CA Harvest Software Change Manager

Plug-In for Microsoft Visual Studio User Guide

Release 12.5

CA Technologies Product References

This document references the following CA Technologies products:

- CA Harvest Software Change Manager (CA Harvest SCM)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 4: Managing Changes in the Repository and Workspace 73

Index 79

Chapter 1: Getting Started

This section contains the following topics:

Basic Methodology

CA Harvest Software Change Manager (CA Harvest SCM) integrates with Microsoft development tools through the CA Harvest SCM Plug-In for Microsoft Visual Studio. The plug-in uses the following applications:

- **CA Harvest SCM**—Manages change processes to include the correct components in a released application.
- **Microsoft Developer Studio**—Provides an integrated development environment (IDE) that lets you build applications efficiently.

The CA Harvest SCM Plug-In for Microsoft Visual Studio uses Visual Studio packages and lets you access CA Harvest SCM user functions from within your integrated development environment (IDE).

Audience

You use the information in this guide to understand and use the CA Harvest SCM Plug-In for Microsoft Visual Studio. This guide is intended for anyone who wants to do the following:

1. Install the CA Harvest SCM Plug-In for Microsoft Visual Studio or upgrade from the Microsoft SCC API-based plug-in.
2. Configure the plug-in (such as the broker, Harweb server, third-party comparison tools, and mode preferences for working, logging, and so on).

   Administrators configure Visual Studio and CA Harvest SCM to work together. Administrators must be proficient users of Visual Studio and CA Harvest SCM. As an administrator, you must understand your CA Harvest SCM implementation including processes, lifecycles, and projects. You must also have administrative access to the CA Harvest SCM repository.

   Important! Two types of projects are referred to in this guide. Unless identified as a CA Harvest SCM project, the term *project* specifies a third-party project, such as a Visual Studio project.

3. Use the CA Harvest SCM Plug-In for Microsoft Visual Studio.

   Users (typically developers) create applications with Visual Studio and use CA Harvest SCM to manage source code. As a user, you must be proficient in Visual Studio and be familiar with the CA Harvest SCM lifecycle and CA Harvest SCM projects in which you work.

   Note: For information about Microsoft Developer Studio, see the Microsoft documentation for the IDE you plan to use. For more information about the Workbench concepts and functions, see the *Workbench User Guide*.

Lifecycle Projects

The term *project* refers to the control framework in CA Harvest SCM that supports a particular development or maintenance activity. A *lifecycle* is the part of a project that determines what activities can take place, when, and by whom. A lifecycle consists of states, processes, views, and access settings. The lifecycle can be considered the heart of a project, because it controls the flow of development life in it.

Contact your CA Harvest SCM Administrator to create and configure the CA Harvest SCM project to use in the Visual Studio IDE using the CA Harvest SCM Plug-In for Microsoft Visual Studio.

Note: For more information about implementing lifecycles and projects, see the *Administration Guide*.

Chapter 2: Installing and Configuring the Plug-In

This section contains the following topics:

Implementation Considerations

Before you install CA Harvest SCM Plug-In for Microsoft Visual Studio, consider the following:

- CA Harvest SCM Release 12.5 is backward compatible. However, Release 12.5 does not support the MS SCC-based plug-in. Release 12.0 clients can connect to a Release 12.5 server, except for Release 12.0 Plug-In for Visual Studio because it is based on MS SCC API.

  To upgrade from the MS SCC-based plug-in, do the following:

  1. Uninstall previous plug-in releases.
  2. Upgrade to the Release 12.5 CA Harvest SCM client.
  3. Install Release 12.5 CA Harvest SCM Plug-In for Visual Studio.

- The Visual Studio IDE must be installed before the installation of the plug-in.
- One or more of the supported versions of Visual Studio IDE can exist on the Visual Studio IDE target computer. All versions can use the integration.
- The CA Harvest SCM client Release 12.5 must be installed before the installation of CA Harvest SCM Plug-In for Microsoft Visual Studio. Upgrade to Release 12.5 when you have a previous release of the CA Harvest SCM client.
- The CA Harvest SCM server Release 12.5 must be installed on a network computer for your Visual Studio integration to connect to the CA Harvest SCM server.

Note: For system information and supported versions, see the *Release Notes*.

How to Implement the Plug-In

In general, you use the following steps to install and configure CA Harvest SCM Plug-In for Microsoft Visual Studio:

1. Read the *Release Notes*. Do not start your implementation until you read and understand that information.

   Read about and understand the CA Harvest SCM product components in the *Administration Guide* and the *Workbench User Guide*.

2. Read the implementation considerations.

3. Follow the operating system-specific steps in the *Implementation Guide* to install or upgrade the CA Harvest SCM server, client, and agent for your operating environment.

4. Install or upgrade the CA Harvest SCM Plug-In for Microsoft Visual Studio.

5. Create and configure the CA Harvest SCM project that you want to use in the Visual Studio IDE that works with the plug-in.

Note: All CA Harvest SCM documents are available in the Docs directory on the CA Harvest SCM installation media or at http://ca.com/support.

How to Prepare for Installation

Perform the following steps to prepare for installation:

1. Verify that you have installed the Release 12.5 CA Harvest SCM client before installing the CA Harvest SCM Plug-In for Microsoft Visual Studio.

2. Verify that a previous version of the plug-in does not exist on the installation computer. If a version does exist, uninstall it.

3. Verify that at least one of the supported IDEs is installed on the computer.

Note: The installation or installation process for CA Harvest SCM Plug-In for Microsoft Visual Studio can be lengthy depending upon the number of supported IDEs that are installed on your system. Do not close or stop the processes until they finish successfully.

Install on a Local Computer

You can install the CA Harvest SCM for Microsoft Visual Studio Plug-In on a local computer.

Note: The CA Harvest SCM Release 12.5 client must be installed on the local computer *before* you install the plug-in.

To install on a local computer

1. Close all Windows applications except Windows Explorer. Verify that you close Microsoft Visual Studio and any other IDE programs.

2. Verify that the CA Harvest SCM for the Microsoft Visual Studio Plug-In installation file (setup.exe) is accessible to your computers. For example, on a network computer or a shared drive.

   Note: You can find two installers for Visual Studio Plug-ins. One is for Visual Studio 2005, 2008, 2010 and the other installer is for Visual Studio 2012. Based on your Visual Studio IDE version, select the appropriate installer.

3. Copy the files to your computer and double-click the setup.exe file.

   Note: If you are installing on a Windows Vista computer, right-click the installer and select Run as Administrator from the shortcut menu.

   The installation wizard appears.

4. Click Next.

   The default folder is the installation directory of the CA Harvest SCM Release 12.5 client.

   Note: If the CA Harvest SCM client is not installed on the computer, an error message informs you that the client is required before you can install the plug-in.

5. Click Next through the remaining screens.

   The installation completes and resets the IDE to accommodate all the CA Harvest SCM-specific functionality in the IDE. The reset time varies depending upon your computer configuration and the Visual Studio version that is installed on your computer.

After You Install

After you install the plug-in, do the following:

1. Verify that CA Harvest SCM is installed and running on all CA Harvest SCM computers in your environment—not only on the computers on which you installed the plug-in.
2. Configure your existing solutions and projects for use with the plug-in. Configuration includes tasks that both CA Harvest SCM administrators and Visual Studio developers must perform.

Set Preferences

You configure user-specific preferences to specify how the plug-in works.

To set preferences

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.
2. Select and expand the Source Control node.
3. Verify that CA Harvest Software Change Manager is selected in the Current Source Control Plug-in drop-down list.

   The plug-in preferences appear.
4. Select the preference you want to set.

   The right window shows options for your selection.
5. Complete the fields as appropriate and click OK.

   Preferences are set for the plug-in.

Specify the Plug-In Provider

User-specific preferences let you switch from among different SCC providers for Visual Studio. To use the plug-in, you must specify CA Harvest Software Change Manager as the provider.

To specify the plug-in provider

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.
2. Select and expand the Source Control node.

3. Verify that CA Harvest Software Change Manager is selected in the Current Source Control Plug-in drop-down list.

   The plug-in preferences appear.

4. Click the Plug-in Selection tab.

5. Select CA Harvest Software Change Manager from the Plug-in Selection drop-down list, and click OK.

   CA Harvest SCM is the provider for use in the plug-in.

Specify the Plug-In Behavior

Preference options let you specify the following plug-in behavior:

To specify the plug-in behavior

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.

2. Expand the Source Control node, and click Behavior.

   The behavior options appear.

3. Select the check-out and commit options that you want to use:

   Checkout while editing the Read-only file (Online Mode)

   Automatically checks out the file you are attempting to edit when the file is not checked out.

   Commit edited item on Latest Trunk (Late Binding Mode)

   Commits the edited file by checking out (implicit, check-in without check-out) the latest version of the file. This option is helpful when you are committing on a branch.

   Ignore Items that are not part of local solution but exist in the file system

   Filters items that are not part of solution and displays the results in the Synchronizer. This option can optimize performance. You can clear this option when you want to know the repository changes to items that are not part of solution but present under the context view path.

   Click OK.

   The plug-in behavior is specified. The check-out and commit actions behave according to your selections.

Specify Compare/Merge Utilities

You select an external (third-party) utility for the CA Harvest SCM compare process to use, because the plug-in does not provide an internal compare/merge utility. After you select any of the utilities, and depending on if the selected utility is installed in the system path, the utility is invoked when you use the plug-in.

The following external compare/merge utilities (and their corresponding command lines) are supported:

- Araxis
- Beyond Compare
- DiffDoc
- Guiffy
- WinMerge

Note: You can also use custom utilities.

A two-way merge configuration is provided for Araxis, Beyond Compare, and Guiffy. The two-way compare-merge tool is invoked for the following scenarios:

1. The Synchronizer detects a conflict.
2. From the Synchronizer tree view, double-click the conflicted item, or right-click the conflicted item and select Compare with Repository.

or

1. The Harvest SCM Explorer tree or Lists view has an item with an M-tag.
2. From the Harvest SCM Explorer Lists view, right-click an M-tagged version and select the interactive merge process.

To specify compare/merge utilities

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.

2. Expand the Source Control node, and click Compare-Merge.

   The Compare/Merge options appear.

3. Complete the options as appropriate.

   Utility Type

   Specifies a utility, from a range of third-party merge utilities that are pre-installed on the local computer, to start when you invoke a compare operation in the IDE.

