



ENOVIA DesignSync

DesignSync for Visual User's Guide

3DEXPERIENCE 2022

©2022 Dassault Systèmes. All rights reserved. 3DEXPERIENCE®, the Compass icon, the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA, NETVIBES, IFWE and 3DEXCITE are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the United States and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

**DASSAULT
SYSTEMES**

Table Of Contents

| | |
|---|----|
| Overview | 1 |
| About the DesignSync Visual Studio Integration | 1 |
| ENOVIA Synchronicity DesignSync® Data Manager for DSVS Capability | 1 |
| Using ENOVIA Synchronicity DesignSync for Visual Studio User's Guide Documentation | 1 |
| Before Reading this Guide | 2 |
| Related Topics | 2 |
| Getting Started with the DesignSync Visual Studio Integration | 2 |
| Methodologies for Working with Objects in DSVS | 2 |
| Locking Model | 3 |
| Merging Model | 3 |
| Using Modules with the DesignSync Visual Studio Integration | 3 |
| Using DSVS with SCC Compliant Applications | 4 |
| Installing the SCC plug-in | 4 |
| Setting up the SCC plug-in | 4 |
| Working Offline in Disconnected Mode | 4 |
| Source Code Control Icons | 5 |
| Using | 7 |
| Get Latest Version | 7 |
| Get Options | 7 |
| The Get Dialog Box | 8 |
| Check Out for Edit | 12 |
| The Check Out Dialog Box | 13 |

| | |
|---|----|
| Check In | 17 |
| The Check In Dialog Box | 18 |
| Undo Checkout..... | 22 |
| The Undo Checkout Dialog Box..... | 23 |
| History Options | 24 |
| The History Options Dialog Box | 24 |
| Show Module Status..... | 29 |
| The Show Module Status Dialog Box..... | 29 |
| Refresh Status..... | 31 |
| Share | 32 |
| Compare..... | 32 |
| The Compare Dialog Box..... | 32 |
| Properties | 35 |
| Launch SyncAdmin..... | 37 |
| Moving an Object..... | 38 |
| Deleting Objects | 39 |
| Visual Studio 2005 and other SCC compliant applications | 39 |
| Visual Studio 2003 | 41 |
| Tag Options | 41 |
| The Tag Dialog Box | 42 |
| Make Branch Options | 46 |
| The Make Branch Dialog Box | 46 |
| Exclude from Source Control | 48 |

| | |
|---|----|
| Setting up a Workspace | 51 |
| Setting up a new Visual Studio Project with DesignSync..... | 51 |
| Creating a New Visual Studio Project and Associating it with DesignSync | 51 |
| Adding a Module | 52 |
| Check In Comment | 54 |
| Set Vault Association | 54 |
| Confirm Source Control Association | 56 |
| Associating a Visual Studio Project with DesignSync..... | 57 |
| Associating a Visual Studio Project with DesignSync | 57 |
| Adding a Module | 58 |
| Selecting an Existing Module | 60 |
| Set Vault Association | 62 |
| Confirm Source Control Association | 63 |
| Associating a DesignSync module or vault with Visual Studio..... | 64 |
| Associating an Existing DesignSync Module or Vault with Visual Studio | 64 |
| Selecting an Existing Module | 66 |
| Set Vault Association | 68 |
| Confirm Source Control Association | 70 |
| Configuring..... | 73 |
| Displaying the Source Control Toolbar | 73 |
| Changing Source Control | 75 |
| The Change Source Control Dialog Box | 77 |
| DSVS Options | 78 |

Troubleshooting 81

- About DSVS 81
- DSVS log files..... 81
- Could Not Find Key Error..... 82
- Reset DesignSync Add-in..... 82
- Unable to specify options or comments when running DesignSync commands 82

Additional Information..... 85

- Module Context Field..... 85
- Select Vault URL Browser 85
- Command Buttons 85

Getting Assistance 87

- About DSVS 87
- Using Help..... 87
- Getting a Printable Version of Help..... 88
- Contacting ENOVA..... 88

Index 91

Overview

About the DesignSync Visual Studio Integration

ENOVIA Synchronicity DesignSync® Data Manager for Visual Studio Integration (DSVS) is an SCC client plug-in that integrates DesignSync with Microsoft Visual Studio (MSVS) and other applications that use an SCC compliant plug-in interface. The DSVS Integration provides DesignSync revision control functions from the SCC compliant GUI directly, without opening the DesignSync client application.

Note: This help features screen captures and procedures from the Visual Studio interface. Other SCC compliant applications function similarly, but not identically. For more information, see **Using DSVS with SCC Compliant Applications**.

ENOVIA Synchronicity DesignSync® Data Manager for DSVS Capability

DSVS provides the major revision control functionality available in DesignSync, including:

- Check In
- Checkout for Edit
- Compare
- Get
- Tag

Using ENOVIA Synchronicity DesignSync for Visual Studio User's Guide Documentation

This guide is single-sourced in HTML and generated to multiple locations.

- Integrated help - When you press F1 within the DSVS application or click on the Help button on DesignSync dialog boxes, the DSVS User's Guide help appropriate for the location or dialog opens in your default Web browser.
- DesignSync Documentation - available from the **Dassault Systems** product group in the Windows **Start** menu or on UNIX, by pointing your web browser to `$SYNC_DIR/share/content/doc/index.html`

Note: References from the *ENOVIA Synchronicity DesignSync for Visual Studio User's Guide* to the *ENOVIA Synchronicity Command Reference* guide always link to the ALL version of the guide, which contain information about all working methodologies for DesignSync. For more information about the available working methodologies, see ENOVIA Synchronicity Command Reference.

Before Reading this Guide

You might need to refer to the following guides if you are learning to use the ENOVIA Synchronicity DesignSync for Visual Studio product.

| | |
|--|---|
| <i>ENOVIA Synchronicity DesignSync User's Guide</i> | Describes the concepts and workflow for DesignSync in detail. |
| <i>ENOVIA Synchronicity DesignSync Administrator's Guide</i> | Describes the customizations available to optimize performance and usability and allows you to enable DSVS. |

Related Topics

Getting Started with the DesignSync Visual Studio Integration

Methodologies for Working with Objects in DSVS

About DSVS

Getting Started with the DesignSync Visual Studio Integration

To use the DSVS Integration:

1. Install the DesignSync SCC Plug-in as described in the *ENOVIA Synchronicity DesignSync Data Manager DesignSync Installation*. For information on accessing the Installation document, see Release Information.
2. Associate the Visual Studio project with a DesignSync vault or module:
 - For a new project: Creating a New Visual Studio Project and Associating it with DesignSync.
 - For an existing Visual Studio project that has never been checked into DesignSync: Associating a Visual Studio Project with DesignSync.
 - For an existing Visual Studio project that has been checked into DesignSync: Associating an Existing DesignSync vault with Visual Studio.
3. Perform Design Management operations as needed, such as:
 - Check In
 - Checkout for Edit
 - Compare
 - Get
 - Tag

Methodologies for Working with Objects in DSVS

Visual Studio supports two working models for Visual Studio projects.

- Locking model
- Merging model

The SyncAdmin preference "Checkout read only when not locking" determines your DSVS working model. To use the locking model, the setting must be checked. To use the merge model, the setting must be unchecked.

Locking Model

When working in the locking model, which depends on the SyncAdmin preference "Checkout read only when not locked" being checked, you always perform a check out with a lock for the object(s) you plan to edit. DSVS locks the branch associated with the version you are checking out, prohibiting other team members from creating new versions on that branch. You, the holder of the lock, reserve the right to create the next version on that branch. Other team members can get the object in an unlocked state, but no one else should make changes while you hold the lock.

Note that in multibranch environments, each branch of an object can be independently locked and unlocked. Therefore, different team members can modify the same object on different branches even when using the locking work model.

When actively editing Visual Studio projects in this mode, you must check the associated files out of DesignSync with a lock. This makes the files writable in the workspace and locks them on the source server to the user who initiated the lock.

Merging Model

When working in the merging model, which depends on the SyncAdmin preference "Checkout read only when not locked" not being checked, you do not check out with a lock. This allows more than one person to fetch the same object with the intention of editing the object. The first person to check the object back in creates the next version. The other person must first merge the changes from this new Latest version into his or her local copy, manually resolve any merge conflicts, then check in the merged object.

Related Topics

Get

SyncAdmin Help: General Options

Using Modules with the DesignSync Visual Studio Integration

Modules are collections of managed objects that together make up a single entity. In Visual Studio, you can have a module at a solution level or at a project level. The module's base directory is set either at the solution level or at the project level. Modules are manipulated with the same command set as all other DesignSync objects. Module-specific operations are called out in the appropriate sections of the DSVS operations.

Note: You may have files that are part of a module but not part of a solution or project within the solution. These files are not available from within Visual Studio as part of the solution or project, but can be managed within a DesignSync client.

Using DSVS with SCC Compliant Applications

The DSVS SCC-compliant plug in allows you to use the DSVS integration with any application that supports the Microsoft SCC plug-in.

The functionality described in this help is fully enabled for SCC compliant applications, although the methods of executing DesignSync actions may vary between applications.

Installing the SCC plug-in

The SCC plug-in option is installed by selecting the Visual Studio 2005 Integration when installing the DesignSync client for Windows.

Setting up the SCC plug-in

The application documentation for your application should contain instructions for enabling the SCC functionality, displaying the SCC toolbar, and selecting DesignSync as your source control provider.

Important: Some applications perform automatic "silent" check-in operations, meaning they run the operations in the background and do not allow you to set any check-in options. An example of when this might happen is if the program checks in any structural changes immediately, such as adding or removing a file.

By default, the DSVS uses a default comment, "autocheckin" for automatic checkins. For information on setting the default comment using SyncAdmin, see SyncAdmin Help: Third-Party Integration Options. Instead of SyncAdmin, you may use a registry key. For more information on the registry key, see SysAdmin Help: Vendor Objects Registry Settings.

Working Offline in Disconnected Mode

Visual Studio provides the option to edit project files which disconnected from the DesignSync server, for example, if the developer is working remotely. DSVS supports merging offline changes back to the server on reconnect.

When you reconnect your project to the DesignSync server, the Visual Studio integration attempts to perform a checkin of the modified files. The default checkin does not attempt to merge changes with the server version and may fail. If the Checkin fails, run a manual Check Out for Edit with the Merge with Workspace option checked. Then perform the Check In.





Related Topics

Check In

Check Out for Edit

Source Code Control Icons

When your project is in source control, an additional icon appears next to the file name in the Solution Explorer. These icons provide information about the status of the files in DesignSync. The following table lists the most common icons.

| Source Control Icon | Description |
|---|---|
| No icon | A file that is not in source control or has an unknown source control status. |
|  | A file in source control checked out with a lock by the current user working in the lock model, or locally modified by a user working in the merge model. |
|  | A file in source control that is not currently locked for edit in the lock model or up-to-date in the merge model. |
|  | A file that is not currently on the source control server, but is pending addition to source control by the current user. (Visual Studio 2005 only) |
|  | A generated file created by Visual Studio that cannot be checked into source control, but can be generated at will from the version in the workspace. |

For a complete list of source control icons, see the MSDN library topic, "How To: Identify Source Control Item Status in Solution Explorer."

Using

Get Latest Version

When you are using a project, another user may check in new objects to the vault. To update your workspace with these latest files, use the menu choice **File > Source Control > Get Latest Version**. This updates your workspace with the Latest version of the files for the branch populated into the workspace.

If you Get Latest Version of a module, the hierarchical references are not traversed and updated. If you want to refresh the module hierarchy, use the Get command and specify the Recursive option.

Get Latest Version is a silent action. The results of the Get Latest Version display in the Visual Studio output window.

You can get the latest versions of all the files in a solution, all the files in a folder, or specific objects within a solution. If you select a solution or project name, you automatically get all the latest files for that solution or project.

The files retrieved with Get follow the "checkout read only when not locking" preference set in SyncAdmin.

Related Topics

Get

Checkout for Edit

Get Options

When you are using a project, you may need to update your local objects from the server, or load a specific version of the files into your workspace. To update your workspace with these latest files, use the menu choice **File > Source Control > Get...**

You can get all the objects in a solution, all the files in a folder, or specific objects within a solution. The results of the **Get** operation display in the Visual Studio output window.


The behavior of Get depends on the source control working model you use and the default checkout setting:

- Lock model behavior - If you are using the lock model, meaning the 'Check out read only when not locking" preference set with SyncAdmin, is checked, the Get operation refreshes the files locally, but does not lock the files. If you want to check out files for edit, use Checkout for Edit.

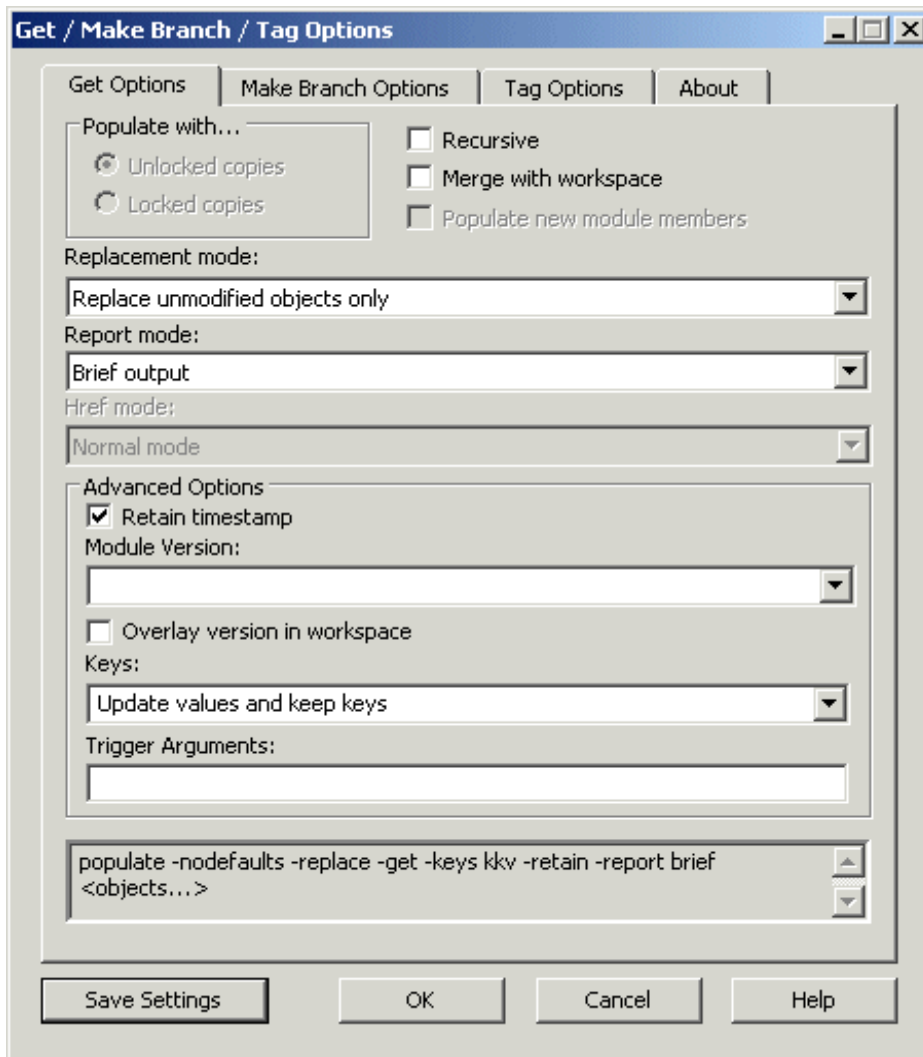
- Merge model behavior - If you are using the merge model, meaning the 'Check out read only when not locking" preference set with SyncAdmin, is not checked, the Get operation follows the SyncAdmin GUI Customization State Settings to determine whether the files are checked out with a lock, or without a lock when Get is run. For more information, see SyncAdmin Help: States Options.

