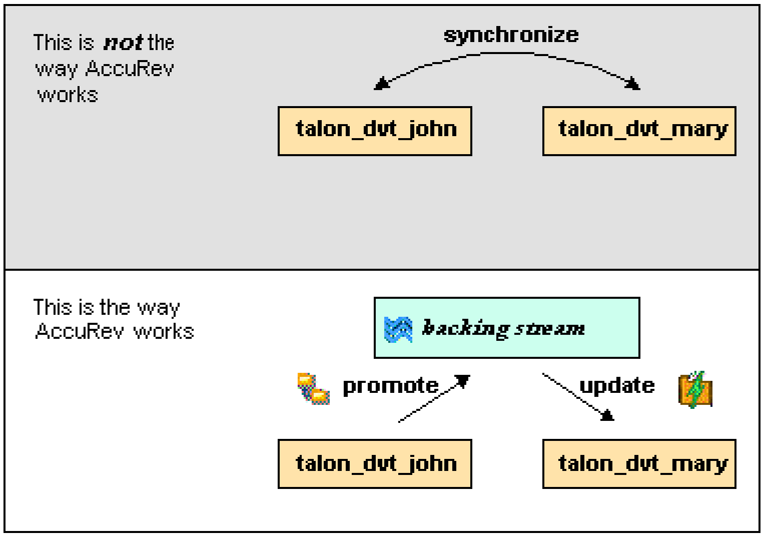
This set of users' workspaces uses the convention of having like names, suffixed with the individual usernames. AccuRev GUI enforces this username-suffix convention. In this example, talon_dvt might mean "development work on the Talon product"; john, mary, and derek would be the users' login names.

From AccuRev GUI'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 GUI organizes the work environment. Instead of transferring data directly between private areas (that is, between users' workspaces), AccuRev GUI 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 GUI 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 GUI has numerous safeguards that prevent one user from overwriting another user's changes, and tools that help you resolve conflicting changes when they occur. 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 GUI 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

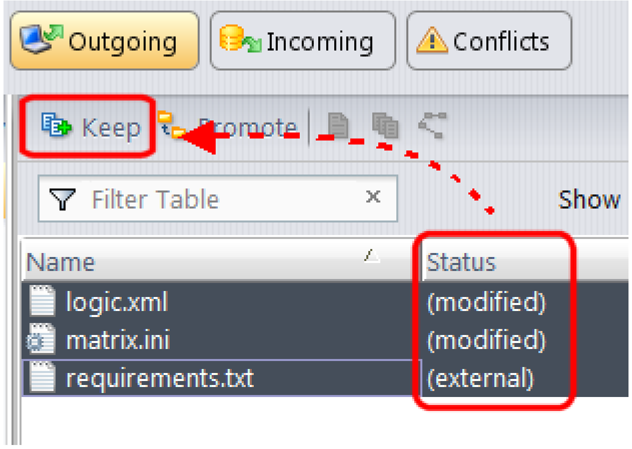
With the usage model described above, you will be able to accomplish most of your work with four simple commands: `Keep`, `Promote`, `Update`, and `Merge`. These commands are described in the following sections.

Keep: Preserving Changes in Your Private Workspace

A 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.

You do not need to perform a *check out* operation or otherwise get permission 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 the `Keep` command.



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

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. 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:

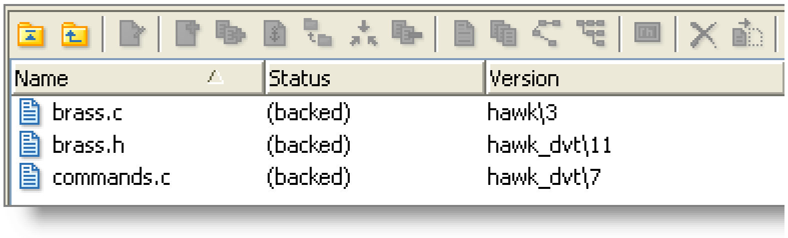
Rename and Move You can rename a file or move it to a different directory (or both), using AccuRev commands. Other users will continue to see the file at its original pathname in their workspaces.

Defunct You can remove a file from your workspace with the `Defunct` command. 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

AccuRev retains all of the versions that you `Keep`. 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 `myWorkspace\2` (which translates as "the 2nd version of this file created in the `myWorkspace` workspace").

AccuRev keeps track of the status of each file in a workspace. After you `Keep` a file, the **Status** column in the **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* in *Promote* for more information). The **Version** column displays the `version-ID`.