   Utility Path

   Specifies a utility and its corresponding path to use for compare/merge functions. When the selected utility is installed, and its executable file exists in the system path, you do not need to select a path. You can also select a utility by using the Utility Path selection button directly.

   CommandLine

   Specifies a command-line invocation argument for the utility type that you specified. The plug-in automatically produces the default command-line invocation argument for that utility. You can modify and save any of the command-line parameters for the preconfigured or custom utilities.

   Click OK.

   The compare/merge utilities are specified for the plug-in.

Specify Log Options

You select log options for the Harvest SCM output to use.

To specify log options

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.

2. Expand the Source Control node, and click Logging.

   The Logging options appear.

3. Complete the options as appropriate. The following fields require explanation:

   Log File Path

   Alters the Log file path. The default location for the log files is %TEMP%.

   Enable Debug Logging Using Debug level

   Enables debug logging and specifies a debug level. At the highest level, all debug logs are posted in the log file that is specified the Log File Path field.

Show all server messages in the Output

Shows all messages from the server in the Harvest SCM Output window. When you do not select this option, information and warning messages from the server do not appear in the output. For other messages (not from the server), this option does not have an impact.

Note: For operations like Open Solution from Harvest SCM Explorer or Add Solution to CA Harvest SCM, we recommend that you do not select this option when you are working with many files (for example, ~20000 files and folders).

Maximum number of logs allowed in the Output

Increases the maximum number of log entries that the Harvest SCM Output window can display. By default, the value is 500. After the maximum value is reached, old log entries are removed as new log entries are appended. The maximum value that you can specify is 5000.

4. Click OK.

   Log options are set for the Harvest SCM output.

Specify General and Work Mode Options

You specify the Harweb URL associated with the broker that the plug-in uses so that the CA Harvest SCM forms work correctly. In addition, work mode options let you work in Late Binding or Online mode.

To specify general and work mode options

1. Open the Visual Studio IDE, and select Tools, Options.

   The Options window appears.

2. Expand the Source Control node, and click Harvest SCM General.

   The General options appear.

3. Complete the options as appropriate. The following fields require explanation:

   Connected Broker

   Displays the connected broker information. This value is read-only and is available only when a broker connection is established.

   Logged in User

   Displays the logged-in user identification. This value is read-only and is available only when a broker connection is established.

   Change Password

   Changes the CA Harvest SCM user password.

Harweb URL

Specifies the Harweb URL that is associated with the broker that the plug-in uses. Harweb lets you view and update any forms that are associated with a CA Harvest SCM package.

Connect automatically while opening Managed Solutions

Checks the broker connection when you open managed solution files on a local file system. When a different broker manages the solution, a warning message appears and you can click Switch Broker to switch to the different broker.

Note: When you connect to a broker, select Save Password to save your password in the registry.

Click Apply.

4. Click the Work Mode node and set the mode options that you want to use in the plug-in. For example, you can switch from Late Binding (the default Work Mode) to Online mode, and conversely.
5. Click OK.

   General and work modes options are set for the plug-in.

How to Set Up Harweb Forms

Harweb forms are integrated with the plug-in. You set up the Harweb service on the broker and specify a Harweb URL in the plug-in preferences to use forms in the plug-in.

Note: For information about installing and configuring Harweb, see the *Implementation Guide*.

To set up Harweb forms for use in the plug-in, do the following:

1. Install, configure, and run Harweb on the broker.
2. Open the IDE.
3. Select Tools, Options, Source Control, and select Harvest SCM General.
4. In the Harweb URL text field, specify the full URL with the port number, for example:

   ```
   http://broker_computer_name:port_numumber/harwebr121
   ```

5. Click Apply.

   Harweb forms are set up.

How to Verify the Installation

To verify the installation, do the following:

1. Verify that the File menu includes a CA Harvest SCM submenu.
2. Verify that the File, New Project page includes an Add Solution to CA Harvest SCM check box.
3. Right-click the toolbar and verify entries for CA Harvest SCM Toolbar and CA Harvest SCC Toolbar. Select each option to verify that different dockable toolbars appear in the Visual Studio IDE, and which can be dragged and dropped at any location in the IDE.
4. Click various toolbar buttons to verify that different CA Harvest SCM tool windows appear in the IDE. For example, click Harvest SCM Explorer, Harvest SCM Output Window, Pending Changes, and so on.
5. Verify that Control Panel, Add/Remove program includes entries for CA Harvest Software Change Manager Release 12.5 Plug-in for Visual Studio.

   The installation and configuration of the CA Harvest SCM Plug-In for Microsoft Visual Studio is complete.

Note: If any of the verifications fail, the installation may be unsuccessful. Try closing all Visual Studio IDE instances, uninstall the plug-in and install it again.

How to Convert From an MS SCC API 1.2-Based Solution

You can convert your Visual Studio .NET projects (including web applications) that were checked in using the MS SCC API 1.2-based plug-in, to the Visual Studio package-based plug-in.

Note: This conversion does not apply if you are installing CA Harvest SCM client and server for the first-time and do not have any repository to upgrade.

To convert projects from an MS SCC API 1.2-based solution, do the following:

1. After you upgrade the product, create a package for the upgrade changes by using CA Harvest SCM Workbench.
2. Perform a check-out for update or concurrent update for the total solution and all the files recursively using the package that you created in the previous step. Use the Preserve and Create Directory Structure check-out option.

3. Open the local solution using the Visual Studio .NET IDE, and when you are prompted for credentials, press Cancel.
4. Select “Remove binding permanently,” click OK, Save All changes, and close the IDE.
5. Use the Workbench to check in the solution using the package.
6. If you have done the changes in branch using concurrent update, do a concurrent merge to verify that your changes are in trunk.
7. Delete all the local files of the solutions.

   Your solution and all the projects are upgraded and ready for the VsPackage-based plug-in to use.
8. Open the solution using the Open Solution from a CA Harvest SCM operation.

More information:

Open Solution From CA Harvest SCM (see page 45)

Remove the Plug-In

If necessary, you can remove the plug-in.

To remove the plug-in, open the Windows Add or Remove Programs, select CA Harvest SCM Plug-In for Visual Studio in the list, and click Remove.

The plug-in is removed.

Chapter 3: Using the Plug-In

This section contains the following topics:

How to Add a Solution to CA Harvest SCM

You can check in a newly created solution to the CA Harvest SCM repository from the Visual Studio IDE by performing an Add Solution to CA Harvest SCM operation. Add Solution to CA Harvest SCM checks in the files under the solution that you select in Visual Studio IDE to CA Harvest SCM, including the solution file. The operation does the following:

1. Navigates the hierarchy of the solution in the Visual Studio IDE, checks in each file, and creates the same hierarchy in a specified view path in CA Harvest SCM.
2. Updates the status for all items that exist in the Solution Explorer so that the files are source controlled.

Note: We recommend that you keep the solution and all its projects in a separate folder in the CA Harvest SCM repository. At the Add Solution to CA Harvest SCM wizard page, create a path in the repository, and select that path as the repository path to store all files. Then, the Synchronizer considers only the files and projects in that folder.

To add a solution to CA Harvest SCM, do the following:

1. Open the solution in the IDE.
2. Right-click the solution file in the Solution Explorer and select CA Harvest SCM, Add Solution to CA Harvest SCM.

   The login page appears.

3. Provide the credentials in the login dialog and click Next.

   The Set Context dialog appears.

4. Select package and context processes. Optionally, you can create a context package. The following fields require explanation:

   On Trunk

   Checks in your solution to the trunk, otherwise, the solution is checked in to a branch.

   CA Harvest SCM Package

   Specifies a package to use for the check-in.

   Note: The Populate and Select Default option automatically populates the default package and available processes for each type of process.

   Click Next.

5. Select the view path in the view tree for the solution check-in destination. Optionally, you can create a path by clicking Create View Path.
6. Click Finish.

   The solution is checked in to the CA Harvest SCM repository and the statuses are updated.

How to Add a New Project to a Managed Solution

You can add a new project to a checked-in solution. By doing this operation, the project and all the files under it are checked in to the CA Harvest SCM repository, the solution file is checked out, updated, and checked in.

To add a new project to a solution that is already managed in the CA Harvest SCM repository, do the following:

1. Open the solution from CA Harvest SCM.
2. Right-click the solution file and select Add, Project.
3. Add the new Visual Studio project to this solution.

   Note: A solution in the Solution Explorer uses a different status indicating that it is modified. A newly added project and its contents use a plus sign (+) status indicating that they are newly added.

4. Right-click a solution in the Solution Explorer and select CA Harvest SCM, Commit.

   The newly added project and the updated solution file are checked in.

How to Add a New File to a Managed Project and Commit

If you have a managed project, you can add a new file and commit the file to the CA Harvest SCM repository.

To add a new file to a managed project and commit, do the following:

1. Open Visual Studio IDE.
2. From the IDE, open a local solution (.sln) file that CA Harvest SCM manages. Otherwise, open a solution using the CA Harvest SCM explorer from the IDE.
3. Right-click the managed project in the Solution Explorer and select Add, New Item from the shortcut menu.
4. Select a file type and provide a file name. Click Add.

   The new file is added under the project. The status of the newly added file is shown, and the status of the project is modified to show the project file has been updated. Both the project file and newly added files display in the Pending Changes window.

5. Set a CA Harvest SCM context, and commit the changes to the CA Harvest SCM repository.

   The file is added to the managed project. You can create a folder under the project, add a new file under that folder, commit changes, and the new changes are published.

How to Add a New File to a Managed Solution and Commit

If you have a managed solution, you can add a new file and commit the file to the CA Harvest SCM repository.

To add a new file to a managed solution and commit, do the following:

1. Open Visual Studio IDE.
2. From the IDE, open a local solution (.sln) file that CA Harvest SCM manages. Otherwise, open a solution using the Harvest SCM explorer from the IDE.
3. Right-click the solution file in the Solution Explorer and select Add, New Item from the shortcut menu.
4. Select a file type and provide a file name. Click Add.

   The new file is added under the Solution Items folder. The status of the newly added file is shown, and the status of the solution is updated to show that the solution file is updated. Both the solution file and newly added files display in the Pending Changes window.
5. Set a CA Harvest SCM context, and commit the changes to the CA Harvest SCM repository.

   The file is added to the managed solution. You can create a solution folder under the solution, add a new file under that solution folder and commit the changes to the CA Harvest SCM repository.