The Get Dialog Box

By default, **Get** opens the Visual Studio **Get** page. This page provides a list of all the selected objects. If you click OK at this point, you perform a Get with the default options.

To specify different options, click the options button () to open the DesignSync **Get** form.

Click on the fields in the following illustration for information.



Populate with Unlocked copies

After the operation is over, keep an unlocked copy in your work area. This is currently the only supported state.

Note: This state overrides any default state set with SyncAdmin.

Populate with Locked copies

After the operation is over, keep a locked copy in your work area.

Recursive

Select this option (Modules only) to populate the entire module recursively from the top-level module through the module hierarchy, even if the selected object is part of a sub-module.

If recursive is not selected, the specific argument(s) to the populate command are the only objects populated. This means that in order to populate the entire module, you must specify the **Recursive** option.

This option is not active for Project based solutions, and for module based solutions when "Locked copies" is selected.

Merge with workspace

Select this option if you want to merge the Latest version of an object in the vault with a locally modified version.

If there are no conflicts, then the merge succeeds, leaving you the merged file in your work area. If there are conflicts, a warning message results. You must edit the merged file to resolve the conflicts before DesignSync allows you to check in the merged version. Conflicts are shown as follows:

```
<<<<<< versionID  
  
Lines from Latest version  
  
=====  
  
Lines from locally modified version  
  
>>>>>>
```

The conflicts are considered resolved when the file no longer contains any of the conflict delimiters (7 less-than, greater-than, or equal signs in the first column).

This option supports the merging work model where multiple team members can check out the Latest version of an object for editing. The first team member to check his changes in creates the next version; other team members merge their local changes with the new Latest version, and then check in the merged version.

Populate new module members

Select this option (Modules only) to populate new module members. This option is only active when the Recursive option is selected indicating that the entire module, and module hierarchy is being populated.

Replacement mode

The **Replacement mode** determines how the populate operation updates your work area with the data you are fetching. Specify one of the following update methods:

- **Do not replace any objects.** For DesignSync data, do not overwrite locally modified files. Do not remove files that do match the requested version.

For module data, preserve local modifications you made to module members, and leave intact any module members that are in your work area that are not in the requested module version.

- This mode causes the least disruption to your work area; however, it may require you to clean up resulting work area data.
- **Replace unmodified objects only.** This replacement mode does not apply to DesignSync data. DesignSync data will not be replaced.

For module data, update module members that have not been locally modified and that are part of the requested module version. Also remove any unmodified module members that are not part of the requested module version.

- This mode, which is the default behavior for module data, leaves intact any module members you have modified in your workspace
- **Force overwrite of modified objects.** For DesignSync data, overwrite locally modified files. Also remove managed objects that do not match the requested selector.

For module data, replace or remove module members, regardless of whether you have modified them locally or whether they are part of the requested module version.

- The intent of this mode is to make the workspace match the data being requested (subject to the **Filter** and **Exclude** fields), as closely as possible. Unmanaged data is never removed.

If there is a conflict with data fetched from another module, that other data is not removed. See DesignSync Data Manager User's Guide: Conflict Handling for details.

If populating a module overlaps with another module already in your workspace, data from that other module is not removed.

This mode is mutually exclusive with the **Merge with workspace** option.

The **Replacement mode** only applies to items that are not filtered out by the **Href filter**, **Filter** or **Exclude** options.

Report mode

For the **Report mode**, choose the level of information to be reported:

- **Brief output:** Brief output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Version of each module processed.
 - Creation message for any new hierarchical reference populated as a result of a recursive module get.
 - Removal message for any hierarchical reference. removed as part of a recursive module get.
 - Success/failure status.
- **Normal output:** In addition to the information reported in Brief:
 - Informational messages for objects that are successfully updated by the get operation.
 - Messages for objects excluded from the operation (due to exclusion filters or explicit exclusions).
 - Information about all fetched objects.
- **Verbose output:** In addition to the information reported in **Normal** mode:
 - Informational message for every object even if it is not updated, for example objects that are skipped because the version in the workspace is the current version.
 - For module data, also outputs information about all objects that are filtered.
- **Errors and Warnings only:** Errors and Warnings output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Success/failure/ status messages.

Href mode

The **Href mode** option lets you specify how hierarchical references should be evaluated in order to identify the versions of submodules to reference when populating a module recursively. This field is only available when operating on module data.

- **Normal mode:** Examines the href's selector. If the href's selector represents a *static* version (a numeric version, a version tag, or a list of such items), the populate fetches the submodule using the selector. In this case, the populate uses **Static mode** for referencing subsequent levels of hierarchy. If the href's selector does *not* represent a static version, the populate still fetches the submodule using the selector, but continues to use **Normal mode** for referencing subsequent levels of hierarchy. This is the default **Href mode**.
- **Static mode:** Populates the static version of the submodule that was recorded with the href at the time the parent module's version was created.
- **Dynamic mode:** Evaluates the selector associated with the href to identify the version of the submodule to populate.

The **Href mode** option is mutually exclusive with the option to fetch **Locked copies**.

Module Version/Version

Specify the version number or tag (or any selector or selector list) of the objects on which to operate. By default, this field is set to the folder's persistent selector list.

Important: If individual module members are selected, the version populated is the module version, not the module member versions.

Overlay version in workspace

Specify this option to overlay the version specified in the **Version** field onto the work area. Overlaying a version does not change the current version status as stored in the workspace metadata. The **Overlay version in workspace** option is often used in conjunction with the **Merge with workspace** option, to merge one branch onto another.

Related Topics

Get Latest Version

Checkout for Edit

Undo Checkout


Check Out for Edit

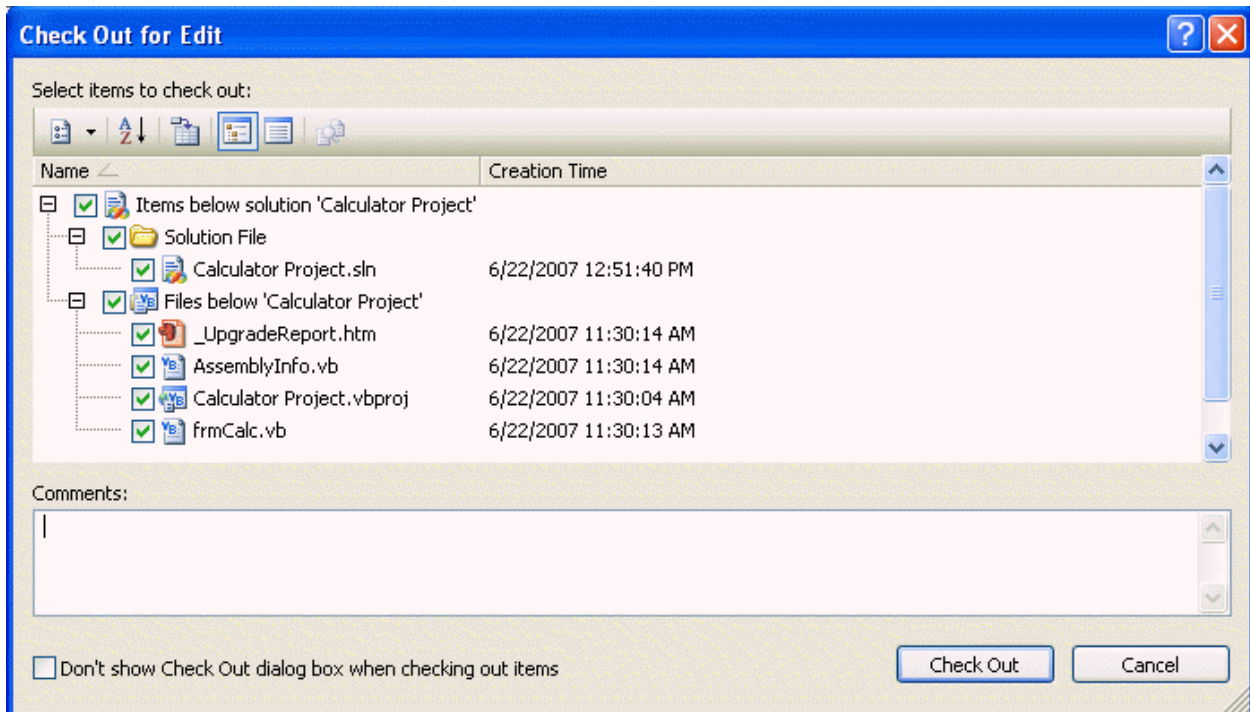
This operation checks out objects for users working in the Lock model, meaning that the user must check out the objects with a lock in order to edit it. For users working in the merge model, use Get Options to update your workspace.

When you want to edit Visual Studio projects, you must Check out the objects for edit.

To check your objects in, use the menu choice **File > Source Control > Check out for edit...**

Note: The menu is context sensitive, so it may provide specific check out options depending on the selected object(s), For instance, if you have a single file in the project selected, you have the option to check out only that file, or the entire solution.

By default, **Check Out** opens the Visual Studio **Check Out** page. This page provides a list of all the files being checked in. Click the options button () to open the DesignSync **Check Out** form.

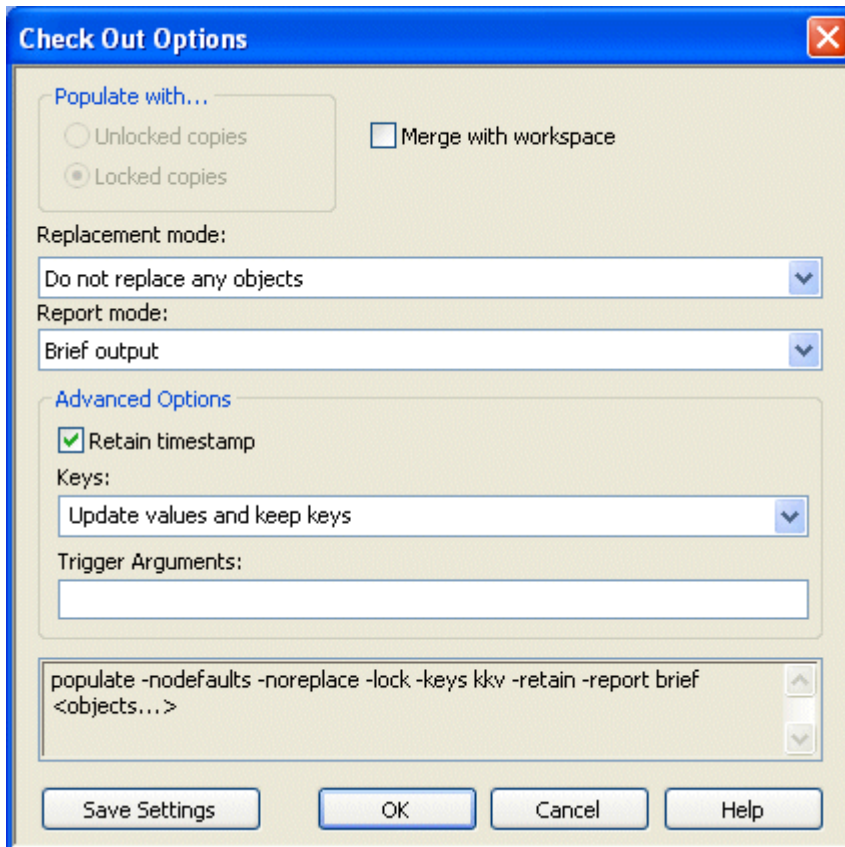


Notes:

- To bypass the Visual Studio **Check Out** form for subsequent check outs, Click the **Don't show Check Out dialog box when checking in items** checkbox.
- Comments entered into the **Check Out** Comments box are not retained in the source control system.

The Check Out Dialog Box

Click on the fields in the following illustration for information.



Populate with Unlocked copies

After the operation is over, keep an unlocked copy in your work area. This state is not supported for Checkout for Edit. If you want to populate with an unlocked copy, use the Get operation.

Populate with Locked copies

After the operation is over, keep a locked copy in your work area.

Note: This state overrides any default state set with SyncAdmin.

Merge with workspace

Select this option if you want to merge the Latest version of an object in the vault with a locally modified version or when you're reconnecting to the SyncServer after working disconnected.

If there are no conflicts, then the merge succeeds, leaving you the merged file in your work area. If there are conflicts, a warning message results. You must edit the merged file to resolve the conflicts before DesignSync allows you to check in the merged version. Conflicts are shown as follows:

```
<<<<<< versionID  
  
Lines from Latest version  
  
=====  
  
Lines from locally modified version  
  
>>>>>>
```

The conflicts are considered resolved when the file no longer contains any of the conflict delimiters (7 less-than, greater-than, or equal signs in the first column).

This option supports the merging work model where multiple team members can check out the Latest version of an object for editing. The first team member to check his changes in creates the next version; other team members merge their local changes with the new Latest version, and then check in the merged version.

Replacement mode

The **Replacement mode** determines how the populate operation updates your work area with the data you are fetching. Specify one of the following update methods:

- **Do not replace any objects.** For DesignSync data, do not overwrite locally modified files. Do not remove files that do match the requested version.

For module data, preserve local modifications you made to module members, and leave intact any module members that are in your work area that are not in the requested module version.

- This mode causes the least disruption to your work area; however, it may require you to clean up resulting work area data.
- **Replace unmodified objects only.** This replacement mode does not apply to DesignSync data. DesignSync data will not be replaced.

For module data, update module members that have not been locally modified and that are part of the requested module version. Also remove any unmodified module members that are not part of the requested module version.

- This mode, which is the default behavior for module data, leaves intact any module members you have modified in your workspace
- **Force overwrite of modified objects.** For DesignSync data, overwrite locally modified files. Also remove managed objects that do not match the requested selector.

For module data, replace or remove module members, regardless of whether you have modified them locally or whether they are part of the requested module version.

- The intent of this mode is to make the workspace match the data being requested (subject to the **Filter** and **Exclude** fields), as closely as possible. Unmanaged data is never removed.

If there is a conflict with data fetched from another module, that other data is not removed. See DesignSync Data Manager User's Guide: Conflict Handling for details.

If populating a module overlaps with another module already in your workspace, data from that other module is not removed.

This mode is mutually exclusive with the **Merge with workspace** option.

Report mode

- For the **Report mode**, choose the level of information to be reported:
 - - **Brief output:** Brief output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Version of each module processed.
 - Creation message for any new hierarchical reference populated as a result of a recursive module get.
 - Removal message for any hierarchical reference removed as part of a recursive module get.
 - Success/failure status.
 - - **Normal output:** In addition to the information reported in Brief:
 - Informational messages for objects that are successfully updated by the get operation.
 - Messages for objects excluded from the operation (due to exclusion filters or explicit exclusions).
 - Information about all fetched objects.
 - - **Verbose output:** In addition to the information reported in **Normal** mode:
 - Informational message for every object even if it is not updated, for example objects that are skipped because the version in the workspace is the current version.

- For module data, also outputs information about all objects that are filtered.
- **Errors and Warnings only:** Errors and Warnings output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Success/failure/ status messages.