Name	Status	Version
brass.c	(backed)	hawk\3
brass.h	(backed)	hawk_dvt\11
commands.c	(backed)	hawk_dvt\7

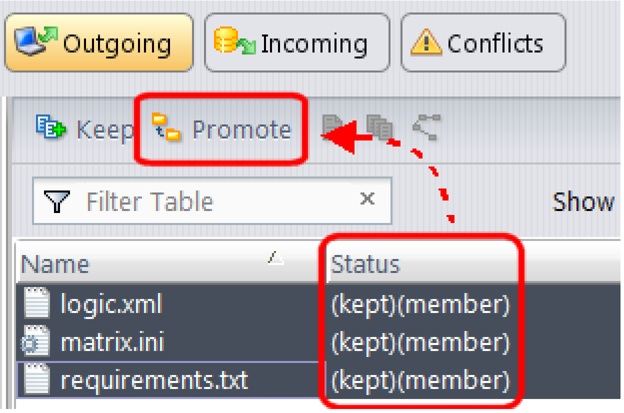
A change to the data within a file, recorded by `Keep`, is termed a *content change*. The change made by `Rename`, `Move` or `Defunct` is termed a *namespace change*. (Many SCM systems do not handle namespace changes at all, or have very limited capabilities in this area.) As noted previously, AccuRev Server 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 Server only needs to keep track of namespace changes, such as 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, when 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.

The following figure shows how to run the `Promote` command from the **File Browser** toolbar. `Promote` is also available from the **Actions** and context (right-click) menus.



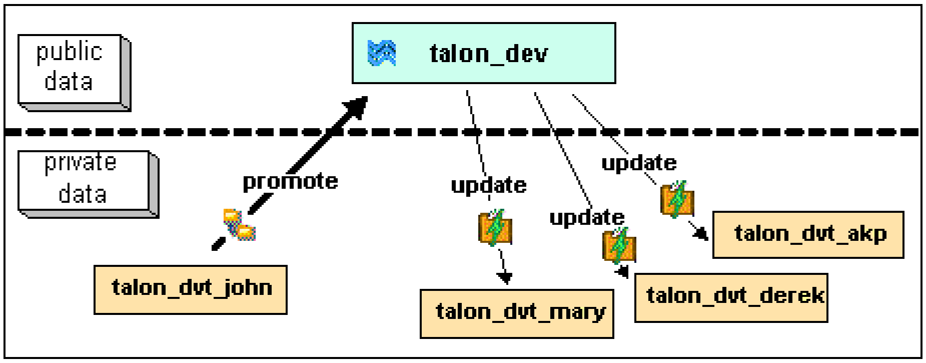
Name	Status
logic.xml	(kept)(member)
matrix.ini	(kept)(member)
requirements.txt	(kept)(member)

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 *Update: Incorporating Others' Changes into Your Workspace*.

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`
- Etc...

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*.

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.

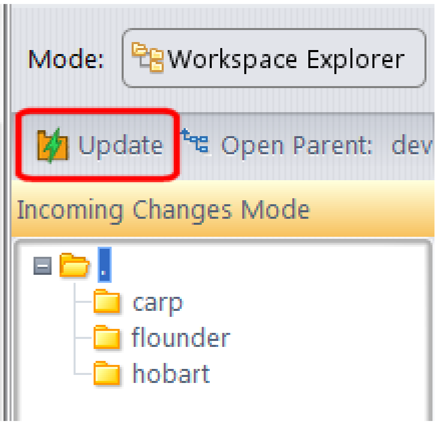
We have discussed the stream that is the basis for a set of workspaces. But a typical development project has many streams, organized into a hierarchy. Promoting a version to a higher-level stream from a lower level stream makes that version "even more public", for example, available to users outside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

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".

Note:

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*. 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*.

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 commands that "activate" a file.) When `Update` copies versions from the repository into your workspace, it skips over all active files.

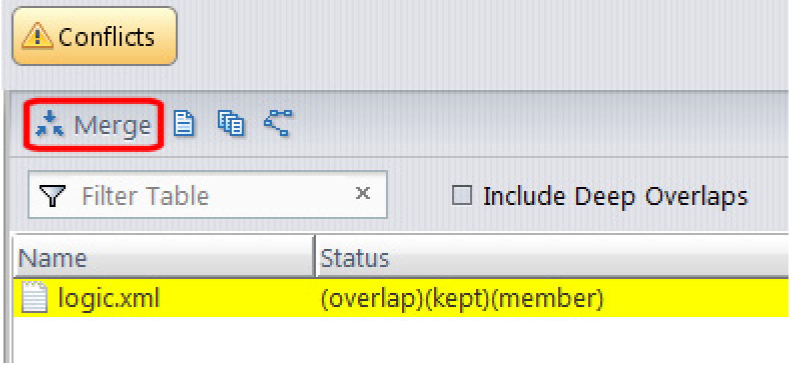
Note: Update can tell if you have modified a file but have not yet stored the changes in the repository as a new *Keep* version. It uses *time-stamps* and *checksums* to determine this. The presence of such files prevents the update from proceeding if updating would overwrite one or more of them with the backing-stream versions. 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 run the `Merge` command from the **File Browser** toolbar. `Merge` is also available from the **Actions** and context (right-click) menus.