Login and Authentication

When your site uses internal authentication (CA Harvest SCM Authentication), login is granted only if the user name and password are valid login credentials in CA Harvest SCM. If you attempt to log in with an expired password, the login attempt fails. In the Visual Studio Plug-in, you can use Tools, Options, Source Control, Harvest SCM General, Change Password to change your password. Alternatively, the Change Password dialog on the CA Harvest SCM Workbench lets you change your password when your site uses internal authentication.

When your site uses external authentication, such as Microsoft Active Directory, login is granted only if the user name exists in CA Harvest SCM and the user name and password are valid login credentials on the authentication server. If login fails due to password expiration, you are not prompted to change the password; change your password using methods provided by the authentication server.

Log In to CA Harvest SCM Using an IDE

You must be logged in to CA Harvest SCM to perform CA Harvest SCM functions from an IDE, such as opening a project under the CA Harvest SCM repository. When you attempt to open a Visual Studio project that is under CA Harvest SCM control, you are prompted to log in to CA Harvest SCM. If you cannot log in to CA Harvest SCM for any reason, you can optionally work offline (not connected to CA Harvest SCM). However, you may need to synchronize your files when you log in the next time.

You can open the login dialog from the following locations:

- Harvest SCM Explorer—New Broker Connection
- CA Harvest SCM-managed solution from a local file system—New Broker Connection
- Add Solution to CA Harvest SCM—New Broker Connection

To log in using the Harevest SCM Explorer

1. Open the Harvest SCM Explorer window in the IDE.
2. Click New Broker Connection from the toolbar of the Harvest SCM Explorer.

   The login dialog appears and you can use it to log in.

To log in using a CA Harvest SCM managed solution from local file system

1. Open the Visual Studio IDE.
2. From the IDE, open a local solution (.sln) file that CA Harvest SCM manages, and the broker connection to the corresponding CA Harvest SCM broker is not available.

   The login dialog appears and you can use it to log in.

To log in when you add a solution to CA Harvest SCM

1. Open the Visual Studio IDE.
2. In the IDE, create a solution (with or without a project).

   The login dialog appears and you can use it to log in.

Set Context or Edit Context

You set or edit a context using the Set Context or Edit Context dialogs, respectively, as follows:

- When you add or associate a solution to CA Harvest SCM, the Set Context dialog appears and you can edit all the following objects and processes.
- After CA Harvest SCM manages the solution, you can change the CA Harvest SCM package and processes using the Edit Context dialog on a managed file. When a solution is on a branch, you cannot change its associated package.

You have the following context options:

- Project
- State

  Note: If you open a solution from a snapshot in the CA Harvest SCM repository, you can change the state.
- View
- Package

  Note: For more information about the restriction information of packages, see the Restriction on Changing the Context Package section.
- Check Out
- Check In
- Rename Item
- Rename Path
- Move Item
- Move Path
- Remove Item
- Remove Path

You invoke the Edit Context dialog from the following locations:

- Solution Explorer
- CA SCC Toolbar (Global)

To edit a context from the Solution Explorer

1. Verify that you have a CA Harvest SCM managed solution open in the Visual Studio IDE.
2. Right-click a managed file in Solution Explorer.
3. Select CA Harvest SCM, Edit Context from the shortcut menu.

   The Edit Context dialog appears and you can use it to edit the context.

To edit a context from the CA SCC Toolbar (Global)

1. Verify that you have a CA Harvest SCM managed solution open in the Visual Studio IDE.
2. Select View, Toolbars, CA SCC Toolbar.

   The toolbar appears.
3. Select a managed file in Solution Explorer.
4. From the CA SCC Toolbar, click Edit Context.

   The Edit Context dialog appears and you can use it to edit the context.

Restrictions on Changing the Context Package

You cannot change the context package in the following circumstances:

1. When the solution file is exclusively checked out using another package due to any of the following reasons:
   - Adding / removing of a new project
   - Adding / removing of a solution item / folder
   - Renaming of any project file
2. When the solution is checked out.
3. When the solution is on a branch.

Note: To change the context package, commit the changes and perform concurrent Merge on the package followed by Synchronize and Get Latest Versions.

View Current Context Package

Context package refers to the package that is set in the current context. This package is used for any transaction that is sent to CA Harvest SCM repository. When you want to see the package contents (including forms, items) or perform some package-related operations on the context package, you can directly view the context package without navigating the folder structure in Solution Explorer.

To view the context package

1. Select the solution file in the Solution Explorer.

   All the packages in the solution are listed.

2. Right-click the Solutions Explorer and select Show Context Package or click the icon in the CA Harvest SCM toolbar.

   The SCM Explorer window opens, shows the context project, and highlights the context package. You can then view the changes checked in to the context package.

   Note: If you have not selected any package in the context, it navigates to the Packages node under the respective project and state.

More information:

Set the Context Package from the List View (see page 43)

View Package Associated with a Checked out Item

Every item that is checked out is associated with a package.

To view the package that is associated with a checked out item

1. Click the Pending Changes Toolbar button.

   The Pending Changes dialog opens.

   For each checked out item, an entry in the Package field indicating the package that is used to check out that specific item displays.

Note: The *only* items that are checked out exclusively in Late Binding Mode are associated with a context package. The other items in the Late Binding Mode, are not associated with the context package. However, in Online Mode every change is associated with a context package. For more information about the Late Binding Mode and Online Mode, see the Online and Late Binding Working Modes (see page 71) section.

CA Harvest SCM Packages

Packages in CA Harvest SCM describe how files are added or modified. Packages also move changes from one state in the lifecycle to another state.

You can add as many packages as you want in the lifecycle in your project. However, if you do not have at least one package defined initially, you cannot check out any files for update or check in files to your project. (Although it is possible to open a solution from the CA Harvest SCM the repository in read-only (Browse) mode.) Additionally, users should create their own individual packages for adding or modifying project files in CA Harvest SCM. We recommend that multiple users do not use the same package.

Not all functionalities that you can execute on a package are available from the plug-in. Processes such as cross project merge, switch package, delete version, move package, take snapshot view, and so on, cannot be executed from the plug-in. However, you can use the CA Harvest SCM Workbench to access all CA Harvest SCM user functions.

The Harvest SCM Explorer tree and lists view lets you execute processes and actions on packages. The following processes are available from a package node:

- Approve Package
- Promote Package
- Demote Package
- Delete Package
- Concurrent Merge

The following actions are available from a package node:

Rename

Renames a package.

Properties

Displays the following properties of the selected package:

Name

Specifies the name for the package.

Assign to

Specifies the Package Assignee name.

Package Groups

Specifies the package groups to be associated with the package. Use the add/remove buttons to add/remove package groups.

Comment

Specifies package description.

Note

Provides notes about the create package process.

Approval

Provides approval history of the current package to discover who approved a package and when it was approved.

History

Provides the history of a package to discover who created, promoted, or demoted a package and when the actions occurred.

You can select multiple packages from the Lists view of the Harvest SCM Explorer and perform these operations on them.

Note: For information about setting access, see the *Administrator Guide*.

CA Harvest SCM Forms

From the Harvest SCM Explorer Tree view, you can perform the following actions on a Form node:

- Rename a form.
- Double-click a form to view and update the form that is associated with the package. The Harvest SCM Explorer Forms view opens through the Harweb interface.

CA Harvest SCM Versions

From the CA Harvest SCM Explorer Lists view, you can perform the following processes on a version:

- Compare
- Compare with Parent Trunk
- Interactive Merge

You can execute the following commands to view the comparisons from the CA Harvest SCM Explorer Lists view:

Compare

Compares any two versions of a file in the SCM Explorer Lists view. You right-click the versions and select Compare. The compare window opens with the selected files.

Compare With Parent Trunk

Compares any version of the file in the SCM Explorer Lists View. You right-click the version and select Compare with Parent Trunk. The compare window opens with the file selected and the parent trunk version of the same file. You can use this option to perform code review from within the Visual Studio IDE.

Note: Verify that you have at least one check-out process that includes the check-out for browse mode and does not have pre-linked or post-linked UDPs associated with it. The plug-in requires this check-out definition to get the item from the CA Harvest SCM repository whenever you invoke a compare/merge.

Interactive Merge Process

The interactive merge process lets you do the following:

- Combine the changes made on unmerged branches with changes on the trunk.
- Resolve a merge-tagged version after it is created by the concurrent merge or cross project merge process.

The interactive merge process lets you perform a two-way or three-way merge depending on the preference settings of the external compare-merge tool (under Tools, Options).

To perform an interactive merge

1. In the Visual Studio IDE, open Harvest SCM Explorer.
2. Navigate to the merge-tagged (M) version you want to resolve.
3. Right-click the version, and select interactive merge process from the shortcut menu.

   The two versions of the item being merged appear in the merge dialog with common blocks (lines identical in both versions) and conflict blocks. Conflict blocks are positioned side-by-side and are outlined.

   Note: A conflict occurs when the same line or block of data is modified in both the branch and the trunk. Insertions and deletions are considered changes.
4. Resolve conflicts by manually editing the left pane text or by clicking the following toolbar actions:
   - Copy all nonconflicting changes from right to left.
   - Copy current change from right to left.

   The left pane represents the branch version being merged; the right pane represents the latest trunk version.
5. Save changes and close the third-party merge tool.

   The interactive merge process dialog appears.

6. Specify values to use for the merge; the versions for the Status, Name, and Path fields can be different.
7. (Optional) Click Note.

   Information about the process appears.
8. Click OK.

   A new, normal-tagged (N) version replaces the merge-tagged (M) version.

More information:

Specify Compare/Merge Utilities (see page 16)

Package Group Support in the SCM Explorer

From the Harvest SCM Explorer, you can browse through package groups. Package groups are visible from each state node in a project and directly under a Project node. You can select multiple package groups from the List view of the Harvest SCM Explorer. Right-click a package group node under a state, to perform the following processes and actions:

Processes

You can execute the following processes on a Package Group node:

- Approve Package
- Promote Package
- Demote Package
- Delete Package
- Concurrent Merge

Actions

You can execute following actions on a Package Group node:

Rename

Renames a package group.

Delete Package Group

Deletes a package group.

Properties

Displays the following properties of the selected package group:

Name

Specifies the name for the package group.

Packages

Specifies the packages to be associated with the package group. Use the add/remove buttons to add/remove packages.

Bind Packages

Binds the selected packages together. For more information about binding, see *Workbench User Guide*.

Note

Specifies notes about the package group, for example, the reason for creating the package group.

Create a Package Group

You can create a package group to group one or more related packages.

While you work in Visual Studio, you can easily create a CA Harvest SCM package group to group one or more packages. The new package group is created in the project and is visible from one or more states from where the packages belong to the package group.