Related Topics

Get Latest Version

Get

Check In

Undo Checkout

Check In

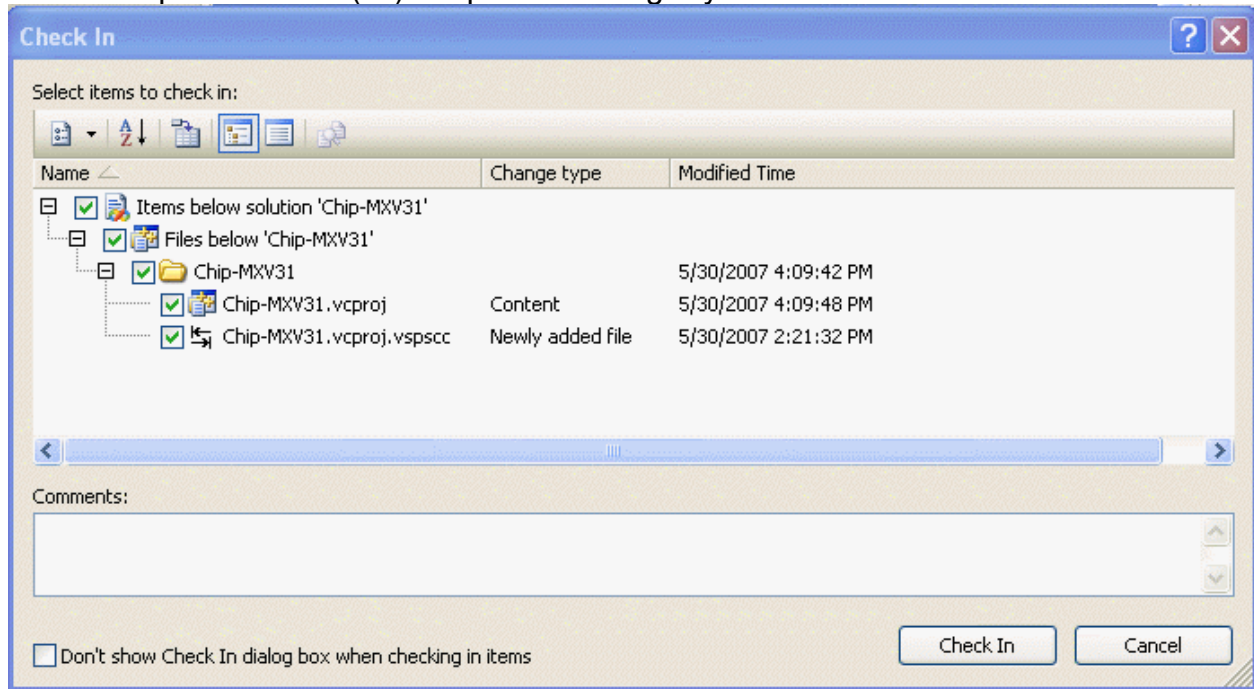
To preserve edits you've made, you must check your objects into source control. To check your objects in, use the menu choice **File > Source Control > Check In...**

Note: The menu is context sensitive, so it may provide specific check in options depending on the selected object(s). For instance, if you have a single file in the project selected, you have the option to check in only that file, or the entire solution.

By default, **Check In** opens the Visual Studio **Check In** page. This page provides a list of all the files being checked in.

Note: In order to add new files to a project or solution, you should select the files specifically. If you select the project or solution, new files are not automatically checked in.

Click the options button () to open the DesignSync **Check In** form.

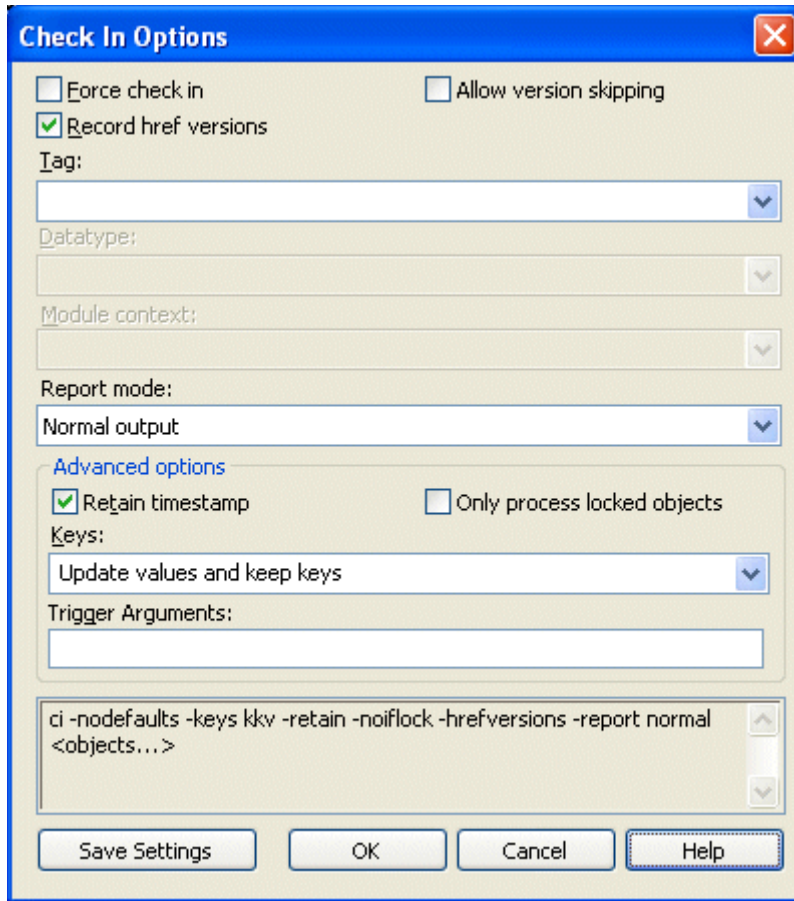


Notes:

- Comments entered here are not shown on the DesignSync **Check In** form, but are submitted to DesignSync and stored in the object version history.
- Comments are limited to a maximum of 1024 characters.
- To bypass the Visual Studio **Check In** form for subsequent checkins, Click the **Don't show Check In dialog box when checking in items** checkbox.

The Check In Dialog Box

Click on the fields in the following illustration for information.



Force check in

If you check out a file, make no changes to it, and then attempt to check it in, DesignSync informs you that it will not check the file in. If you want to check the file in anyway, you must select this check box. The file will be checked in and a new version created that is identical to the version already in the vault.

Note that you must have a local copy of the file in your working directory for a new version to be created. If the object doesn't exist, you cannot check in a new version.

Tip: Using this option allows you to keep version numbers synchronized.

Record href versions

Specifies whether to update the static hierarchical references associated with the module.

When this option is selected and a module is checked in (either an entire module or any of its contents), DesignSync captures the currently populated versions of the module's

hierarchically referenced sub-modules, and records those as part of the next module version, updating the static hierarchical references. (Default)

When this option is not selected and a module is checked in, the module members are checked in, but the hierarchical references are ignored (not updated). This is particularly useful if you have out-of-date submodules, or submodule changes that are not ready to be checked in.

If the **Recurse into folders** option is selected, and the check-in is operating in a **Module context**, then this option cannot be de-selected.

Allow version skipping

By default, this option does not appear in the **Check In** dialog box. For the **Allow version skipping** option to appear, the option must be set in the Sync Administrator application. See SyncAdmin Help: Command Options for more information.

This option allows you to create a new version, even if the new version is not derived from the Latest version. This happens if your modifications were to a non-Latest version. The new version skips over changes made in intermediate versions, which is why the option is hidden by default.

For module data, the objects that are being considered for checkin may still be checked in if there is a later module version that has affected those objects. For example, suppose you have module version 1.4 of Chip, which contains version 1.2 of file.txt. You modify your local copy of file.txt. Meanwhile, a later version of the module Chip has been checked in, containing version 1.3 of file.txt. Opting to **Allow version skipping** will include your file.txt modifications in the new version of the module that is created, skipping over the changes in version 1.3 of file.txt. However, if a module member file was renamed or removed in intermediate module versions, then those structural changes are also skipped. **Allow version skipping** also applies to changes to hierarchical references. If the checkin is capturing a new static version of a submodule for an href, and an intermediate module version also changed the href, **Allow version skipping** will cause the href change in the workspace to be applied.

Tag

Tags the file version or module version on the server with the specified tag name.

For module objects, all objects are evaluated before the checkin begins. If the objects cannot be tagged, for example if the user does not have access to add a tag or the tag exists and is immutable, the entire checkin fails.

For other DesignSync objects, if the user does not have the proper access permissions to add a tag, the object is checked in without a tag.

For more information on the access control for the tag command, see ENOVIA Synchronicity Access Control Guide: Access Controls for Tagging. For more information on branch and version tags, see Tagging Versions and Branches.

Notes:

- If both a tag and a comment are specified for a module version or branch checkin, the comment is also used as the tag comment.
- You cannot tag modules stored on DesignSync server versions prior to 5.1.

Datatype

Specifies the data type for any module object or any new vault object being checked in. If the data type field does not allow you to enter any information, DSVS uses auto-detect to determine the data type.

- **Auto-detect** uses a built-in algorithm to determine whether the object contains only ASCII text or a binary file. (Default)
- **ASCII** creates the new object with a vault data type of ascii.
- **Binary** creates the new object with a vault data type of binary. Binary objects cannot be merged, they can only be replaced. ZIP vaults are always checked in using binary mode, regardless of whether the vault's data type is designated as ascii.

Note: To change the data type of an existing object, use the url setprop command. For more information, see ENOVIA Synchronicity Command Reference: url setprop.

Report mode

For the **Report mode**, choose the level of information to be reported:

- **Brief output:** Brief output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Informational messages concerning the level of the hierarchy being processed.
 - Success/failure status.
- **Normal output:** In addition to the information reported in Brief mode, output normal mode reports:
 - Informational messages for updated objects.
 - Information about all objects processed.
- **Verbose output:** In addition to the information reported in **Normal** mode:
 - Informational message for every object examined but not updated.

- Information about all filtered objects.
- **Errors and Warnings only:** Errors and Warnings output mode reports the following information:
 - Failure messages.
 - Warning messages.
 - Success/failure status messages.

Only process locked objects

Specifies whether to check in all modified objects in the workspace or only targeted files. Changes that are targeted (or locked) are:

- Locked DesignSync vault files or module members.
- Objects that have been added to a module.
- Module members that have been renamed or removed since the last module checkin.


Related Topics

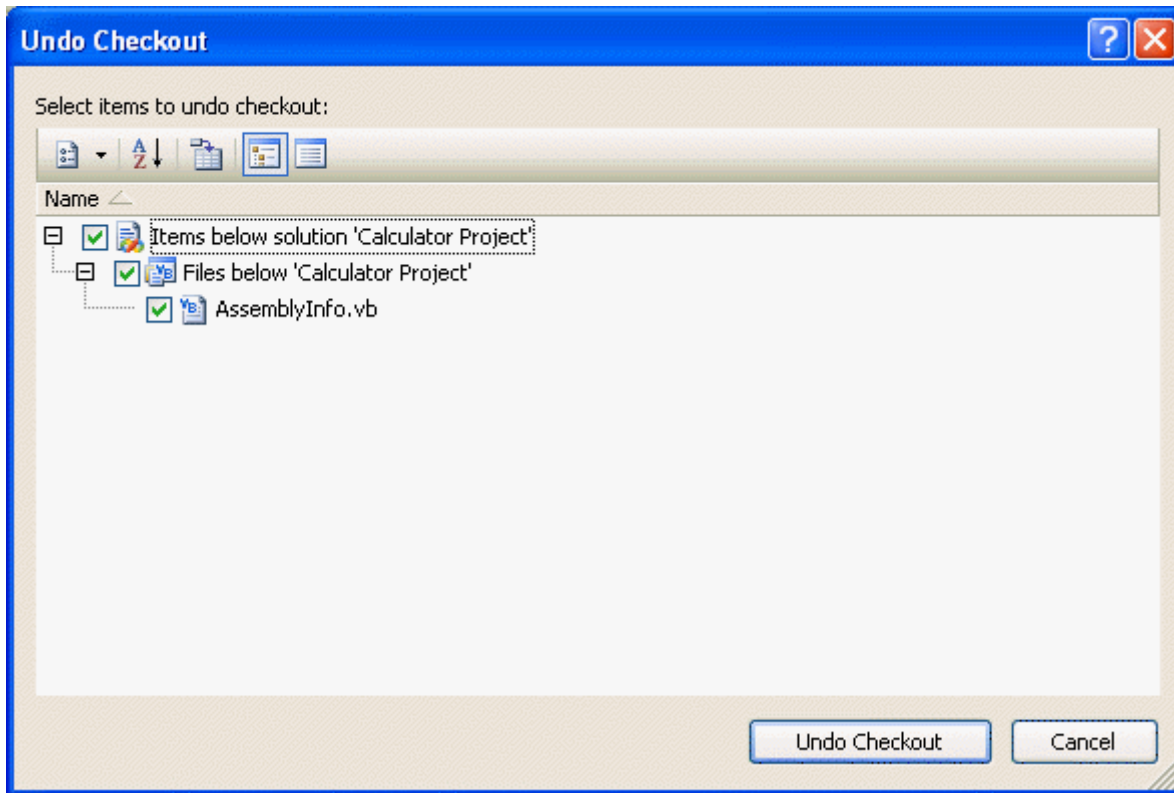
Checkout for Edit

Undo Checkout

If you check out an object for editing (with a lock) and then change your mind about creating a new version, you can undo (cancel) the checkout with the **Undo Checkout** dialog box. When you cancel a checkout, an unlocked copy of the object remains in your work area. You can also request a different object state if you use the options to the **Undo Checkout** command. If you have made changes to the object prior to canceling the checkout, you cannot change the object state, thereby overwriting your local modifications, unless you specify the **Force Overwrite of Local Modifications** option (**cancel -force**).

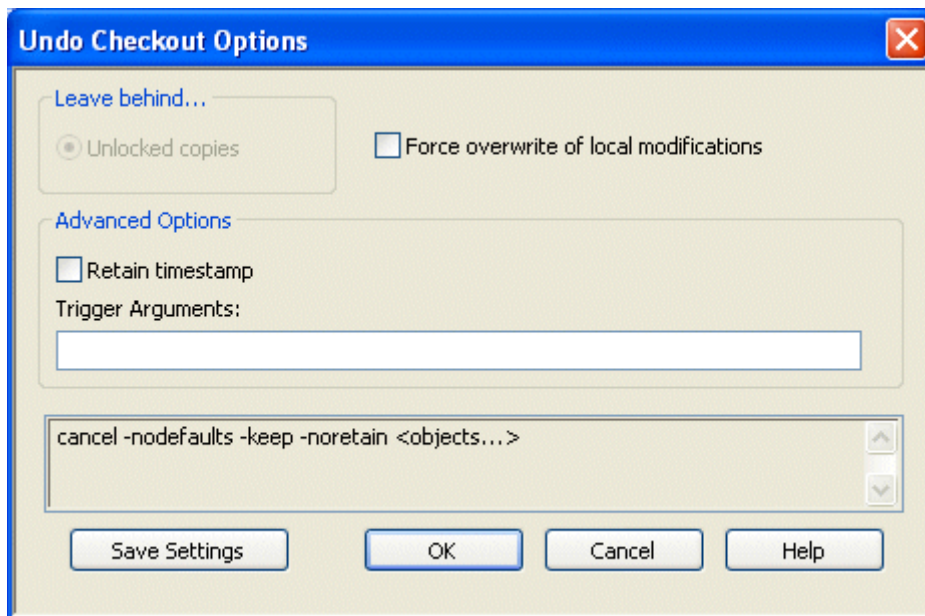
Note: This command only cancels checkouts performed by you.

To check your files in, use the menu choice **File > Source Control > Undo Checkout...** This dialog provides a list of all the files being unchecked out. Click the options button () to open the DesignSync **Undo Checkout** form.



The Undo Checkout Dialog Box

Click on the fields in the following illustration for information.



Leave behind Unlocked copies

After the operation is over, keep an unlocked copy in your work area. This is the default unless your project leader has defined a default fetch state.

Because you have relinquished any lock you may have had on the file, someone else can check the file out from the vault with lock to modify it.

The SyncAdmin setting **Check out read only when not locking** determines whether these files are read/write or read-only.

Force overwrite of local Modifications

Select this option if you want to overwrite locally modified files with the vault version and remove managed objects that do not match the requested selector.

The intent of this mode is to make the workspace match the data being requested as closely as possible. Unmanaged data is never removed.

If this option is not specified, DesignSync updates any unmodified files.