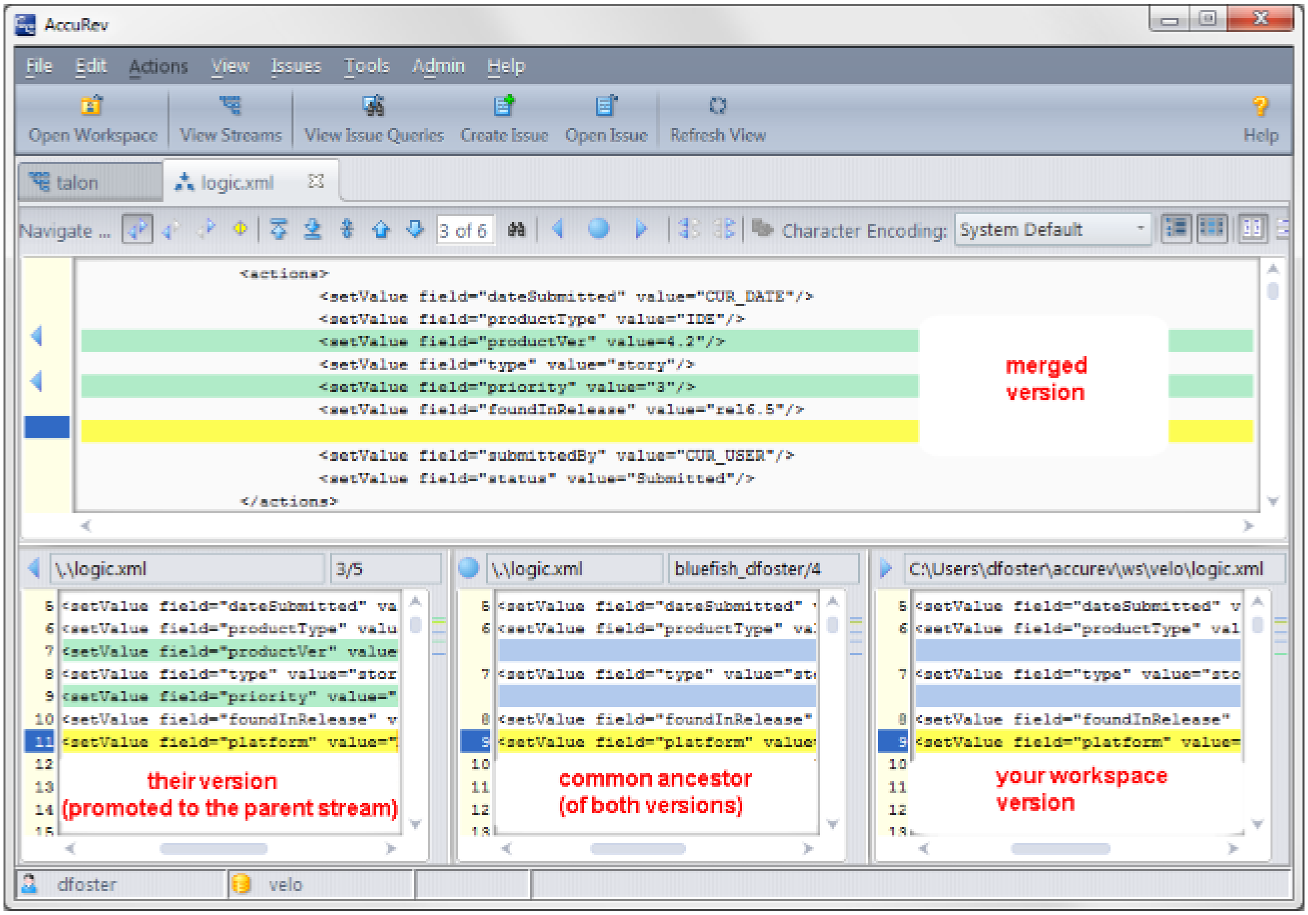
Tip: Use the **File Browser Conflicts** mode to view overlap and other conditions that prevent files from being updated and promoted.

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>
```

```
<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="
```

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

<setValue field="type" value="st

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

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

<setValue field="type" value="sto

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

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.

Index