To create a package group

1. Open the Harvest SCM Explorer.
2. Connect to the broker if you are not already connected.
3. Navigate to a state node under a project and expand the state node.
4. Right-click the Package Groups node under the state node, and select New Package Group.

   The Create Package Group dialog opens.
5. Specify the name of the package, the packages you want to associate with the group, and the packages you want to bind. Click OK.

   The package group is created under the project. The package group is also visible from one or more states to which the associated packages belong. If you have not associated any package to the package group, the package group is visible only at the project level.

Add Multiple Packages to a Package Group

If you want to associate additional packages to a package group, after creating the group, you can add those packages to the group.

To add multiple packages to a package group

1. In the Harvest SCM Explorer, right-click the package group to which you want to add the package and select Properties.

   The Properties dialog opens.
2. In the Packages field, click Add to add a package and repeat this process until you are done with all the packages you wanted to add.
3. Click OK.

   The selected packages are added to the group.

Add a Package to Multiple Package Groups

If you want to associate a package to multiple package groups, you can add the package to the groups that you need.

1. In SCM Explorer, right-click the package you want to add to the package group and select Properties.

   The Properties dialog opens.

2. In the Package Groups field, click Add to add a package group. Repeat this process until you are done with all the package groups to which you want to add the package.

3. Click OK.

   The package is added to the selected package groups.

Package Dependency Support in Promote and Demote Dialogs

You can view the list of dependent packages and their associated versions that make them dependent for a particular package. The Package Dependency report is displayed in the Promote or Demote dialogs when you promote or demote a package.

View, Add, and Promote Dependent Packages

Perform the following steps:

1. Open the Harvest SCM Explorer.
2. Connect to the broker.
3. Navigate to a state node under a project and expand the state node.
4. Expand the Packages node under the state node.
5. Right click the required package and select the appropriate promote process.
   The Promote Dialog opens.
6. Click the Dependent Packages tab to view the package dependency report.
7. Select the Show Version Details check box.
8. Select a package in the Dependent Packages list to view the versions causing dependency and all others versions available in that package.

   The version details appear in the Versions Causing Dependency list and the All Other Versions list.
9. Click the Add All To Promote List button.
   All the dependent packages are added to the Promote List and the Option tab is displayed.

   Note: If the dependent packages are present in other states, an appropriate message is displayed.
10. Click OK.
    All the dependent packages are promoted together with the selected package.

View, Add, and Demote Dependent Packages

Perform the following steps:

1. Open the Harvest SCM Explorer.
2. Connect to the broker.
3. Navigate to a state node under a project and expand the state node.
4. Expand the Packages node under the state node.
5. Right click the required package and select the appropriate demote process.
   The Demote Dialog opens.
6. Click the Dependent Packages tab to view the package dependency report.
7. Select the Show Version Details check box.
8. Select a package in the Dependent Packages list to view the versions causing dependency and all others versions available in that package.

   The version details appear in the Versions Causing Dependency list and the All Other Versions list.
9. Click the Add All To Demote List button.
   All the dependent packages are added to the Demote List and the Option tab is displayed.

   Note: If the dependent packages are present in other states, an appropriate message is displayed.
10. Click OK.
    All the dependent packages are demoted together with the selected package.

Add and Promote or Demote Dependent Packages Without Viewing Package Dependency Report

Perform the following steps:

1. Open the Harvest SCM Explorer.
2. Connect to the broker.
3. Navigate to a state node under a project and expand the state node.
4. Expand the Packages node under the state node.
5. Right click the required package and select the appropriate promote or demote process.
   The Promote or Demote Dialog opens.
6. Select the Add Dependent Packages check box.
   All the dependent packages are added to the Promote or Demote list.
7. Click OK.
   All the dependent packages along with the selected package are promoted or demoted.

How to Create Packages

While you work in Visual Studio, you can easily create a CA Harvest SCM package to check out one or more items. The new package is created in the project and in the initial state specified by the create package process defined for the project and state of the item that you selected.

To create the package, do the following basic steps:

1. Click the Create Package button on any of the following CA Harvest SCM elements in Visual Studio:
   - Harvest SCM Explorer
   - Set Context dialog
   - Edit Context dialog

   The Create Package process dialog appears.
2. Complete the following fields on this dialog:

   Name

   Specifies the name for the new package.

Assign To

Specifies the user to whom this package will be assigned. The default is the user currently logged in to CA Harvest SCM using Visual Studio.

Package Groups

Specifies the Package Groups to be associated with the package. Use the add/remove buttons to add or remove package groups.

Set this package in the existing Solution Context (Optional)

Sets the newly created package as the context package. This helps in the changing the context package while creating the package.

Comment

Specifies comments about the package, such as the reason for creating the package.

3. Click OK.

   The package is created in the project and in the initial state specified by the create package process.

Create a Package from the Harvest SCM Explorer

Without leaving Visual Studio, you can use the Harvest SCM Explorer to create a CA Harvest SCM package. The new package is created in the project and in the initial state specified by the create package process defined for the project and state of the item that you selected.

To create a package from the Harvest SCM Explorer

1. In the Visual Studio IDE, select View, Harvest SCM Explorer.

   Note: A connection to the CA Harvest SCM broker must exist.

2. Expand the Broker node in the tree, and then the project and state nodes.
3. Right-click the Packages node and select a create package process from the shortcut menu.
4. Complete the fields on the create package process dialog. Click OK.

   A new package is created in the initial state specified by the create package process.

Create a Package from the Set Context Dialog

Without leaving Visual Studio, you can use the Set Context or Edit Context dialog to create a CA Harvest SCM package.

To create a package from the Set Context or Edit Context dialog

1. Navigate to the Select Package and Processes group and click the plus sign (+) next to the Package field.

   The create package process dialog appears.

2. Complete the fields on the create package process dialog. Click OK.

   The package is created in the project and in the initial state specified by the create package process defined for the project and state.

Set the Context Package from the List View

You can quickly set the context package of your solution, directly from the SCM Explorer List view instead of using the Edit Context dialog.

To set the context package from the Harvest SCM Explorer List view

1. Verify that a managed solution is loaded in the IDE.
2. Navigate to the Context path in the left tree.
3. Expand the Project, State context, and select the Packages node.

   The Lists view lists packages.

4. Right-click the package for which you want to set the context, and select Set to Solution Context from the shortcut menu.

   The context package of the loaded solution is set.

To set to the solution context from the create package process

1. Verify that a managed solution is loaded in the IDE.
2. Navigate to the context path in the left tree.
3. Expand the Project, State context, and select the Packages node.
4. Right-click the Packages node, and select Create Package.

   The Create Package dialog appears.

5. Complete the Create Package dialog fields including the “Set this package in the existing Solution Context” check box.
6. Click OK.

   The package is created and is set to the solution context.

More information:

View Current Context Package (see page 31)

Update Package Properties

If you want to change the package properties after creating the package, you can update the same.

To update the package properties

1. Right-click the package and select Properties.

   The Properties dialog opens.

2. Update the property you want to change and click OK.

   The changes are saved.

Display the CA Harvest SCM Toolbars in the IDE

You can display and use the following CA Harvest SCM toolbars in the IDE:

CA Harvest SCM Toolbar

Provides quick access to frequently used advanced features of the plug-in. Displays the following CA Harvest SCM windows:

- Pending Changes
- Harvest SCM Output
- Harvest SCM Explorer

CA Harvest SCC Toolbar

Provides the following source control commands:

- Add Solution to CA Harvest SCM
- Checkout
- Commit
- Version List
- Refresh Checkout Status
- Refresh Checkout Status (Recursive)
- Edit Context
- Undo
- Synchronize
- Get Latest Version
- Properties

To display the toolbars in the IDE, right-click in the blank toolbar area of the IDE and select either or both of the toolbars.

Open Solution From CA Harvest SCM

When you open a solution from CA Harvest SCM, the following occurs:

- All its projects open.
- The solution includes the latest version of all its files based on the package and working view you selected.

Alternatively, you can open a solution that is already checked out to a local directory by providing login credentials.

To open a solution from the CA Harvest SCM repository

1. Open Visual Studio IDE.
2. In the CA Harvest SCM Toolbar, click the Harvest SCM Explorer button.

   The Harvest SCM Explorer toolbar appears in the IDE.
3. In the Harvest SCM Explorer, expand the broker, project, state, and so on, and navigate to the solution you want to open.
   - When the solution is on a branch, the solution is located on the version list of the package for which it is checked in.
   - When the solution is on a trunk, the solution is located in the package version list and in the corresponding working view.
4. Right-click the solution file and select Open Solution from the shortcut menu.

   The Open Solution dialog appears.
5. Select the client path where the solution is to be checked-out on the local computer.

   Note: When you open a solution from the working view, you can specify a package to check out the package and working view contents.
6. Select Replace Read-Only files, Replace writable files in the Open Solution dialog if you want to replace read only and writable files, respectively.
7. Click OK.

   The required files in the local file system are checked out and the solution opens in the IDE. The statuses are updated and show the status of the items in the repository.

How to Use a Private Solution

When you want to work only on a subset of projects or websites in a solution that includes numerous projects or websites, you can use a private solution.

A private solution lets you do the following:

- Focus and work on selected projects or websites rather than including other unwanted projects or websites in a solution.
- Improve performance, because the plug-in manages fewer items.

Consider the following information about a private solution:

- You cannot check in the solution file or monitor it for changes because the solution resides only on the local computer.
- You can perform any type of changes like edit, check out, undo, commit, move, rename, and delete on files and folders like in a normal managed solution.
- We do not recommend that you perform refactoring operations such as rename or remove directly on the project or website. Refactoring a project or website makes the master solution file obsolete. Renaming the project file is supported, however make sure that the master solution file is updated manually. The plug-in attempts to stop remove on the project or website due to technical limitations.
- You can add managed Visual Studio projects such as C++, C#, and VB, and managed websites of type “File System” and HTTP”.
- Add Project to Solution or Add as Website to Solution is not supported for a project file that has a branch version.

Add Managed Project to a Private Solution

To be able to work on managed projects from a private solution, you must first add them to your private solution. You can either add a parent folder that contains multiple projects or add specific project folder, depending on whether you will work on most the projects under a parent folder or only on few of them. If you will be working only on few projects in the parent folder, it is a best practice to add those projects individually to the private solution instead of adding the parent folder.

To add managed project to a private solution

1. Select File, New, Project, Visual Studio Solutions, Blank Solution in the Visual Studio IDE.

   A blank solution is created. This solution acts as private solution to hold the managed projects.