Related Topics

ENOVIA Synchronicity Command Reference: cancel

Check In

Checkout for Edit

History Options

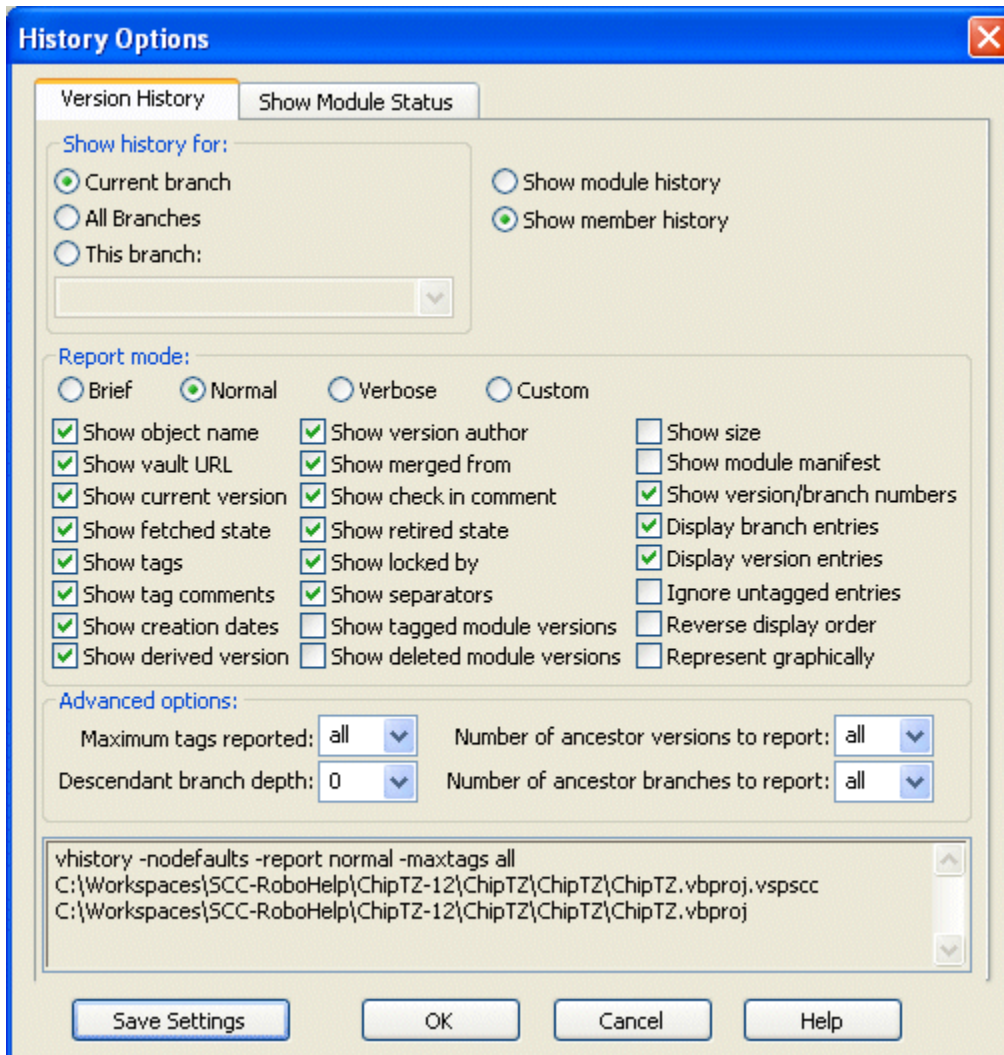
The Version History report displays version history for managed objects.

File >Source Control >View History opens the **Version History** dialog box. This report is available for any object in the Visual Studio project under revision control.

The **Version History** operation results display in the Output window.

The History Options Dialog Box

Click on the fields in the following illustration for information.



Show history for

The default setting shows the history of the **Current branch** (the branch of the object in the workspace). You can also show the history for **All branches** of the object, or a specific branch (**This branch**).

To select the history of a particular branch, select "Show history for This branch". The field below becomes active and you can type the branch name or click the pull-down menu to select a branch.

Note: The pull-down menu is only populated when you have a selected a single object.

Show Module History/Show Member History

When a module object is selected, this field allows you to choose whether the history is reported for the module, or the specified module members.

Report mode

Specify how the information is reported. The choices **Brief**, **Normal**, and **Verbose** represent defined reports. Selecting one of these defined reports automatically enables all of the report options that comprise the selected mode. Report options not in the selected mode are automatically disabled. The **Normal** report mode is selected by default. Select the **Custom** report mode to specify your own combination of report options.

Show object name

Show the workspace path to the object, or to the vault URL.

Show vault URL

Show the vault URL associated with a workspace object.

Show current version

Show the version currently in the workspace.

Show fetched state

Show the fetched state in the workspace.

Show tags

Show branch and version tags. Immutable tags are shown with "(immutable)" appended.

Show tag comments

Show the comments associated with version and branch tags. This isn't applicable for Visual Studio data.

Show creation dates

Show a version's creation date.

Show derived version

Show the numerical parent version. This maintains the continuity between versions for merge and rollback operations.

Note: If a merge, skip, rollback or overlay operation occurs to create this version, the referenced version is shown as "Merged from" version.

Show version author

Show a version's author.

Show merged from

Show the version used to create the current version when the current version was created as the result of a rollback, merge, skip, or overlay operation requiring an alternate parent version.

Show check in comment

Show a version's check in comments, and any checkout comments. Checkout comments are only visible from the workspace in which the checkout occurred.

Show retired state

Show whether a branch is retired.

Show locked by

Show the lock owner of a locked branch and the "version -> upcoming version" information.

Show separators

Show separators between items and versions.

Show tagged module versions

Show module version that have tags, even if a module member being queried version has not been changed in that module version.

Show deleted module versions

Show module version that were purged or deleted.

Show size

Show the size of the object version in KB.

Note: Collections and module versions, both of which contain more than one object, display with a size of zero.

Show module manifest

For a module, show the manifest of changes in each version. For a module member, show only the changes to that member.

Show version/branch numbers

Show the version number for versions, and the branch number for branches. For branches, indicate whether any versions exist on the branch.

Display branch entries

Show information for branch objects.

Display version entries

Show information for version objects.

Ignore untagged entries

Do not show entries that have no tags.

Reverse display order

Show the versions/branches in reverse numeric order.

Represent graphically

Show a graphical representation of the version history, as a text graph.

Maximum tags reported

Sets the maximum number of tags shown for any object. You can select a value from the pull-down list, or type in a positive integer. By default, all tags are shown. This option is only available when Show tags is selected.

Number of ancestor versions to report

Sets the number of previous versions to report. By default, all versions on a branch are reported. You can select a value from the pull-down list, or type in a positive integer. Specifying the number of ancestor versions to report sets the Descendant branch depth value to 0. This option is only available when Show history for **Current branch** or **This branch** is selected.

Number of ancestor branches to report

Sets the number of previous branches back to report. By default, only versions on the specified branch are reported. You can select a value from the pull-down list, or type in

a positive integer. Specifying the number of ancestor branches to report sets the Descendant branch depth value to 0. This option is only available when Show history for **Current branch** or **This branch** is selected.

Descendant branch depth

The number of levels of descendant branches to report, from the starting branch. By default, the report is limited to the starting branch (a value of 0). You can select a value from the pull-down list, or type in a positive integer. Specifying a descendant branch depth sets the Number of ancestor versions to report and the Number of ancestor branches to report to **all**.

Related Topics

Properties

Show Module Status

Show Module Status

The Module Status report displays the status of the module including:

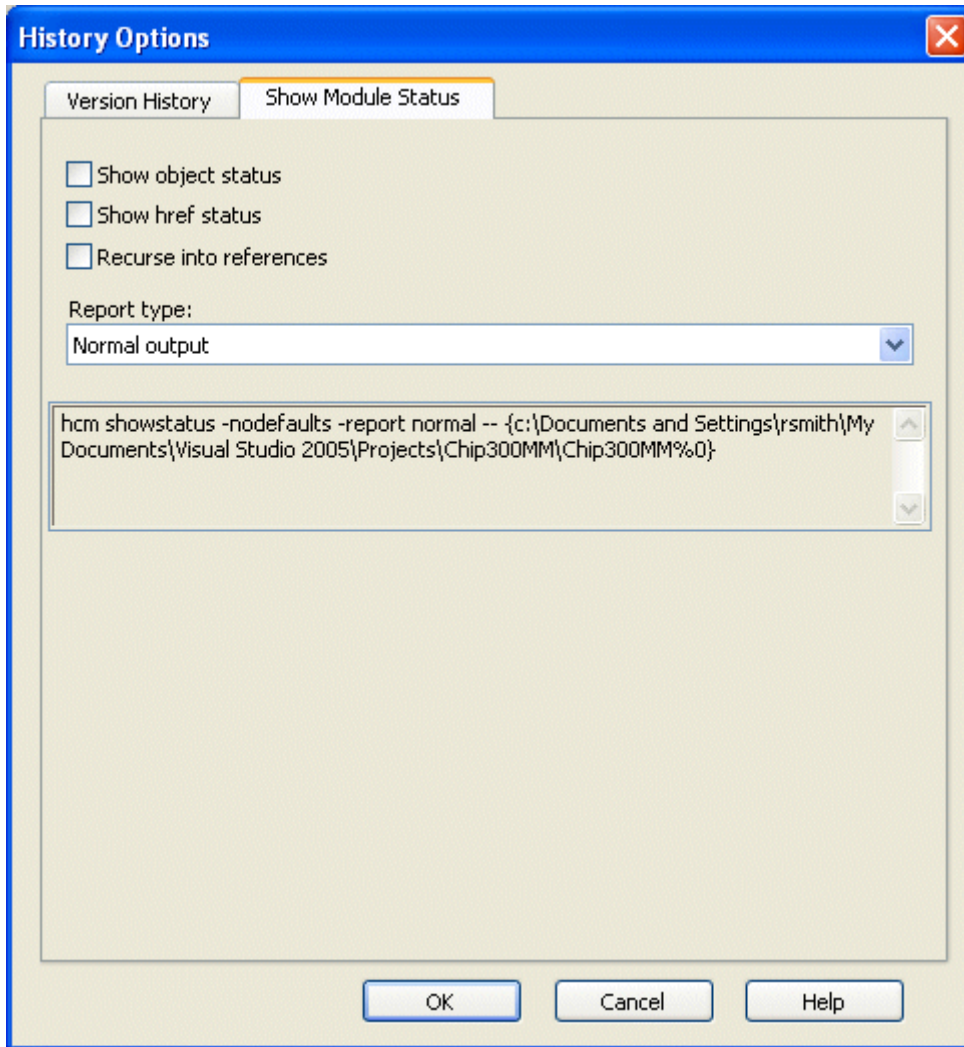
- The status of individual module members.
- The status of hierarchical references.
- The status of module members in referenced modules

File >Source Control >View History opens the **Version History** dialog box. From this dialog box, select the **Show Module Status tab**. This report is available for any module object in the Visual Studio project under revision control.

The **Show Module Status** operation results display in the Output window.

The Show Module Status Dialog Box

Click on the fields in the following illustration for information.



Show object status

Select this option to compare the status of each workspace object with the server module. When this option is not checked, the command returns only the status of the module and the status of its hierarchical references.

Show href status

Select this option to verify that the reported status of the hierarchical references is current. When this option is not checked, the hierarchical references are not verified.

Recurse into references

Select this option to displays the status for the specified module and all referenced modules. When this option is not checked, the report displays the status for the specified module only.

Report type

Select the type of report from the pull down list to be displayed on the Module Status tab. The default choice is Normal output. There are four report modes:

- **Brief output** – Displays a summary and lists hierarchical references that are out-of-date. Lists file status for files that are out-of-date. Also displays a table of conflicts if conflicts exist between the expected submodule and the actual submodule.
- **Normal output** – Displays the status of the hierarchical references and file status for the module. Displays a table of conflicts if conflicts exist between the expected submodule and the actual submodule.
- **Summary output** – Displays the target and base directory of the module, the status of each module, and the overall status of the module in the workspace. Also displays a table of conflicts if conflicts exist between the expected submodule and the actual submodule.
- **Verbose output** – Displays the status of the hierarchical reference and file status for the module. Also displays a table of conflicts if conflicts exist between the expected submodule and the actual submodule.

Related Topics

History Options

Properties

ENOVIA Synchronicity Command Reference: showmods

Refresh Status

Refresh Status updates the status of objects in the workspace. This is particularly useful if you are working on a collaborative project and want to check your workspace against the other work being done on the project.

This option is a silent option, taking no options. The results of the Refresh Status command are displayed in the options window and the status icons and other status related information is updated.

Related Topics

Get

Get Latest

Share

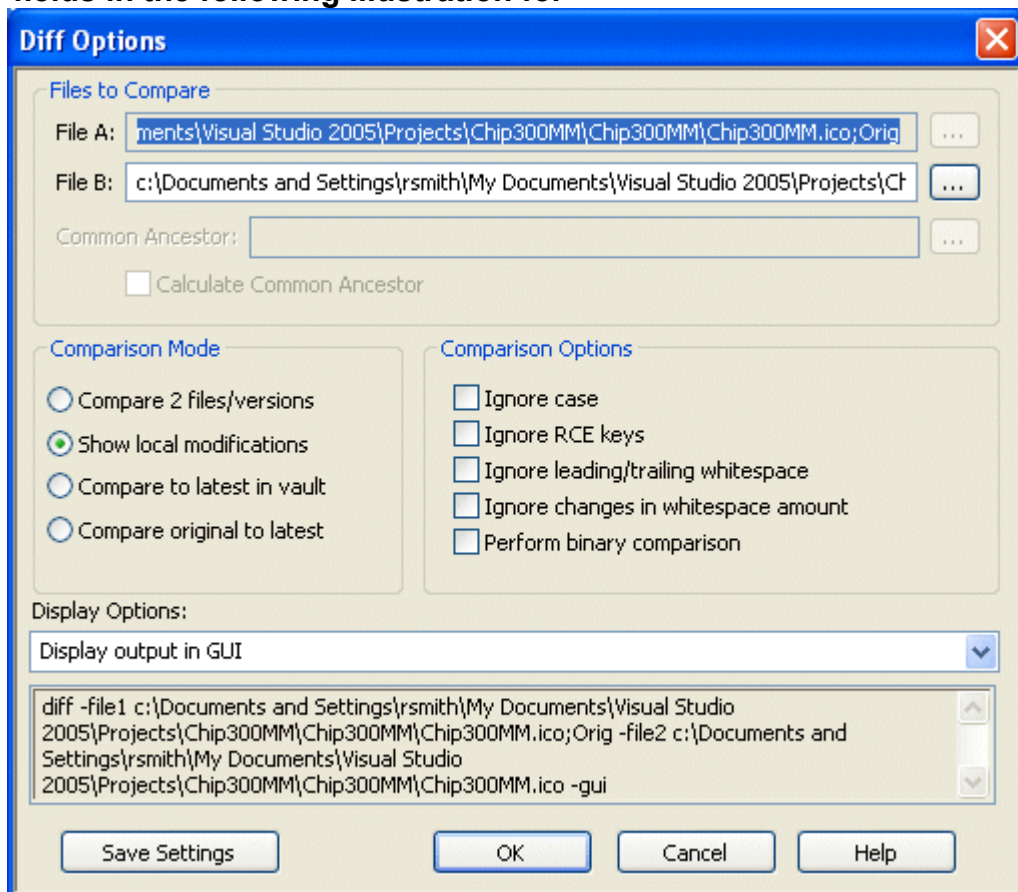
This button is provided as part of the source control toolbar, but is not applicable to DesignSync objects.

Compare

The diff operations provided in the **Source Control => Compare...** menu allow you to check the contents of a selected workspace object against the server or other workspace objects.

The Compare Dialog Box

Click on the fields in the following illustration for



information.

Files to Compare

Specify one or two files depending on the type of comparison you want to perform. If you choose to compare 2 files or versions in the Comparison Mode, enter the URLs of the objects to compare in the **File A** and **File B** fields.