2. On the CA Harvest SCM Toolbar, click the SCM Explorer button.

   The SCM Explorer appears.

3. Expand the Broker, Project, State, and View, and navigate to the location of the projects that you want to add to the blank solution.

4. In the Lists view of the SCM Explorer, right-click the project directory that you want to add, and select Add Project to Solution from the shortcut menu.

   The Add Project to Solution dialog opens with the following options:

   Checkout Process

   Lists the available checkout process. Select the checkout process that has at least a browse mode enabled.

   Root Context View Path

   Specifies the root view path for the list of projects that are going to be added to the local solution.

   Client Path

   Specifies the client path.

   Solution Name

   Specifies the name of the solution. This field is disabled because it takes the existing private solution name as input.

5. Click OK.

   The Select Projects dialog appears.

6. Select the projects you want to add to the blank solution, and click OK.

   The selected projects are added as a private solution on your local computer.

Add Managed Website to a Private Solution

To be able to work on managed websites from a private solution, you must first add them to your solution. You can either add a parent folder that contains multiple websites or add specific website folder, depending on whether you work on most of the websites under a parent folder or only on few of them. If you work only on few websites in the parent folder, it is a best practice to add those websites individually to the private solution instead of adding the parent folder.

Note: CA Harvest SCM adds a folder as a website. When you add a parent folder that has multiple websites, the complete folder is added as a single website and you cannot select specific websites in the parent folder that you want to add to the private solution.

To add managed website to a private solution

1. Select File, New, Project, Visual Studio Solutions, Blank Solution in Visual Studio IDE.

   A blank solution is created. This solution acts as private solution to hold the managed websites.

2. On the CA Harvest SCM Toolbar, click the SCM Explorer button.

   The SCM Explorer appears.

3. Expand the Broker, Project, State, and View, and then navigate to the location of the websites that you want to add to the blank solution.

4. In the Lists view of the SCM Explorer, right-click the website directory, and select Add as Website to Solution.

   The Add as Website to Solution dialog opens with the following options:

   Checkout process

   > Lists the available checkout process. Select the checkout process that has at least a browse mode enabled.

   Root Context View Path

   > Specifies the root view path for the list of websites that are going to be added to the local solution.

   Client Path

   > Specifies the client path. For local IIS websites, provide the IIS location (Example: C:\Inetpub\wwwroot). For Local "File System" websites, provide any path on your local system.

   Solution Name

   > Specifies the name of the solution. This field is disabled because it takes the existing private solution name as input.

5. Click OK.

   The Add Existing Website dialog appears.

6. Select the type of website you want to add. For File System websites, click File System and locate the directory where the website folder is checked out. For Local IIS website, click Local IIS and select the website folder listed in the Local Web Servers tree and click Open.

   The selected website is added to the private solution on your local computer.

Open Solution From the Local File System

You can open a solution from your local file system.

To open a solution from your local file system

1. Open the Visual Studio IDE.
2. Go to File, Open Project, and navigate to the solution (.sln) file that was already added to CA Harvest SCM.

   If you are not connected to the CA Harvest SCM broker already, the login dialog appears.
3. Provide your login credentials and click OK.

   The solution and all other files open and the Solution Explorer displays their statuses.

Switch Broker

Multiple Visual Studio solution files can exist under different brokers. Because the plug-in shows only one broker connection at a time, when you want to switch between solution files, you must switch broker connections. The Connect automatically while opening Managed Solutions option enables the switch broker capability. When you open managed solution files on a local file system, the plug-in checks the broker connection. When a different broker manages the solution, a warning message appears and you can click Switch Broker to switch to the different broker.

More information:

Specify General and Work Mode Options (see page 18)

Open Solution for Website Project Paths

When opening a solution that has website projects, open it to the client path in which the website projects were present earlier. If many end users are using the solution in their respective systems, create the solution and projects on a mapped drive (for example, O:\ ). A mapped drive folder lets users open the solution to a common root directory. For example, when the solution and all its projects are under O:\SolutionPath, and you perform an Add Solution to CA Harvest SCM (user1 in computer1), others using the solution from within the Visual Studio IDE can use O:\SolutionPath as the client path. If the solution and website are in different locations under the same drive, the plug-in shows the website project items as newly added files and does not show them as managed.

Checkout

You can use the Visual Studio IDE to check out a managed file, but not an unmanaged file or new file, and reserve the file in the repository. Before the check-out occurs, the CA Harvest SCM context is verified. If the context is not set, an error message appears.

Use any of the following procedures to check-out files:

To check out files from the Solution Explorer

1. Verify that you have set a correct CA Harvest SCM context.

   From the Solution Explorer, right-click one or more checked in files that have normal or N-tags in the CA Harvest SCM repository, and select CA Harvest SCM, Checkout from the shortcut menu (or click Checkout from the CA Harvest SCC Toolbar).

   The files are checked out from the CA Harvest SCM repository. The checked-out versions are the latest versions in the CA Harvest SCM context.

To check out files from the Version List

1. Verify that you have set a correct CA Harvest SCM context.

   From the Solution Explorer, right-click one or more checked in files that have normal or N-tags in the CA Harvest SCM repository, and select CA Harvest SCM, Versions from the shortcut menu.

2. The files are checked out from the CA Harvest SCM repository.

   The Version List shows all versions for the selected file.

3. Right-click a version in the Version List and select Check Out from the shortcut menu.

   The selected file version is checked out from the CA Harvest SCM repository.

To check out files using silent Checkout—Online Mode

1. Verify that you are working in Online mode.
2. Attempt to edit a managed file that is not checked out.

   A dialog appears with check-out options.

3. Select one of the following options, and click OK.

   Check Out

   Checks out the file from the CA Harvest SCM repository.

   Check Out and Remember

   Checks out the file from the CA Harvest SCM repository, and updates the ”Checkout while editing the Read-only file (Online Mode)” preference setting. The next time if you edit a managed file, the file is automatically checked out without a prompt appearing. To clear the preference setting, go to Tools, Options.

   Check Out Latest Version (Keep Content from Workspace)

   Checks out the latest version from the repository but retains the file content from local workspace.

   Check Out Latest Version (Update Content from Repository)

   Checks out the latest version from repository and replaces the workspace file. Any changes made locally are overwritten.

   Don’t Check Out

   Cancels editing the file and does not check out files.

   The files are checked out.

Undo Changes

You can undo CA Harvest SCM operations in the Visual Studio IDE, such as check-out, move, rename, and delete. A modified file also can be undone, so that it restores the file back to the version existing in CA Harvest SCM repository.

For a checked-out or reserved file, the Undo command (from the plug-in) corresponds to the Release Only mode of the check-in process in CA Harvest SCM. This operation lets you release the reserve tag version without checking in any new changes. The reserve tag is released and the attributes of the files on the client are changed to read-only.

For local changes other than check-out, the Undo operation removes only local changes made to the file in workspace. Undo does not change the CA Harvest SCM repository.

To undo a checked-out file

1. From the Solution Explorer, check out a file.

   The status of the file is updated to mark it as checked out or reserved.

2. Right-click the checked-out file, and select CA Harvest SCM, Undo from the shortcut menu.

   CA Harvest SCM releases the reserve tag and a message appears in the output log. The red checkmark icon in the IDE is replaced with a cylinder icon (managed, checked in).

To use Late Binding mode to undo the local changes (add, edit, rename, move, and remove) made to a file

1. From the Solution Explorer, perform any of the following operations.

   a. Add a new file or project.

   b. Edit a managed file.

   c. Rename, move, or remove a file.

   The status of the file (or parent project or solution) is updated to mark it as edited in Late Binding mode.

2. Right-click the edited file (project or solution) and select CA Harvest SCM, Undo from the shortcut menu.

   The local changes made to the file are removed and the file is reverted to the corresponding version in the CA Harvest SCM repository.

Note: The Undo operation is also available from the Pending Changes and Synchronizer views.

Synchronize a Project

After you complete your project changes, synchronize and then commit your changes to the repository so that others can have access to the latest versions. While you were working on your files, other developers may have committed changes to the repository.

Synchronizing projects rather than individual files is the recommended and most efficient method of using CA Harvest SCM. You track only your project, not individual files. New, changed, and deleted files are tracked for you. You can synchronize multiple files or folders, or any combination of projects, folders, or files, using one synchronize operation. For each project or folder, all files that are descendents of the project or folders are included in the synchronization.

To synchronize a project

1. Navigate the Synchronizer view tree, and select the project that you want to synchronize.
2. Right-click the project folder, and select Synchronize from the shortcut menu.

   If you are synchronizing many files, an information meter shows the synchronization progress.

   The hierarchical structure of the folder you selected displays in the view. Arrows next to files show whether the change is incoming, outgoing, or conflicting.
3. (Optional) Click a toolbar icon to filter the changes:

   Show Incoming Additions

   Shows downward arrows with plus signs (v+) that indicate incoming additions. An incoming addition means that a new item has been added to the CA Harvest SCM repository. A Get Latest Version action transfers the new item to your Visual Studio solution.

   Show Incoming Changes

   Shows downward arrows with pencil signs that indicate incoming changes. An incoming change means that a new version has been created in the CA Harvest SCM repository. A Get Latest Version action transfers the new version to your Visual Studio solution.

   Show Incoming Deletions

   Shows downward arrows with minus signs (v -) that indicate incoming deletions. An incoming deletion means that either the item does not exist or a new D-tagged version is created for that item in the CA Harvest SCM repository. A Get Latest Version action deletes the corresponding item from your Visual Studio solution.

   Show Outgoing Additions

   Shows upward arrows with plus signs (^ +) that indicate outgoing additions. An outgoing addition means that a file, folder, or both is added to your Visual Studio items and is not yet in the repository. A commit transfers the new file, folder, or both to the CA Harvest SCM repository.

   Show Outgoing Changes

   Shows upward arrows (^) with pencil signs that indicate outgoing changes. An outgoing change means that a file, folder, or both is modified locally. Modification can be edit, rename, move, or a combination of them. A commit transfers the changes to the repository and create a version.

   Show Outgoing Deletions

   Shows upward arrows with minus signs (^ -) that indicate outgoing deletions and means that the file or folder is deleted locally. A commit creates a D-tagged version in the CA Harvest SCM repository.

Show Conflicting Changes

Shows double arrows (<->) that indicate that the file has been modified locally and remotely. The conflict must be resolved before you can perform a commit. Optionally, you can update your local Visual Studio solution with repository changes by performing a Get Latest Version action.

The view shows files that match the filter mode which you selected.

Note: You can right-click a file in tree to open a shortcut menu with options determined by the direction of the change.

4. (Optional) Click a toolbar button that corresponds with one of the following actions:

Commit

Commits the local changes to repository.

Note: This action is disabled when none of the selected items has an outgoing change or the selected item’s parent has an outgoing change and the parent node is not selected.

Get Latest Version

Updates your local items with repository changes.

Note: This action is disabled when none of the selected items has an incoming change or the selected item’s parent has an incoming change and the parent node is not selected.

Mark as Resolved

Confirms that the reported conflict of an item has been resolved. You can right-click a conflicting item and select Mark as Resolved to mark an item as resolved. When you perform a Mark as Resolved, the item appears as an outgoing change and you can Commit the changes to the repository.

Note: This action is disabled when more than one item is selected and the selected item does not have a conflicting change.

Show in Solution Explorer

Transfers the focus from an item in the Synchronizer view to its corresponding item in the Solution Explorer. This action is helpful when you are working in a large solution. You can right-click an item in the Synchronizer view and select Show in Solution Explorer to move the focus to the corresponding Solution Explorer item.

Note: This action is disabled when the selected item is not a file.

Compare with Repository

Compares the local file with its corresponding repository version. You can right-click an item in the Synchronizer view and select Compare with Repository to compare an item. Alternatively, you can compare any item in the Synchronizer view by double-clicking the item.

Get Repository—Actions

Enables the following advanced actions, which you can use in a specific scenario where a conflict occurs for an item that has been refactored on a local computer:

Get Repository—Name

Gets the name of the current repository version and lets you rename the local name with the current repository name.

Get Repository—Path

Gets the path of the current repository version and lets you move the local item to the location that corresponds to the current repository location.

Get Repository—Name and Path

Gets the name of the current repository version and lets you rename the local name with the current repository name.

Gets the path of the current repository version and lets you move the local item to the location that corresponds to the current repository location.

Note: These actions execute the refactoring changes only; they do not get the content of the current repository version. These actions can help you resolve conflicts easily. These actions may not work with folders.

The action you selected executes.

Use the Harvest SCM Explorer

Harvest SCM Explorer lets you navigate through the CA Harvest SCM objects under the connected broker. You can show CA Harvest SCM objects under connected broker, and list packages and versions.

To list CA Harvest SCM objects in a tree hierarchy

1. Click the New Broker Connection button on the Harvest SCM Explorer toolbar.

   Note: You can connect to only one broker at a time.

   The New Broker Connection dialog appears.

2. On the left window, expand the top node (connected broker) in the tree.

   All active CA Harvest SCM projects appear as child nodes under the broker tree. The following CA Harvest SCM objects require more explanation:

   View

   Displays the CA Harvest SCM view path object under a working view or another view path.

   Snapshot View

   Displays the CA Harvest SCM snapshot view in a snapshot state.

3. Select a node of any type in the tree.

   The list view is updated with minimum information that is specific to the object you selected.

4. (Optional) Refresh the selected CA Harvest SCM object in the tree view by clicking the Refresh button on the Explorer toolbar.

   The view is refreshed to show the latest information from CA Harvest SCM.

List Pending Changes

Pending Changes shows details of changes, including refactoring changes, that you made locally to a Visual Studio solution. For example, when you change a project file, the file is added to the Pending Changes view.

To list pending changes

1. Browse the Pending Changes window.

   All changed items under the solution (including the solution file, if changed) are listed. The following file changes update the list:

   - Dirty (changes unsaved)
   - Edited (changes saved)
   - Checked out
   - Renamed
   - Removed
   - Moved
   - Implicit check-out (for example, in Online mode, when you add a new file to a project in Visual Studio, the corresponding project file is implicitly checked out).
   - Implicit edit (for example, in Late Binding mode, whenyou add a new file to a project in Visual Studio, the corresponding project file is implicitly edited).

2. (Optional) Select files and click the corresponding toolbar button to perform any of the following actions:

   Commit

   Commits files.

   Undo

   Reverses changes made to files.

   Comment

   Provides check-in comments while performing a commit. A text area lets you enter comments and then continue with the commit operation.

View Children of an Item

You can view information about the children of a selected item.

To view children of an item

1. Select an item in the Synchronizer tree view.

   The Synchronizer list view shows the following information:

   - Name
   - Change
   - Change Type
   - Local Version
   - Repository Version
   - Local Name
   - Repository Name
   - Local Path
   - Repository Path

   Note: The Synchronizer Graphical Preview shows a preview of changes and provides suggestions for the next action that you can perform on the selected item. If the selected node has changes associated with it, a graphic illustrates the local version and repository version. The preview is optional and you can hide it by clicking the Hide Preview button in the Synchronizer view.

2. (Optional) Select files and click a toolbar button to perform any of the following actions:

   - Commit
   - Get Latest Version
   - Undo
   - Mark as Resolved
   - Show in Solution Explorer
   - Compare with Repository

   The action executes.

Commit Files to the CA Harvest SCM Repository

You can use the Visual Studio IDE to commit (check in) a reserved file, a new file, or a changed file. The following occurs before the commit:

- A prompt appears to provide a comment when the current CA Harvest SCM check-in process requires a comment during the check-in.
- The CA Harvest SCM context is verified. If the context is not set, an error message appears.
- Integrity is checked for the selected files to commit. When the integrity check fails, an error message appears.

Use any of the following procedures to check out files:

To commit files from the Solution Explorer

1. Verify that you have set a correct CA Harvest SCM context.
2. (Online or Late Binding mode) Right-click one or more files from the Solution Explorer to commit to the CA Harvest SCM repository, and select CA Harvest SCM, Commit from the shortcut menu.

   Note: You can also commit files by clicking the Commit button on the Solution Explorer toolbar.

   The selected files are committed to the CA Harvest SCM repository.

To commit files from the Pending Changes view

1. Open the Pending Changes view.
2. Verify that you have set a correct CA Harvest SCM context.
3. Select one or more files from Pending Changes to commit to the CA Harvest SCM repository, and click the Commit button on the Pending Changes toolbar.

   The selected files are committed to the CA Harvest SCM repository.

To commit files using the CA Harvest SCC toolbar button

1. Verify that you have set a correct CA Harvest SCM context.
2. Open the CA Harvest SCC toolbar.
3. Select one or more files from the Solution Explorer to commit to the CA Harvest SCM repository, and click the Commit button on the CA Harvest SCC toolbar.

   The selected files are committed to the CA Harvest SCM repository.

To commit files from the Synchronizer

1. Open the Solution Explorer.
2. Select one file from the Solution Explorer to commit to the CA Harvest SCM repository. You can select either a single file that you want to synchronize with the CA Harvest SCM repository, or you can select the parent of the file in the Solution Explorer hierarchy.
3. Right-click any of the selected files in the Solution Explorer and select CA Harvest SCM, Synchronize from the shortcut menu.

   Note: Alternatively, click the Synchronize button on the Solution Explorer toolbar.

   The Synchronizer view appears.
4. Right-click the files that you want to commit in the Synchronizer view, and select Commit from the shortcut menu.

   The files are committed to the CA Harvest SCM repository.

To commit files from the graphical view

1. Open the Solution Explorer.
2. Select either a single file that you want to synchronize with the CA Harvest SCM repository, or the parent of the file in the Solution Explorer hierarchy.
3. Select the files that you want to commit from the Synchronizer view and click the Show Graphical View toolbar button.

   The graphical view shows the current change from the repository version.
4. In the graphical view, click Commit to Repository.

   The selected files are committed to the CA Harvest SCM repository.

Update the Workspace From the CA Harvest SCM Repository

The Visual Studio IDE lets you update your local workspace for an open solution with versions from the CA Harvest SCM repository. Options let you update the local workspace for the entire solution or at any level under the solution. Updating the workspace fetches the latest available version for the selected files in the CA Harvest SCM repository and brings them to the local workspace. Be cautious when you update the workspace because it replaces the local files with the files from the repository.

To get the latest version using the Solution Explorer

1. Open the Solution Explorer.
2. Verify you that have set a correct CA Harvest SCM context.
3. Right-click the file in the Solution Explorer that you want to update, and select CA Harvest SCM, Get Latest Version from the shortcut menu.

   The selected file in the local workspace is updated with the latest version from the CA Harvest SCM repository.

Note: The Get Latest Version function is also available from the Solution Explorer toolbar button and the CA Harvest SCC toolbar button.

To get the latest version using the Synchronizer graphical view

1. Open the Solution Explorer.
2. Verify you that have set a correct CA Harvest SCM context.
3. In the Solution Explorer, right-click the file that you want to update, and select CA Harvest SCM, Synchronize from the shortcut menu.

   The Synchronizer window appears.
4. Select the file in the Synchronizer tree and click the Show Graphical View toolbar button.

   The graphical view shows the current change from the repository version.
5. In the graphical view, click Get Latest Version.

   The selected file in the local workspace is updated with the latest version from the CA Harvest SCM repository.

Remove Files

You may want to remove files from your local workspace, the CA Harvest SCM repository, or both. For example, you may want to remove a file from your local workspace temporarily for testing purposes. Additionally, you may want to remove a file from both the workspace and the repository when it is no longer needed.

To remove a file from the local workspace, the repository, or both

Note: Use the Late Binding mode to perform this operation.

1. Verify that a remove item process is available in your current context. If the remove item process is not available, do not use the Visual Studio IDE to remove a file from the CA Harvest SCM repository, because your local workspace and the repository will become unsynchronized.

   Note: To define a remove item process, use the CA Harvest SCM Administrator application. For more information, see the *Administrator Guide*.

2. Verify that all files that you want delete are *not* checked out. Deleting a checked out file causes the local workspace and the repository to become unsynchronized.

3. Select a file in the Solution Explorer and click Delete.

   A confirmation dialog appears.

4. Click OK.

5. Select the parent project or solution file, and commit the changes from the project or file to the CA Harvest SCM repository.

   The file is removed from your local workspace and CA Harvest SCM repository.

6. Read the CA Harvest SCM log to view if the task completed successfully. If the remove process does not work, the CA Harvest SCM repository may not reflect the change. Use the Workbench to fix a problem, and then synchronize your local workspace with the CA Harvest SCM repository.

   Note: You can also exclude files using the Exclude function. This function is a Visual Studio IDE function *only*. This function excludes the item from the project but does not remove the item from the local workspace or from the CA Harvest SCM repository.