Optionally, you may specify a common ancestor. If you specify a common ancestor, a 3-way diff is performed. For example, if you specify the following:

| | |
|------------------|--|
| File A | file:///home/aurora/projects/alu/sym12.dat;exper1 |
| File B | file:///home/aurora/projects/alu/sym12.dat;release17 |
| Com. Anc. | file:///home/aurora/projects/alu/sym12.dat;golden |

the **exper1** and **release17** versions of the given file are compared using the **golden** version as the common ancestor.

Instead of specifying a common ancestor, you can select **Calculate Common Ancestor**, and the common ancestor will be determined, using the object's branch and merge history. If no common ancestor can be determined, a 2-way diff is performed. For example, if you select **Calculate Common Ancestor** and specify the following:

| | |
|---------------|---|
| File A | file:///home/aurora/projects/alu/sym12.dat;1.12.1.5 |
| File B | file:///home/aurora/projects/alu/sym12.dat;1.17 |

a 3-way diff will be performed using version 1.12 as a common ancestor (unless a merge has created a more recent common ancestor).

Compare 2 Files

This mode compares the two selected objects. You can use this report to compare two files or two versions of the same file. Depending on the default settings, if two versions of a file are selected, this report may attempt to identify a common ancestor for a 3-way diff.

Show Local Modifications

This mode compares the selected object in your work area with the original version that you checked out. This report shows changes made in your work area since the object was checked out.

Compare to Latest in vault

This mode performs a 3-way diff comparing the selected object in your work area with the current version in the vault, using the original version in the vault as the common

ancestor. This report can be used to show the results that occur if a **checkout** and **merge (co -merge)** operation is performed, including any conflicts that occur.

Compare Original to Latest

This mode compares the original version of the selected object with the latest version in the vault. This report shows changes made to the vault by others since the object was checked out.

The default settings used by the common diff operations can be modified by selecting options and pressing the **Save Settings** button.

Comparison Options

Select from the following comparison options:

- **Use module versions:** Select this option to use module versions rather than module member versions when comparing two files/versions.
- **Ignore case:** Select this option for case-insensitive comparisons.
- **Ignore RCE keys:** Determines whether differences in RCE keyword values are ignored. RCE keywords are tokens (such as \$Revision: 1.4.1.6.1.11\$, \$Author: FYL\$, and \$Log: /advanced_diff.htm\$, and , and Revision: 1.4.1.6.1.11 Wed Sep 12 14:24:02 2018 GMT FYL , and updates for 2019x stuff, and , and Revision: 1.4.1.6.1.10 Thu Jun 16 10:45:29 2016 GMT FYL, and various updates for various guides , and , and Revision: 1.4.1.6.1.9 Thu Nov 13 03:57:51 2014 GMT mhopkins , and updates for external linking, and , and Revision: 1.4.1.6.1.8 Fri Nov 07 21:02:00 2014 GMT mhopkins , and changes to syncref links etc. , and , and Revision: 1.4.1.6.1.7 Tue Oct 15 22:01:19 2013 GMT mhopkins , and updated to new robohelp version. Updated cover page, and , and Revision: 1.4.1.6.1.6 Tue Jul 09 13:27:24 2013 GMT mhopkins , and checking in latest versions for software update. , and , and Revision: 1.4.1.6.1.5 Thu Jul 29 14:04:34 2010 GMT mhopkins , and Unable to run a diff to see what the differences in this version are, and Revision 1.4.1.6.1.4 Thu Nov 12 19:23:13 2010 GMT mhopkins , and Updated to RH8. class=Mono> Revision 1.4.1.6.1.3 Fri Sep 19 18:47:11 2008 GMT mhopkins class=Mono> SD 43791: Missing link and other assorted corrections from Lynn's review) that you can add to your files to provide revision information (such as revision number, author, and comment log). If you select this option, DesignSync hides the keyword values (collapses the keywords) prior to comparing the files. For more information on RCE key processing, see DesignSync Data Manager User's Guide: Advanced Diff Options.

- **Ignore leading/trailing white space:** Determines whether white space (spaces, tabs) differences at the beginning or end of a line are ignored. For example, if a line in one file starts with a tab character whereas the same line in the other file starts with a space, Advanced Diff ignores the difference if you select this option.
- **Ignore changes in white space amount:** Determines whether differences in white space within a line are ignored. For example, if a line in one file has three spaces between two words whereas the same line in the other file has only one space, Advanced Diff ignores the difference if you select this option.
- **Perform binary comparison:** Performs the comparison in binary mode. When comparing files in binary mode, Advanced Diff only reports whether the files are identical or different. No other comparison options are available in binary mode.

Display Options

The **Display Options** determines how the results of a compare operation are displayed.

You can choose one of the following display options:

- **Display only the diffs (standard format)** uses the standard UNIX output format to display only lines that have been added, removed, or changed.
- **Display only the diffs (unified format)** uses the gnu unified output format to display both lines that are the same and lines that have changed.
- **Display only the diffs (syncdiff format)** uses a DesignSync-specific diff output format to display only lines that have been added, removed, or changed.
- **Display diff-annotated file** displays both lines that are the same in both files and lines that have changed.
- **Display output in GUI** displays the results graphically. You must have defined a graphical Diff utility in order to use this option. For more information, see System Administration Help: DesignSync diff Display Registry Settings.

The first four options display the results in the output window. The last option displays the results in the Graphical Diff utility.

See DesignSync Help: Reading Diff Results for examples of the different types of output.

Related Topics

[View History](#)

[ENOVIA Synchronicity Command Reference: diff](#)

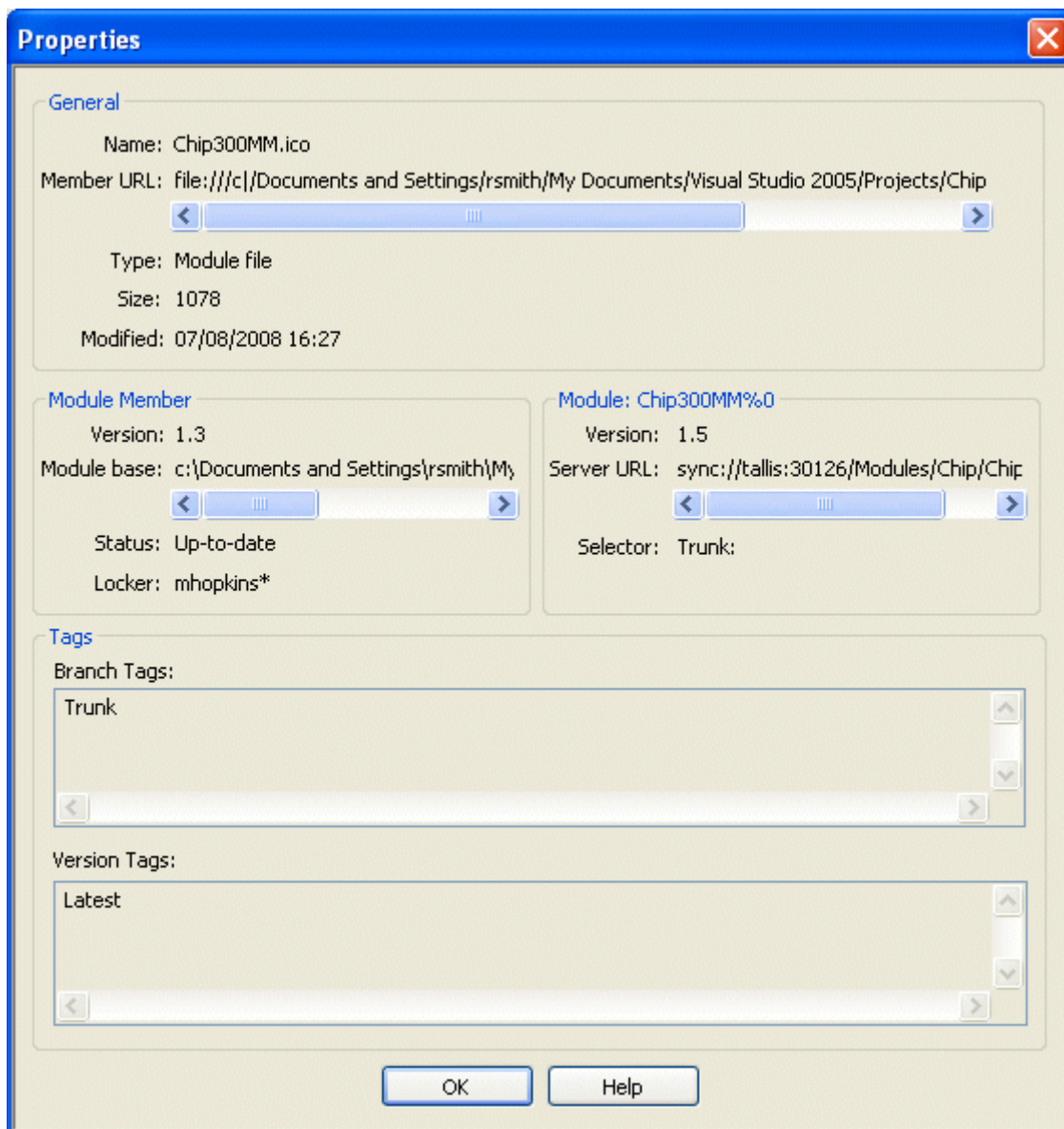
Properties

The **Properties** dialog box lets you view properties for a selected object. You select the object whose properties you want to view, then select **File > Source Control > DesignSync Properties**.

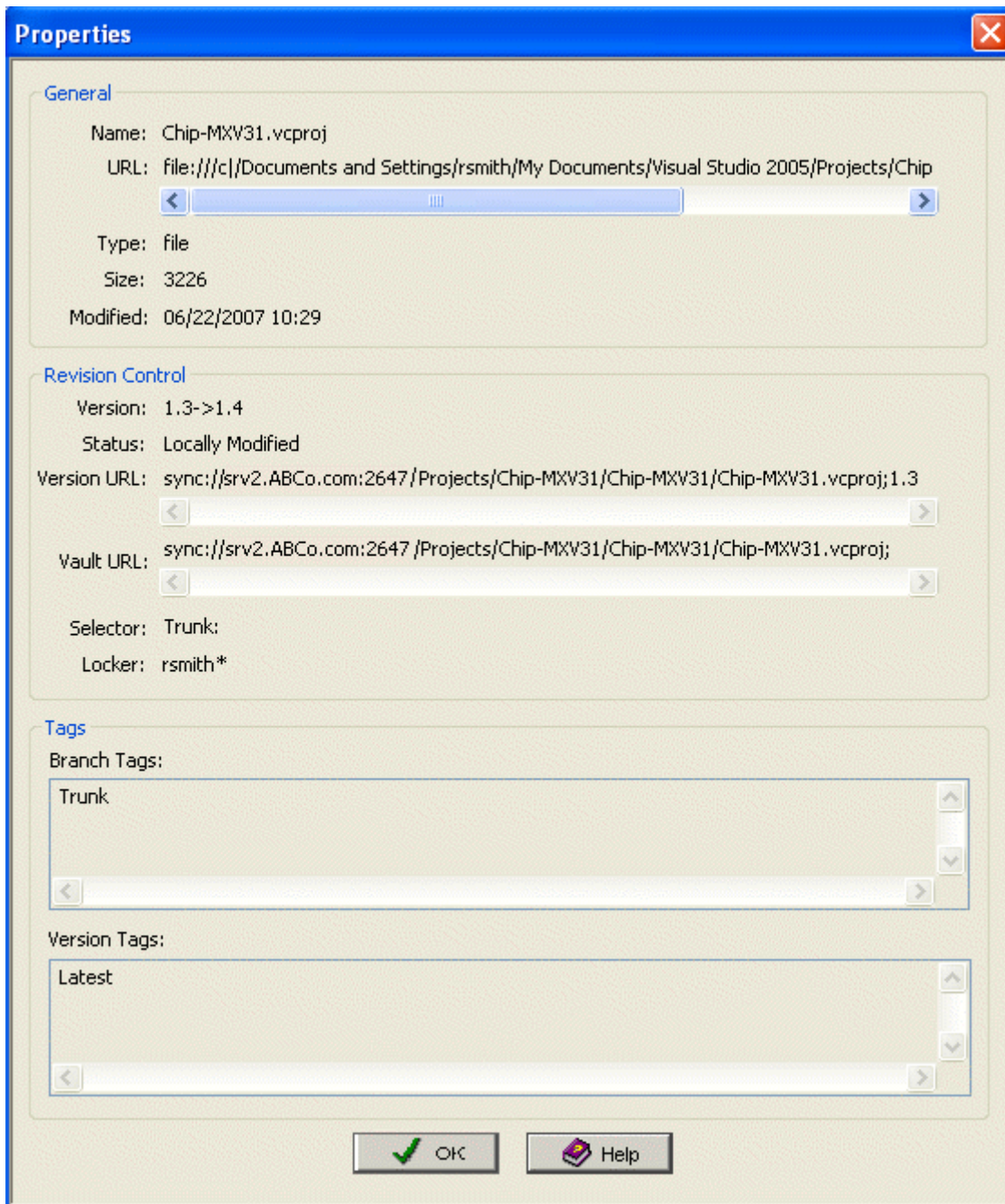
The information available from the **Properties** dialog box depends on the type of object you have selected.

Note: Unlike the Properties page available in the DesignSync GUI client, you cannot edit the object properties from this dialog box or select multiple objects.

This Properties sheet shows an example of a module member:



This Properties sheet shows an example of a non-module object:



Related Topics

[View History](#)

Launch SyncAdmin

The **File > Source Control > Launch DesignSyncSCC** option invokes the Synchronicity Administrator dialog box. See the SyncAdmin Help system for more information.

SyncAdmin allows you to control some of the options for the DSVS integration, such as specifying the location of the DSVS log file or defining a minimum comment length.

Related Topics

DSVS log files

Check In

Moving an Object

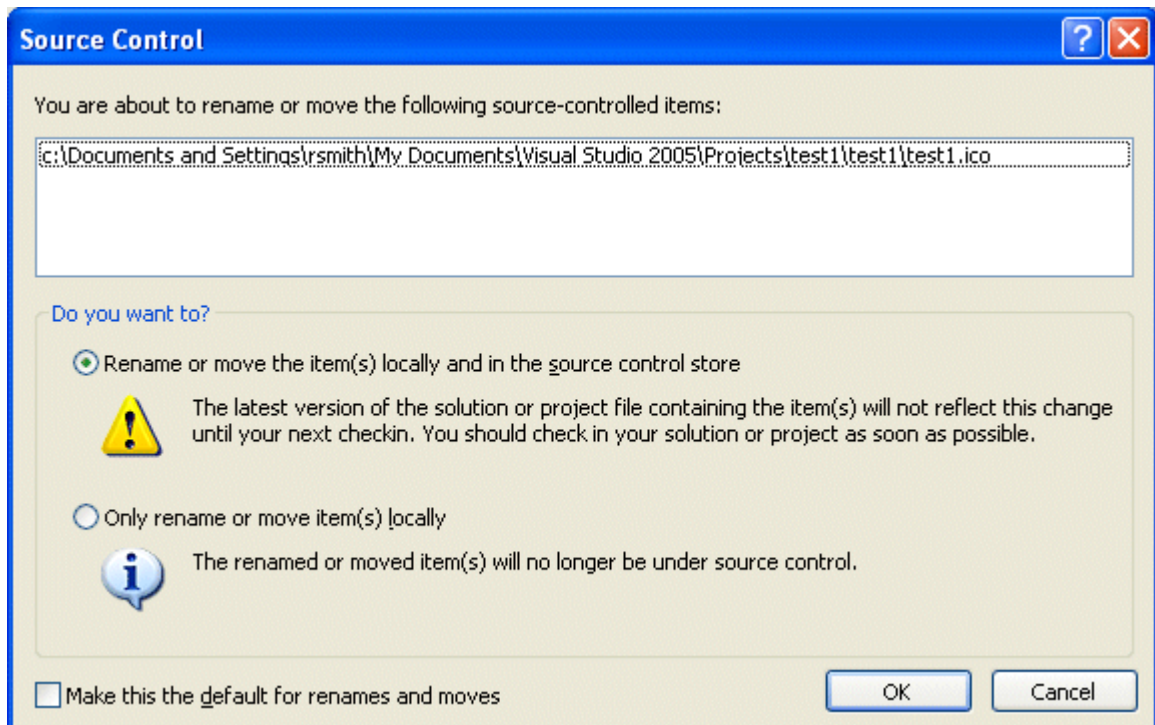
You can rename or move an object in the workspace and the corresponding server object with a single command.