Remove a Project From the Workspace

You may want to remove projects from your local workspace. For example, you may want to remove a project from your local workspace when it is no longer needed locally.

Important! The Remove Projects function is an IDE function only. The function removes the project from the IDE list of items, and CA Harvest SCM removes it from the repository and applies the delete tag (D) on the project.

To remove a project from the local workspace

1. Verify that you are in Late Binding mode to perform this operation. This operation may not work properly in Online mode.
2. Select a project in the Solution Explorer.
3. Click Remove.

   A confirmation dialog appears.
4. Click OK.

   The deletion is confirmed, and the project is removed from your local workspace only. The solution file is automatically checked out or edited (based on working mode).

   The solution file is visible in Pending Changes view.
5. Commit the solution file to CA Harvest SCM.

View File Properties

When CA Harvest SCM manages the Visual Studio solution, you can view CA Harvest SCM-specific information in the Visual Studio Properties.

To view the properties of a file, in the Visual Studio Properties Window select any managed file in Solution Explorer.

File information, such as local path, status, context information, associated package, process names, and so on, appears.

To view additional information about a file, right-click any managed file in Solution Explorer and select CA Harvest SCM, Properties.

A dialog appears and displays CA Harvest SCM repository information about the file.

Refresh Check-Out Status

For a managed file that CA Harvest SCM controls, you can update its status in the Solution Explorer. Depending upon the status of the file in CA Harvest SCM repository at that time, the file is refreshed with the latest status.

To refresh the checked-out status of a file

1. From Solution Explorer, select a managed file.
2. Right-click the file and select CA Harvest SCM, Refresh CheckOut Status from the shortcut menu.

   The status is updated.

Note: In a parallel development environment, this function is useful because it refreshes the local file and its status from the CA Harvest SCM repository. The updated status helps reduce server errors because the local file status is synchronized with the repository.

Refresh Check-Out Status Recursively

For a managed file that CA Harvest SCM controls, you can update its status in the Solution Explorer. Depending upon the status of the file in CA Harvest SCM repository at that time, all the files beneath it including the selected file are refreshed with the latest status.

To refresh the checked-out status of a file recursively

1. From the Solution Explorer, select a managed solution, project, folder, or file.
2. Right-click the selected node and select CA Harvest SCM, Refresh CheckOut Status Recursive from the shortcut menu.

   The status is updated for the selected node and any files beneath it.

Associate a Solution to CA Harvest SCM

A Visual Studio solution that is checked out from CA Harvest SCM clients, such as the Workbench, command-line utility, and so on, can work using the plug-in by associating the solution to CA Harvest SCM. The content and the hierarchy of all the files under the solution must match the hierarchy of the CA Harvest SCM repository.

Note: When a solution is opened in the IDE using the plug-in, you do not need to associate the solution.

To associate a local solution present in Visual Studio to CA Harvest SCM

1. Open a Visual Studio solution that CA Harvest SCM controls, but is checked out using a different CA Harvest SCM client such as the CA Harvest SCM Workbench.
2. Select the solution file in the Solution Explorer.
3. Select File, CA Harvest SCM, and select Associate to CA Harvest SCM.
4. Provide your login credentials and click Next.
5. Set the context and click Next.

   Note: If the solution is on a branch, select the respective package in the context.
6. Select the solution file.
7. Select the "All Files in the repository and that of the local machines are exactly the same” option, and click Finish.

   The solution is associated with the CA Harvest SCM repository.

   Note: An item that is part of a project file, but absent in the CA Harvest SCM repository, causes the association to fail. Conversely, if the item is absent in the local file system but is in the repository, the item is skipped and the association succeeds.

Disassociate a Solution From CA Harvest SCM

You can disassociate any CA Harvest SCM-managed solution. Disassociating a solution is useful if you do not want the local solution to have any CA Harvest SCM linkage. Before you disassociate a solution, consider the following:

- Any changes done on a solution that is disassociated from CA Harvest SCM are not checked in or tracked when you later associate the solution.
- You cannot work in Late Binding mode in a disassociated solution.
- When a solution is opened in the IDE using the plug-in, you do not need to associate the solution.

To disassociate a local solution in Visual Studio from CA Harvest SCM

1. Open a Visual Studio solution that CA Harvest SCM controls.
2. Select the solution file in the Solution Explorer.
3. Select to File, CA Harvest SCM, and select Disassociate from CA Harvest SCM.

   The solution is disassociated from CA Harvest SCM.

Compare Two Versions of a File

CA Harvest SCM lets you view the differences between two versions of a file. When you compare two versions of a file, CA Harvest SCM gets the version from the CA Harvest SCM repository and makes it available for comparison with the version on your local computer.

Use the Compare Versions process to compare the contents of the version in your local file system with the version in the CA Harvest SCM repository. Depending on the file comparison tool, you can merge the contents.

You can execute the compare command in the following ways:

- From the Solution Explorer, right-click an item and select one of the following options from the shortcut menu:

  Compare with Repository

  > Compares the local copy of the file with repository version. For example, if you have version 1.1.1 on your local computer and you edit it, this option shows the difference of the local and repository version: 1.1.1 plus local changes and 1.1.1. (For any N-tagged version, the local and remote files are the same.)

  Compare with Parent Trunk

  > Compares the local copy of the file with the parent trunk version. Using the previous example, 1.1.1 is compared with version 1.

  Compare with Latest Trunk

  > Compares the local copy of the file with that of Latest Trunk version available in the repository.

  The differences appear according to the option you selected.

- From the versions list, select two versions, right-click and select compare option from the shortcut menu.

  The differences between the two versions appear.

- From the Synchronizer view or graphical view, double-click an item that has local changes, or perform Compare with Repository.

  The compare window appears and shows the difference between the local changes and that of the latest repository.

The compare option is not available for an uncontrolled item (an item that CA Harvest SCM does not manage).

More information:

Specify Compare/Merge Utilities (see page 16)

Get

You can get the specific version of a file that is checked in to Visual Studio. If the file has been refactored, you are not allowed to get the previous version of file.

Note: Verify that you have at least one check-out process that includes the check-out for browse mode and does not have pre-linked or post-linked UDPs associated with it. The plug-in requires this check-out definition to get the item from CA Harvest SCM repository.

To get the specific version of a file in Visual Studio

1. Open the Visual Studio solution that is CA Harvest SCM controlled, and select a file for which you want to view a specific version.
2. Right-click the file for which you want to get a specific version, and select Versions List from the shortcut menu.

   The CA Harvest SCM Versions List lists entries for every version of the file.
3. Right-click the version you want to view and select Get from the shortcut menu.

   The file in the local workspace is replaced with the selected version of the file.

Note: Check-out for update is not allowed for an intermittent trunk. When you have replaced an intermittent trunk in the workspace, check-out for update results in an error. However, check-out for concurrent update is allowed on the replaced file.

Late Binding

You can modify files without checking them out. This method is best suited for concurrent development. Users can work in parallel and modify their local copy, and commit their completed changes. This manner of working is named *Late Binding* and can help prevent build problems. The Online mode with the branch placement option is almost equivalent to the Late Binding mode, where any developer is allowed to modify the files by creating their own branch. But the branch placement requires a check-out and does not provide a lock on the file whereas Late Binding does not require a check-out.

Late Binding can help you avoid the following check-out problems:

Note: These problems occur in Online mode with the trunk placement option.

- Can cause administrative problems.

  Sometimes Developer1 checks out a file and then forgets about it. Meanwhile, Developer2 is waiting to edit the file. This situation causes unnecessary delay and wasted time.

- Can create unnecessary locks on files.

  Suppose Developer1 is editing the beginning of a text file, and Developer2 wants to edit the end of the same file. Their changes do not overlap. Both developers can safely edit the file simultaneously, and later merge the file.

- Can create a false sense of security.

  Suppose Developer1 checks out and edits file A, while Developer2 simultaneously checks out and edits file B. Suppose that file A and file B depend on one another, and the changes made to each are semantically incompatible. Suddenly A and B do not work together anymore. The reserve-tag created by the check-out did not prevent the problem—yet it provided a false sense of security. Developer1 and Developer2 may think that by checking out files their changes are isolated, and thus are unaware of incompatible changes.

Changes in Late Binding

You can modify files, folders, or both in Late Binding mode as if you are not using the plug-in. No automatic check-out occurs, but all changes are monitored in the background. Every modified file is decorated with a delta symbol. Additionally, the modified items display in the Pending Changes view.

How to Commit Changes in Late Binding

You can commit modified items from the Solution Explorer, Pending Changes view, or the Synchronizer view. When you commit changes and your context is incomplete, the plug-in prompts you to complete the context, for example, set a package or process.

To commit changes, do any of the following:

- From the Pending Changes view, select the file that you want to commit, and click the Commit button on the toolbar.
- From the Solution Explorer, right-click the file that you want to commit, and select CA Harvest SCM, Commit, or click the Commit button on the toolbar.
- From the Synchronizer view, right-click the file or the container (project or solution file) in the Solution Explorer, and select CA Harvest SCM, Synchronize. Resolve any conflicts and click the Commit button.

Changes are committed to the trunk or branch based on the placement option set in the context.

Always Commit on Latest Trunk

When you set the preference “Commit edited item on Latest Trunk (Late Binding Mode),” commit always creates versions on the latest repository trunk. For example, if a developer changes version 1 of a file, but the latest version of the file in the repository is 5, then, on commit a conflict occurs. The developer must resolve the conflict using the Synchronize view. After the developer resolves the conflict and commits the file, version 5.1.1 is created.

When you do not set the preference “Commit edited item on Latest Trunk (Late Binding Mode),” commit always creates versions on the parent trunk of the local copy. For example, if a developer changes version 1 of a file, but the latest version of the file in repository is 5, then, on commit a version 1.2.1 is created.

Note: The “Commit edited item on Latest Trunk" has no effect on the Trunk placement option.

Synchronize Changes

You can do a “diff” of local items and repository items before committing your changes to the CA Harvest SCM repository. A "diff" helps you to do the following:

- Resolve conflicts with repository changes in a concurrent development environment.
- Review changes before you commit them to the CA Harvest SCM repository.

To synchronize from the Solution Explorer, right-click the item in the Solution Explorer and select CA Harvest SCM, Synchronize (or click the Synchronize button on the Solution Explorer toolbar).

The “diff” results appear in the Synchronizer tree in a separate Synchronizer tool window.