Note: You cannot move a module. You can move module members within a module.

Important: The objects being moved must be checked in before moving them.

To move a Design object:

1. Select the desired object to rename or move.
2. Perform the move or rename as follows:
 - Open the context menu (right-click) and select the **Rename**. The file name becomes editable. Rename the file.
 - Drag the file to the new location to move it.
3. When you release the file, by pressing return or clicking on another element, the Source Control dialog box opens. This dialog provides a list of the objects moved or renamed and provides the option to either **Rename or move the item(s) locally and in the source control store**, or **Only rename or move the item(s) locally**.



Note: This example shows the Visual Studio dialog. Your SCC compliant may display a different window.

4. If you select **Only rename or move item(s) locally**, the delete operation renames or moves the file from the workspace and the project, but it remains on the server. If you select **Rename or move the item(s) locally in the source control store**, the operation performs the rename or move on the DesignSync server and moves the object in the workspace.

Related Topics

Check In

Deleting Objects

Visual Studio 2005 and other SCC compliant applications

If a design object is obsolete or no longer part of the design project, you can remove the design object from the project.

- **DesignSync vault objects:** DesignSync vault objects are retired from the branch of the project in the workspace. If other branches of the object exist, the

object remains on those branches. If a folder becomes empty as the result of retiring objects, the folder no longer appears in DesignSync. Retiring a branch prevents the branch from participating in future populate operations and prevents new versions from inadvertently being created on the branch. Users must specify the checkin of new items, which unretires the branch.

- **DesignSync module members:** DesignSync module members are removed from the module. They continue to exist in previous module versions.
- **DesignSync modules:** You can remove the module from your project or solution from within Visual Studio, but you must use a DesignSync client to remove a module from the DesignSync server.

For more information on deleting, retiring, or removing DesignSync objects, see DesignSync Help: Deleting Design Data. For information on un-retiring objects, see DesignSync Help: Retiring Design Data.

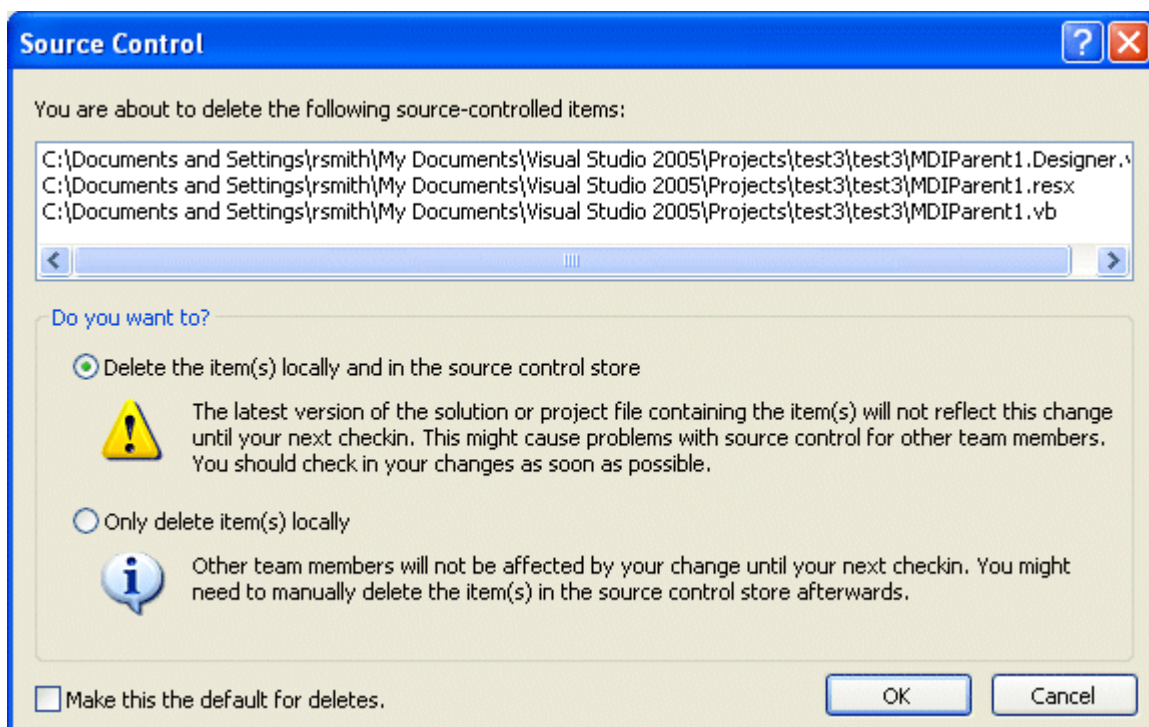
Important: The objects being deleted must be checked in to be removed from source control.

To delete a Design object:

1. Select the desired object to delete.
2. Select the **Edit > Delete** to open a warning about deleting objects from the project.

Note: Visual Studio 2005 uses **Edit > Remove** for module objects. Other SCC compliant applications may either use **Edit > Remove** or **Edit > Delete**.

3. Select **Ok** to delete the object. This opens the Source Control dialog box. This dialog provides a list of the objects selected for deletion and provides the option to either **Delete the item(s) locally and in the source control store**, or **Only delete item(s) locally**.



Note: This example shows the Visual Studio dialog. Your SCC compliant application may display a different window.

4. If you select **Only delete item(s) locally**, the delete operation removes the file from the workspace and the project, but it remains on the server. If you select **Delete the item(s) locally in the source control store**, the operation performs a remove on module members or a retire on module objects and deletes the object from the workspace and the project.

Visual Studio 2003

Delete the files from the project, then use a DesignSync client to remove the object from the module or retire the object from source control.


Related Topics

DesignSync Help: Deleting Design Data

Tag Options

Tagging is the application of a symbolic name, called a tag, to a version or a branch. Tags can only be applied to objects that are under revision control. For more information on the concept of tagging object versions or branches and naming conventions for tags see the DesignSync Help: Tagging Versions and Branches.

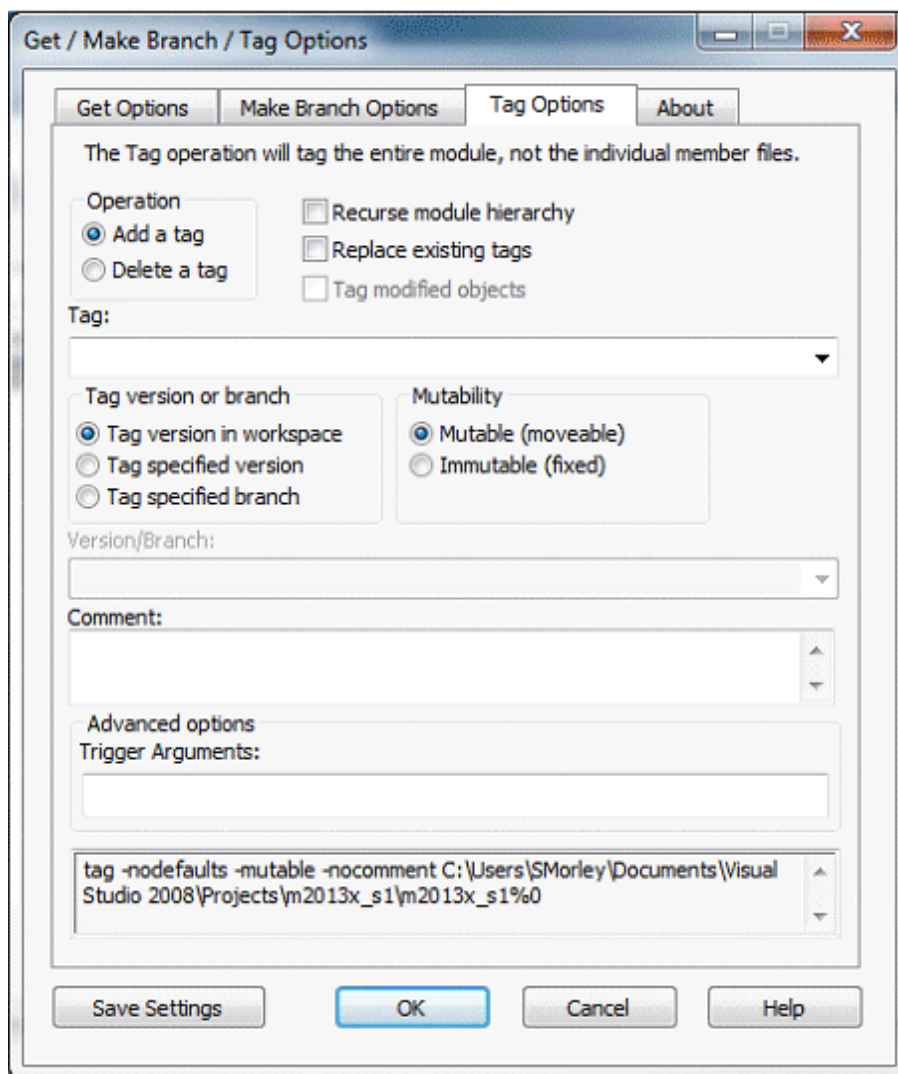
To Launch the Tag Options Dialog Box

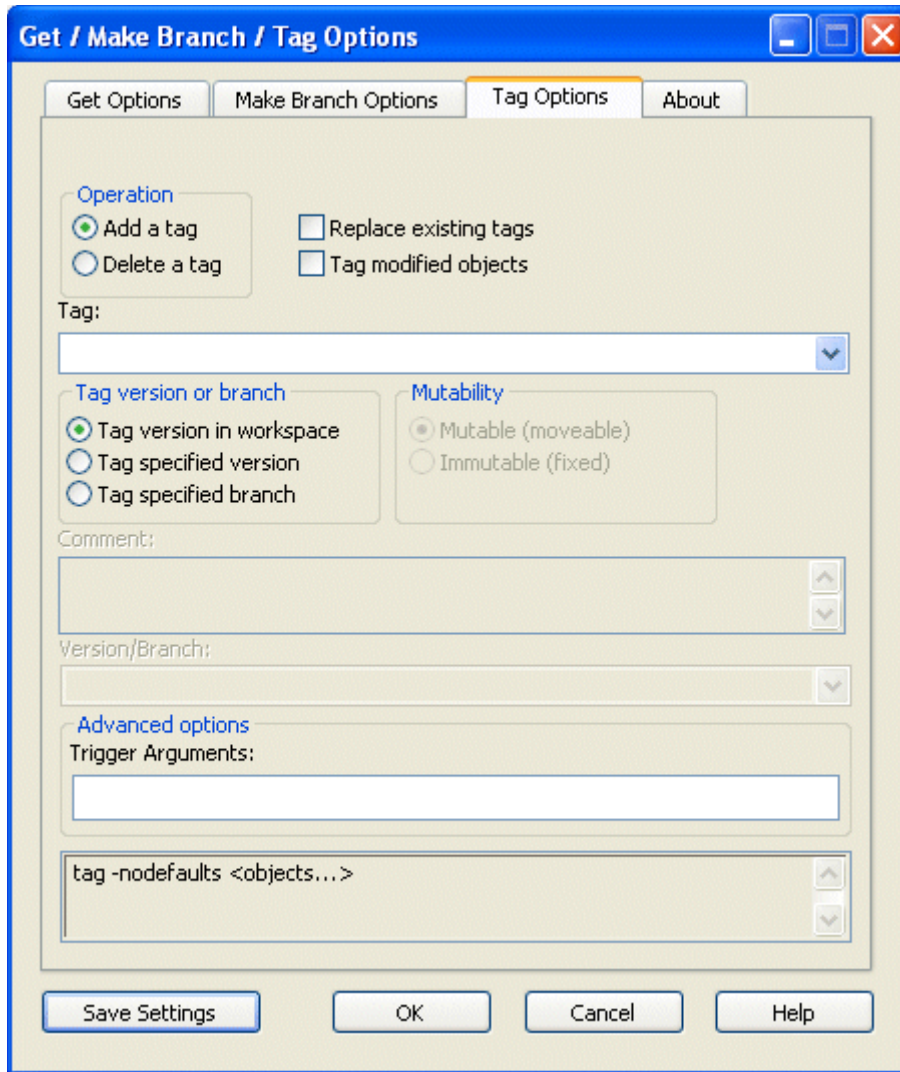
1. Select the module or objects to tag.
2. Select the **File > Source Control > Get...** to open the Get dialog box.
3. Select the options button () to launch the **Tag options** dialog.
4. Select the **Tag Options** tab.

The Tag Dialog Box

This shows the Tag Dialog box for both a module and module tag operation.

Click on the fields in either of the following illustrations for information.





Add a tag

Select this option to add a tag.

Delete a tag

Select this option to delete a tag. Because a tag can exist on only one version of a file at a time, the **Tag version or branch** field is ignored when you delete a tag.

Recurse

Replace existing tags

Move a tag to the version or branch specified in the **Version/Branch** field, even if that tag is already in use on another version or branch. For example, suppose that at the end of every week you want to select the latest files that produce a good demo and tag them "current_demo". To reuse the "current_demo" tag in this way, you must check the **Replace existing tags** check box.

By default (if you do not select **Replace existing tags**), a tag operation fails if the tag is already in use, because a tag can be attached to only one version or branch of an object at a time. Note that you can move a tag from a branch to a version or a version to a branch. DesignSync provides a warning message when you do so.

Tag modified objects

For modified objects in your work area, tag the version in the vault that you fetched to your work area. You might use this option, for example, if you have modified objects in your work area and you want to take a "snapshot" of them as they were before you made the changes.

If you do not specify this option, when the tag operation encounters a locally modified object, the operation displays an error for the object and does not tag any version of that object in the vault.

Note: This option affects modified objects only. If a work area object is unmodified, the tag operation tags the version in the vault that matches the one in your work area.

Tag

Enter a tag name here that is easily understood - for example, "Rel2.1", "ready_for_simulation", "current_demo", "Golden".

Tag names:

- Can contain letters, numbers, underscores (`_`), periods (`.`), hyphens (`-`), and forward slashes (`/`). All other characters, including white space, are prohibited.
- Cannot start with a number and consist solely of numbers and embedded periods (for example, 5, 1.5, or 44.33.22), because there would be ambiguity between the tag name and version/branch dot-numeric identifiers.
- Cannot end in `--R`. (The `--R` tag is reserved for use by legacy modules.)
- Cannot be any of the following reserved, case-insensitive keywords: Latest, LatestFetchable, VaultLatest, VaultDate, After, VaultAfter, Current, Date, Auto, Base, Next, Prev, Previous, Noon, Orig, Original, Upcoming, SyncBud, SyncBranch, SyncDeleted. Also, avoid using tag names starting with "Sync" (case-insensitive), because new DesignSync keywords conventionally use that prefix.

Tag version or branch

Specify which version or branch of the object to tag.

- To tag the version of the object in the vault that is the same as the one you have in your work area, select **Tag version in workspace**.
- To tag an object version in the vault different from the one in your work area, select **Tag specified version** and specify the version number or name in the **Version/Branch** field.
- To tag a specific branch of an object, select **Tag specified branch** and specify the branch name in the **Version/Branch** field.

Note: You can only tag the local workspace version of a module. If you select **Tag specified version** or **Tag specified branch** for a module, the current workspace version or branch is automatically populated into Version/Branch field.

Mutability

These options only pertain to module data. They are ignored when other types of objects are tagged.

When a tag is added, the new tag is marked as **Immutable (fixed)** or **Mutable (movable)**. A mutable tag can be replaced or deleted. To replace or delete an immutable tag, **Immutable (fixed)** must be selected. The default **Mutability** selection is **Mutable (movable)**.

Comment

Specify a branch creation comment (modules only). This comment is displayed in the object version history. Comments are limited to a maximum of 1024 characters.

Version/Branch

Specify the version or branch of the objects to tag. If you selected the **Tag specified version** option, you must specify a version selector or selector list. If you selected the **Tag specified branch** option, you must specify a single branch tag, a single version tag, a single auto-branch selector tag, or a branch numeric, but not a selector or selector list.

The **Version/Branch** field has a pull-down menu from which you can query for existing versions and branches. See Suggested Branches, Versions, and Tags for details.

Related Topics:

DesignSync Data Manager User's Guide: Tagging Versions and Branches

SyncAdmin Help: Tags


Make Branch Options

When a project requires a new branch, a single person, typically a release engineer or project manager, uses the **Make Branch** dialog box to branch all design objects at the same time. Users can then create a new work area for the new branch. The branch point version for make branch operations initiated within DSVS is always the version of the selected objects or module loaded into the workspace.

To Launch the Make Branch Options Dialog Box

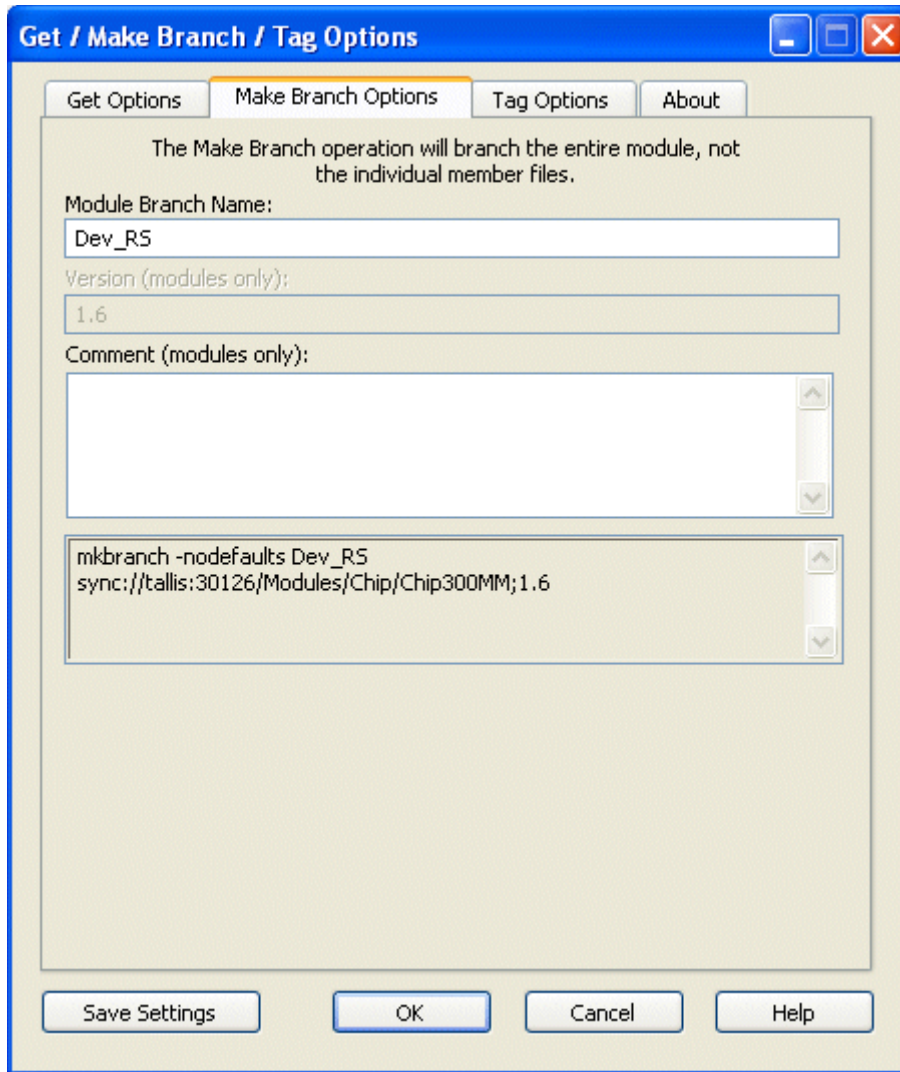
1. Select the module or objects to branch.

Note: You cannot branch individual module members, only the whole module.

2. Select the **File > Source Control > Get...** to open the Get dialog box.
3. Select the options button () to launch the **Make Branch options** dialog.
4. Select the **Make Branch Options** tab.

The Make Branch Dialog Box

Click on the fields in the following illustration for information.



Module Branch Name/Branch Name

Enter a branch tag name here that is easily understood - for example, "Rel2.1", "ready_for_simulation", "current_demo", "Golden".

Branch names:

- Can contain letters, numbers, underscores (`_`), periods (`.`), hyphens (`-`), and forward slashes (`/`). All other characters, including white space, are prohibited.
- Cannot start with a number and consist solely of numbers and embedded periods (for example, 5, 1.5, or 44.33.22), because there would be ambiguity between the tag name and version/branch dot-numeric identifiers.
- Cannot end in `--R`. (The `--R` tag is reserved for use by legacy modules.)
- Cannot be any of the following reserved, case-insensitive keywords: Latest, LatestFetchable, VaultLatest, VaultDate, After, VaultAfter, Current, Date, Auto, Base, Next, Prev, Previous, Noon, Orig, Original, Upcoming, SyncBud,

SyncBranch, SyncDeleted. Also, avoid using tag names starting with "Sync" (case-insensitive), because new DesignSync keywords conventionally use that prefix.

Note: Referring to a branch by its branch number is not recommended. When the branch is created, the branch name is automatically created as branch tag. You can add additional tags to the branch using Tag.

Version (modules only)

This read-only field displays the module branch point version for module branch operations. Because all make branch operations occur on the selected workspace version, this field is blank for non-module objects.

Comment (modules only)

Specify a branch creation comment (modules only). This comment is displayed in the object version history. Comments are limited to a maximum of 1024 characters.

Related Topics

Tag

DesignSync Data Manager User's Guide:Parallel (Multi-Branch) Development

Exclude from Source Control

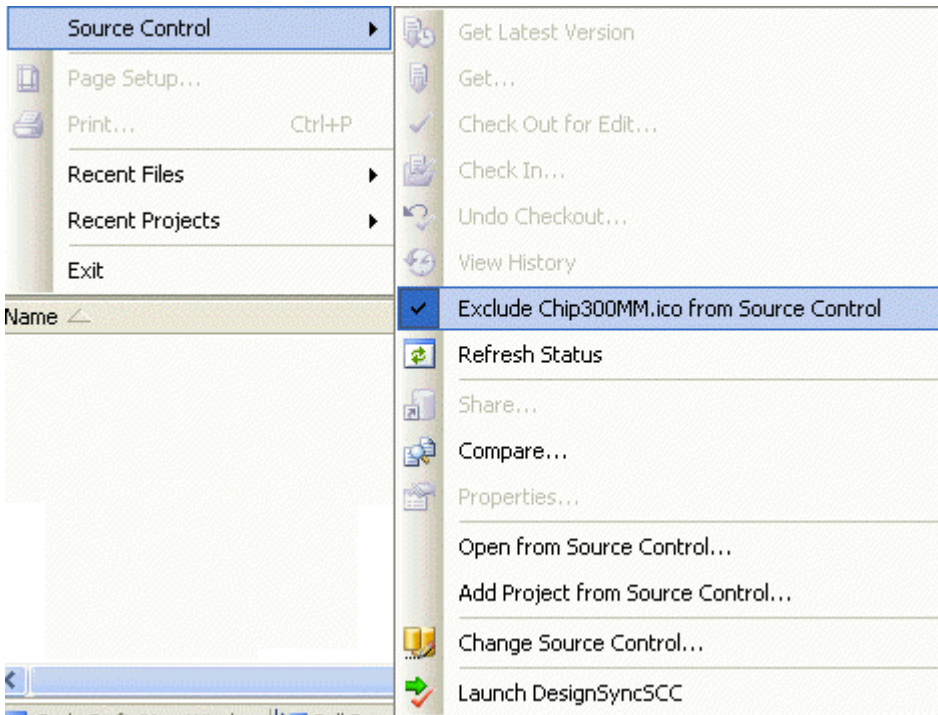
If the solution or project you are working on contains files that do not require source control, you can mark the file for exclusion. When you do this, the file remains in the solution, but it is no longer checked in or out with the containing project.

To exclude a file from source control:

1. Select the file in the Solution Explorer.
2. Select The **File > Source Control > Exclude from Source Control**.

This option is a toggle. When the **Exclude From Source Control** option appears as "checked," the file is excluded from source control options, as shown in the following image.

DesignSync for Visual Studio User's Guide



To re-add a file that has been excluded, repeat the exclude procedure to remove the check and include the file in subsequent source control operations.

Setting up a Workspace

Setting up a new Visual Studio Project with DesignSync

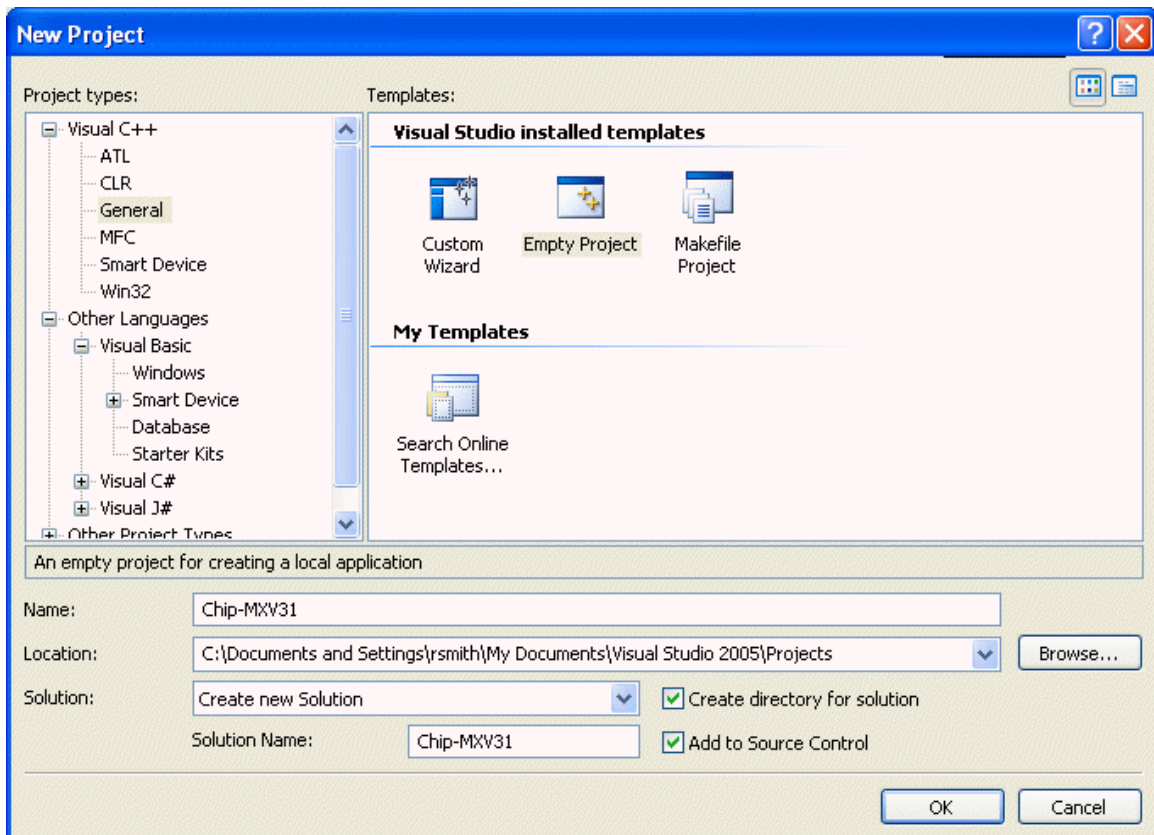
Creating a New Visual Studio Project and Associating it with DesignSync

When you create a new Visual Studio project or solution, you can immediately place it under DesignSync source control.

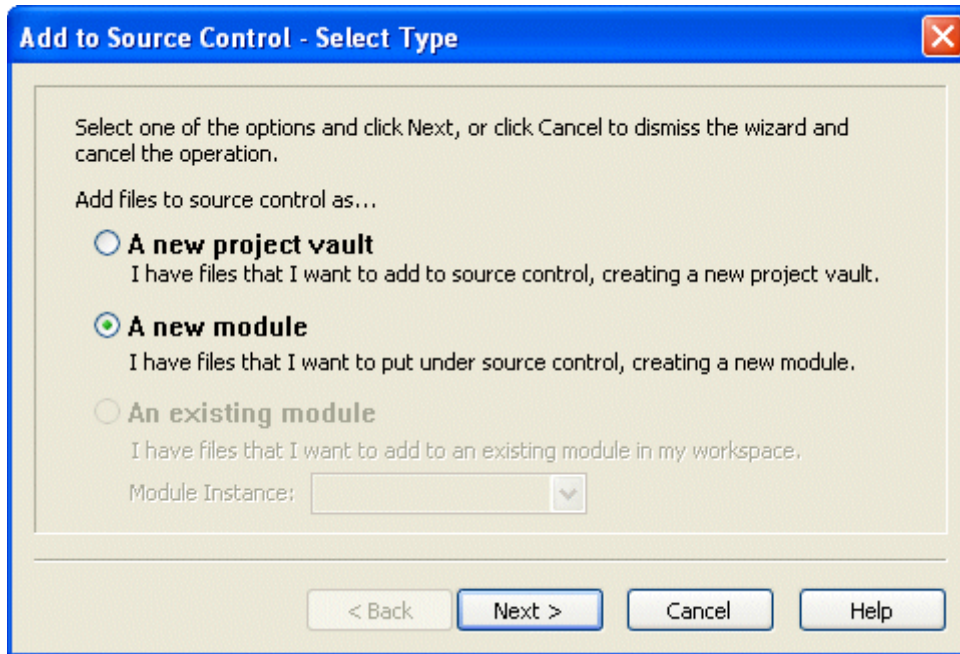
Important: After you've placed the project into DesignSync with Visual Studio, you should primarily use the DSVS integration to manage source control. If you are using the DesignSync GUI or other DesignSync clients, do not use these clients to change the module root directory or vault association for the project workspace or the workspace selector.

To create a new Visual Studio project and place it in DesignSync Source Control:

1. Create a new project as you would normally and check the **Add to Source Control** button on the New Projects panel.



2. Click **OK** to advance to next screen which allows you to select whether you are creating a new module or creating a new project vault.



3. Click **Next>** to advance to next screen:

For modules: Adding a Module

For DesignSync vaults: Set Vault Association.