How to Get Changes From the CA Harvest SCM Repository

You can update your local copy of a Visual Studio solution by getting the latest changes from the CA Harvest SCM repository. The changes that are fetched from the CA Harvest SCM Repository depend on the context settings as follows:

- If a package is set, the latest branch versions in the package and latest trunk version in the view are fetched.
- If a package is not set, the trunk version in the view is fetched.

The context must have at least one check-out process.

You can get changes from the repository in the following ways:

- From the Solution Explorer, right-click any item and select CA Harvest SCM, Get Latest Version (or click the Get Latest Version button on the Solution Explorer toolbar).
- From the Synchronizer view, right-click the file or the container (project or solution file) in the Solution Explorer, and select CA Harvest SCM, Synchronize. Right-click the file in the Synchronizer view and select Get Latest Version.

Your local copy of a Visual Studio solution is updated with the latest changes from the CA Harvest SCM repository.

View the Output Log

Descriptions of all Source Code Control (SCC)-related and CA Harvest SCM-related operations that occur during an IDE session are presented in the Harvest SCM Output log.

To view the output log

1. In the Visual Studio IDE, select View, Harvest SCM Output.

   Note: If the CA Harvest SCM toolbar is active, you can click the Harvest SCM Output button.

   The log reports the following types of messages:

   - Error
   - Warning
   - Information

2. (Optional) Narrow the output by selecting the type of log to ignore.

   The log shows only messages of the type that you selected.

3. (Optional) Perform other actions by clicking any of the following toolbar buttons:

   Clear All

   Clears all the logs at once. All the counters are set to zero.

   Detail

   Opens the log file for the particular session.

4. (Optional) Right-click a particular log and select one of the following:

   Copy to clipboard

   Copies the details of the selected log to the system clipboard.

   Properties

   Provides extended information about the log (if available).

Online and Late Binding Working Modes

The plug-in works in the following different modes:

- **Online**—Always connects to a CA Harvest SCM broker. When you attempt to edit a file, the plug-in checks out the file. Checking out the file synchronizes the CA Harvest SCM repository with your local workspace. Then, you have visibility to what other users are working on by seeing the files that are checked out or reserved.

  Online mode does not support all refactoring operations (rename, move, and remove). If you want to use these processes from the Visual Studio IDE, use Late Binding mode.

- **Late Binding**—This mode means that you are not always connected to a CA Harvest SCM broker. If you want to edit a file, you may need to check out the file because the CA Harvest SCM repository may not be synchronized with your local workspace.

  If you are performing an operation in this mode that requires a server transaction (CA Harvest SCM server), the operation fails with a message informing you that the connection to the CA Harvest SCM broker is not available. In that case, connect to the CA Harvest SCM broker and perform the operation again. You do not need to switch to different mode.

  Late Binding mode supports all the refactoring operations.

Verify the Current Work Mode

You can verify the current work mode in the following ways:

To verify the current work mode from the Solution Explorer, right-click a file and see if the CA Harvest SCM, Work in Late Binding Mode option indicates the following:

- If selected, you are working in Late Binding mode.
- If not selected, you are working in Online mode.

To verify the current work mode from the Solution Explorer toolbar, view the Work Mode button as follows:

- Online
- Late Binding

The status icon indicates the current working mode.

To verify the current work mode from the Preferences page

1. Select Tools, Options from the Visual Studio IDE.
2. Select and expand Source Control, and select Work Mode.

   The current work mode and a brief description about the mode appears.

Switch the Work Mode

You can switch the work mode in any of the following ways. Before you switch work modes, verify that your Visual Studio IDE has the plug-in as the active provider of Source Code Control.

- From the Solution Explorer, right-click a file in the Visual Studio IDE and select CA Harvest SCM, Work in Late Binding Mode from the shortcut menu.
- From the Solution Explorer toolbar, click either of the Work Mode buttons (Online or Late Binding).
- From the Visual Studio IDE Preferences page, select Tools, Options, expand Source Control, and select Work Mode. Select the mode button that is not already selected, and click Apply.

  Note: Use this approach if there is no solution open or the solution is not managed.

The mode switches from the current working mode to the other mode. For example, Online mode switches to Late Binding mode, and conversely.

View a Harweb Form

Forms display in the Forms view through Harweb. The fields in a form are specific to the type of form loaded and vary by form type.

To view a Harweb forms

1. Open an IDE instance and select View, SCM Explorer.
2. Verify that Harweb is running on the broker you want to use, and connect to that broker.
3. Expand the Broker, Project, State, Packages nodes.

   A list of packages appears.
4. Select one package and expand it.

   A form node appears.
5. Double-click the form node.

   The right window shows the Harweb Login page.
6. Enter your user credentials and log in to Harweb.

   The form appears in the Form view. When you view additional forms in the same session, the forms appear in new Form tab pages.

Chapter 4: Managing Changes in the Repository and Workspace

This section contains the following topics:

Refactoring Limitations in Online Mode

The plug-in does not support the following refactoring operations while working in online mode:

- Move item
- Move path
- Remove item
- Remove path
- Rename item
- Rename path

Switch to Late Binding mode (which is the default preference) to perform these operations. When you perform these refactoring operations in Online mode, unexpected results can occur.

Integrity

In Visual Studio, a solution file keeps information about all projects in it. Every project file keeps information about all files and folders in it. Solution and project files are the containers. At the same time, the actual files and folders are physically present in the file system.

Typically, when you perform any refactoring operation (for example, rename a file in a project from the Visual Studio IDE), the container (solution or project) is also modified. So when committing the changes to repository, both the changes must be committed together. Select all related changes together to commit; otherwise, the integrity of the changes can be lost and you may encounter unexpected behavior. Integrity checks help ensure that the information about files and folders (name and location) in solution and project files are synchronized with the corresponding file system information.

Move an Item

In Visual Studio, you can move files in the local workspace and their corresponding items in the CA Harvest SCM repository.

Note: A move item process is required. If the CA Harvest SCM administrator has not defined a move item process and granted you access to run it, you cannot use Visual Studio to move an item in the CA Harvest SCM repository. In addition, move item is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To move an item

1. In the Solution Explorer, drag the file that you want to move, and drop it in the new location.

   The statuses of the file that is moved and its associated project file are updated to delta symbols, which indicate that implicit changes occurred in the project.

2. Right-click the project file and select CA Harvest SCM, Commit from the shortcut menu.

   The item is moved using the move item process that is set for your context.

Move a Path

In Visual Studio, you can move folders in the local workspace and their corresponding paths will be moved in the CA Harvest SCM repository.

Note: A move path process is required. If the CA Harvest SCM administrator has not defined a move path process and granted you access to run it, you cannot use Visual Studio to move a path in the CA Harvest SCM repository. In addition, move path is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To move a path

1. In the Solution Explorer, drag the folder that you want to move, and drop it in the new location.

   The status of the project file is updated to a delta symbol indicating that a change occurred in the project.

2. Right-click the project file and select CA Harvest SCM, Commit from the shortcut menu.

   The path and any subitems are moved to the new location using the move path process that is set for your context.

Remove an Item

The remove item process lets you logically delete selected items from a view in the CA Harvest SCM repository. When you remove an item from a view, the item is not deleted; the remove item process creates a version tagged as removed (D). This version has attributes like other versions; you can view this version in the version list of the package in the Harvest SCM Explorer view or in CA Harvest SCM Workbench, but not in the item view.

Note: A remove item process is required. If the CA Harvest SCM administrator has not defined a remove item process and granted you access to run it, you cannot use Visual Studio to remove an item in the CA Harvest SCM repository. In addition, remove item is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To remove an item

1. In the Solution Explorer, right-click the file that you want to delete or remove, and select Delete or Remove from the shortcut menu.

   The status of the project file or solution file is updated to a delta symbol, which indicates that a change occurred in the project or solution.

2. Right-click the project or solution file and select CA Harvest SCM, Commit from the shortcut menu.

   The item is removed and a D-tagged version is created for the removed item in CA Harvest SCM.

Remove a Path

The remove path process lets you logically delete selected paths from a view in the CA Harvest SCM repository. When you remove a path from a view, the path is not deleted; the remove path process creates a version tagged as removed (D). This version has attributes like other versions; you can view this version in the version list of the package in the Harvest SCM Explorer view or in the CA Harvest SCM Workbench, but not in the item view.

Note: A remove path process is required. If the CA Harvest SCM administrator has not defined a remove path process and granted you access to run it, you cannot use Visual Studio to remove a path in the CA Harvest SCM repository. In addition, remove path is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To remove a path

1. In the Solution Explorer, right-click the path that you want to remove, and select Remove from the shortcut menu.

   The status of the project or solution file is updated to a delta symbol, which indicates that a change occurred in the project.

2. Right-click the project or solution file and select CA Harvest SCM, Commit from the shortcut menu.

   The path and its subitems are removed from the IDE. D-tagged versions are created for the removed path and its subitems in CA Harvest SCM.

Rename an Item

In Visual Studio, you can rename a file in the local workspace and the corresponding item will be renamed in the CA Harvest SCM repository.

Note: A rename item process is required. If the CA Harvest SCM administrator has not defined a rename item process and granted you access to run it, you cannot use Visual Studio to rename a file in the CA Harvest SCM repository. In addition, rename item is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To rename an item in the repository

1. In the Solution Explorer, right-click the file that you want to rename, and select Rename from the shortcut menu.

   You can edit the file name.

2. Modify the name, and press Enter.

   The statuses of the file that is renamed and its associated project or solution file are updated to delta symbols, which indicate that changes occurred in the project or solution.

3. Right-click the project or solution file and select CA Harvest SCM, Commit from the shortcut menu.

   The item is renamed using the rename item process that is set for your context.

Rename a Path

In Visual Studio, you can rename a path in the local workspace and the corresponding path will be renamed in the CA Harvest SCM repository.

Note: A rename path process is required. If the CA Harvest SCM administrator has not defined a rename path process and granted you access to run it, you cannot use Visual Studio to rename a path in the CA Harvest SCM repository. In addition, rename path is not supported for Online mode. For more information about creating processes, see the *Administrator Guide*.

To rename a path in the repository

1. In the Solution Explorer, right-click the path that you want to rename, and select Rename from the shortcut menu.

   You can edit the path name.

2. Modify the name, and press Enter.

   The status of the associated project or solution file is updated to a delta symbol, which indicates that a change occurred in the project or solution.

3. Right-click the project or solution file and select CA Harvest SCM, Commit from the shortcut menu.

   The path is renamed using the rename item process that is set for your context.

Index