Related Topics

Associating a Visual Studio Project with DesignSync

Check In

Confirm Source Control Association

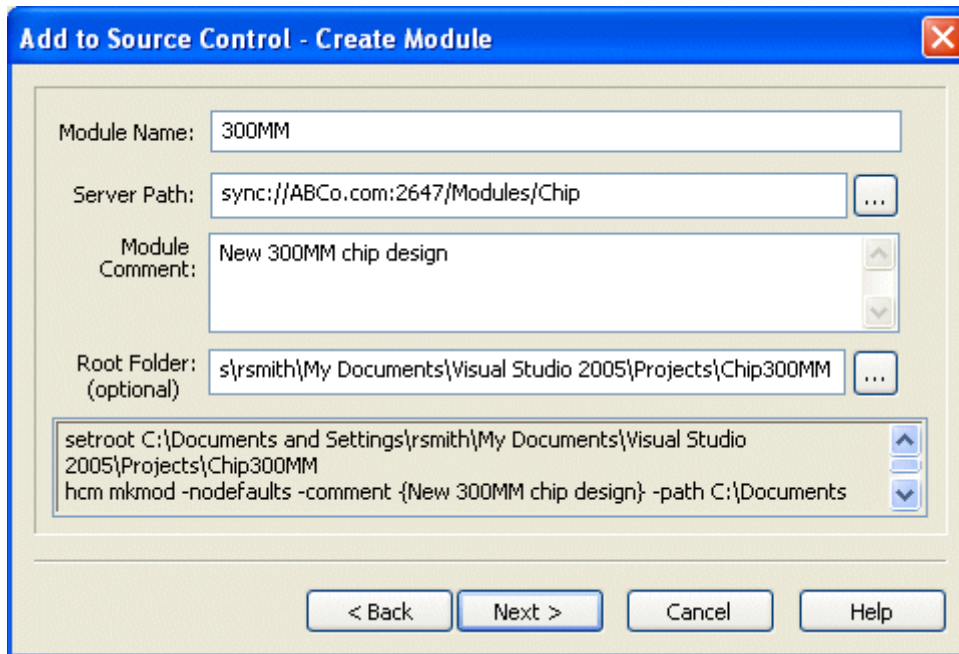
Adding a Module

When you connect a Visual Studio project, new or existing, to a DesignSync module, you must specify the location and name of the module on the server.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to DesignSync, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Module Name

The name of the module. See DesignSync Data Manager User's Guide: URL Syntax for a list of illegal characters.

Server Path

The server URL and category path for the new module, in the form:

```
sync://<host>:[<port>]/Modules/[<category>...]
```

The `<category>` is optional, and may be a path, such as `/Project1/mymods`. A module *category* is similar to a *path* to a DesignSync project or vault. The term *category* is used to indicate that this is a virtual path, rather than a physical path. And that its use is intended to categorize modules.

Module Comment

An optional description of the module.

Root Folder

Specifies the workspace root folder. This should not be confused with the module's base directory. The workspace root is typically one folder above the module's base directory. If the workspace root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the workspace root is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created as a module.

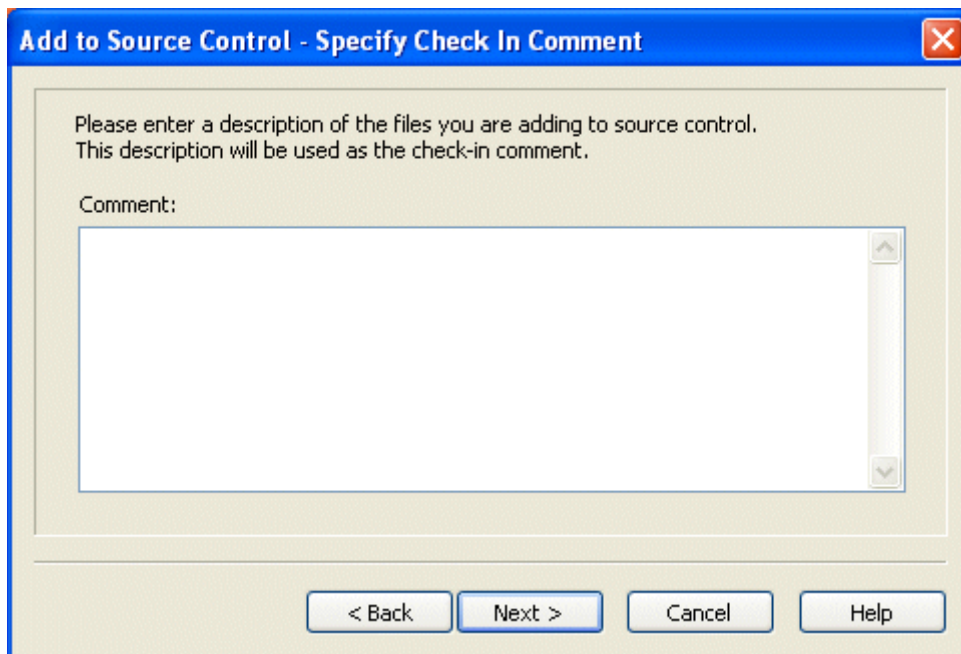
Related Topics

Check In

Check In Comment

This checkin comment is used for the initial check-in of files added to the module or project vault during the initial checkin.

Note: This is different than the module comment applied to the module at module creation, which is applied to version 1.1 of the module.



Before the checkin begins, you must confirm the Add parameters for the checkin.

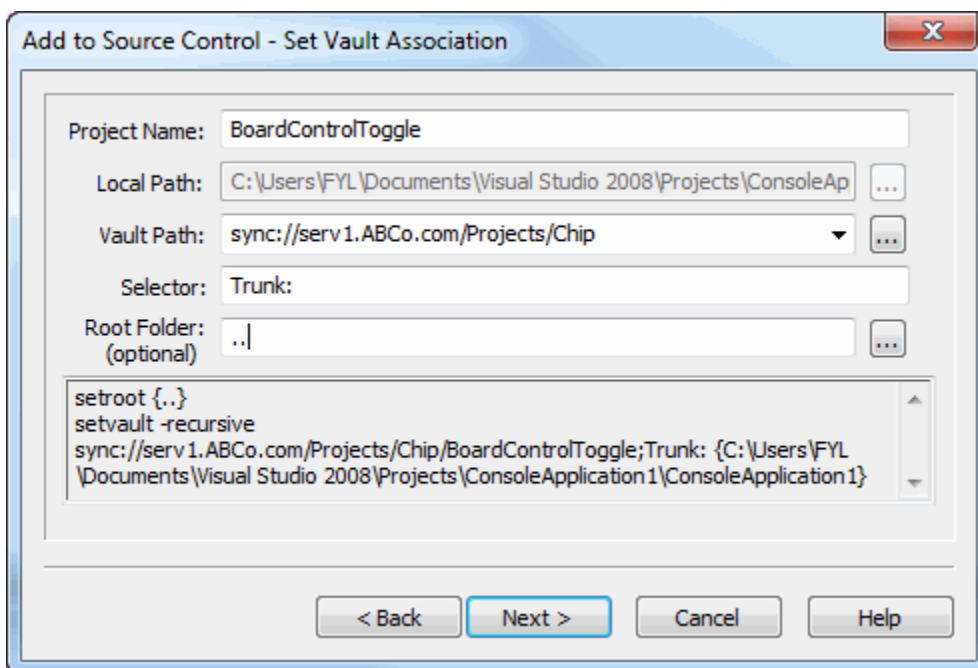
Set Vault Association

When you connect a Visual Studio project, new or existing, to a DesignSync vault using the Visual Studio Integration, you must set the vault association. You cannot set the vault association on a module directory.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to a DesignSync vault, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Project Name

The name of the top-level vault folder for the project.

Local Path

The directory path to the project on your local machine.

Note: If the path doesn't exist already, Visual Studio prompts you to confirm creation.

Vault Path

The vault path to the project on the DesignSync server.

Selector

The design objects version designator. You can specify:

- A branch selector, typically Trunk.
- A project configuration, as created using ProjectSync, that has been applied to the project vault you are accessing. See the ProjectSync Help for additional information about project configurations.

Root Folder

Specifies the workspace root folder. This should not be confused with the local path directory. The root folder is typically one folder above the local path directory.

If the workspace root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the root folder is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created.

The results of the command appear in the output window.

Related Topics

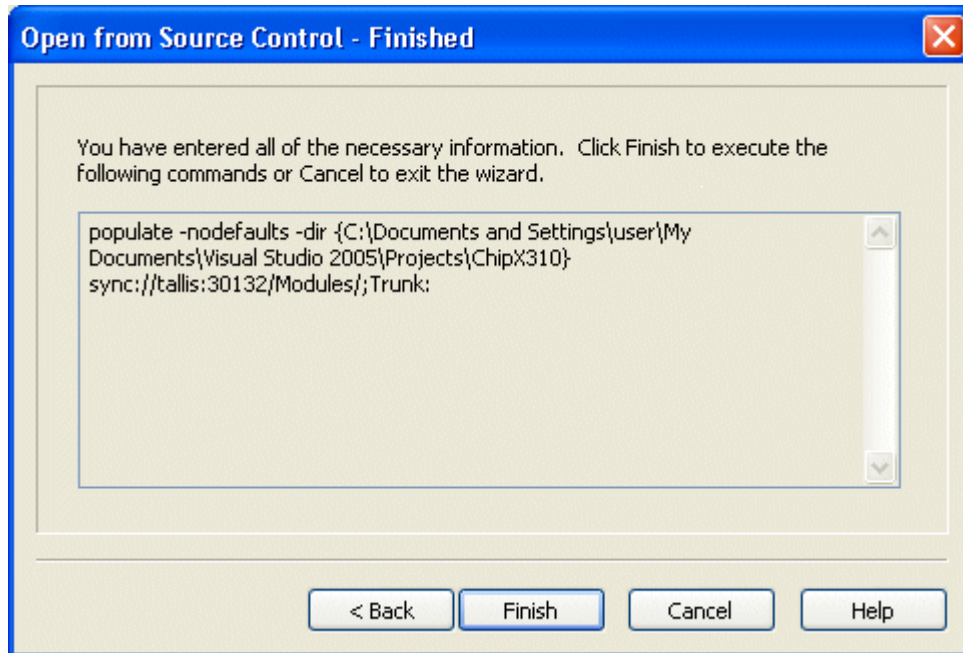
[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Check In](#)

Confirm Source Control Association

After you've defined the settings for associating the Visual Studio Project with the corresponding DesignSync module or vault location, you receive a confirmation screen that displays the command you will run. If the command is correct, press finish to

launch the



command.

Related Topics

[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Associating a Visual Studio Project with DesignSync](#)

[Associating an Existing DesignSync module or vault with Visual Studio](#)

Associating a Visual Studio Project with DesignSync

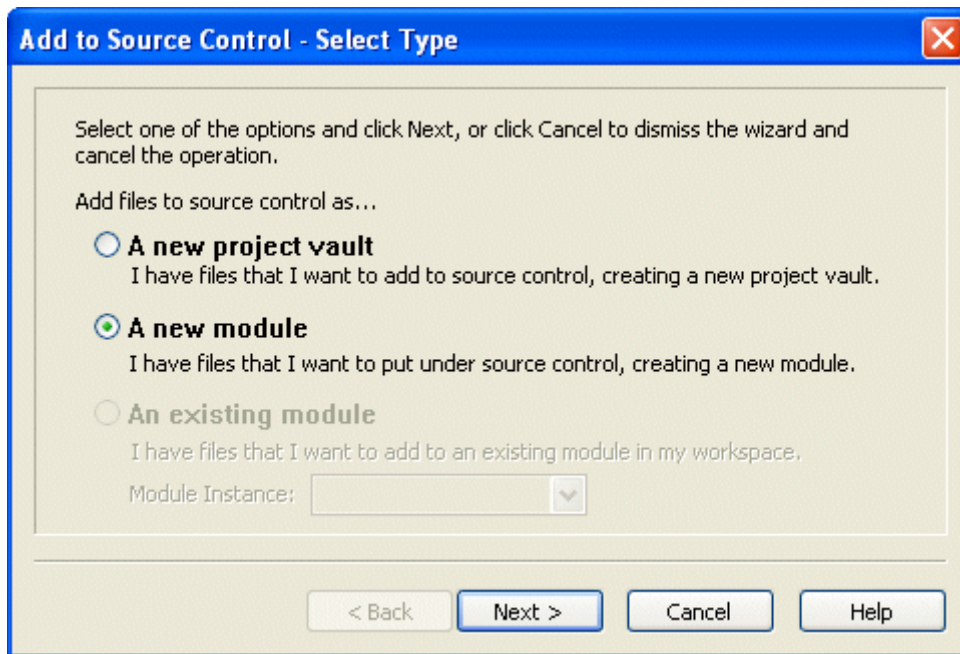
Associating a Visual Studio Project with DesignSync

When you use a Visual Studio Project with DesignSync, you must use the Visual Studio environment to associate the project with the data vault in DesignSync.

Important: After you've placed the project into DesignSync with Visual Studio, you should primarily use the DSVS integration to manage source control. If you are using the DesignSync GUI or other DesignSync clients, do not use these clients to change the vault association for the project workspace or the workspace selector.

1. Open the Visual Studio Project.
2. Select the appropriate option from the menu: **File > Source Control > Add Solution to Source Control...** or **File > Source Control > Add Selected Projects to Source Control....**

- This opens the Add to Source Control - Select Type wizard, allowing you to decide whether the project should be placed under source control as a project vault or a new module.



Select the appropriate type and press **Next>** to advance to the next screen:
 For **A New Project vault**: Set Vault Association.
 For **A New Module**: Adding a Module.
 For **An existing module**: Selecting an Existing Module.

Related Topics

Creating a New Visual Studio Project and Associating it with DesignSync

Confirm Source Control Association

Check In

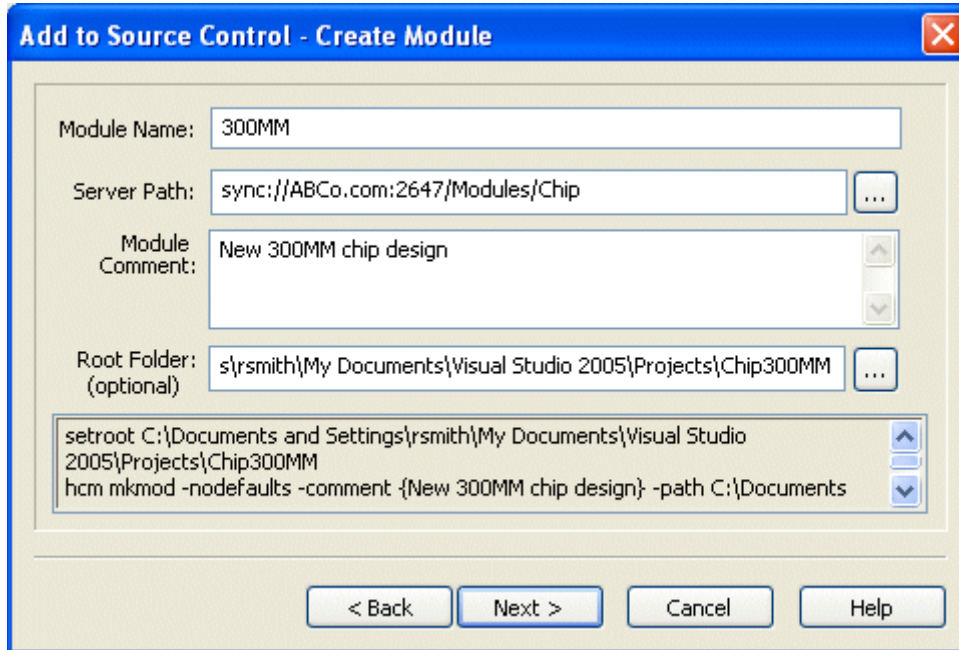
Adding a Module

When you connect a Visual Studio project, new or existing, to a DesignSync module, you must specify the location and name of the module on the server.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to DesignSync, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Module Name

The name of the module. See DesignSync Data Manager User's Guide: URL Syntax for a list of illegal characters.

Server Path

The server URL and category path for the new module, in the form:

```
sync://<host>:[<port>]/Modules/[<category>...]
```

The `<category>` is optional, and may be a path, such as `/Project1/mymods`. A module *category* is similar to a *path* to a DesignSync project or vault. The term *category* is used to indicate that this is a virtual path, rather than a physical path. And that its use is intended to categorize modules.

Module Comment

An optional description of the module.

Root Folder

Specifies the workspace root folder. This should not be confused with the module's base directory. The workspace root is typically one folder above the module's base directory. If the workspace root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the workspace root is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created as a module.

Related Topics

Check In

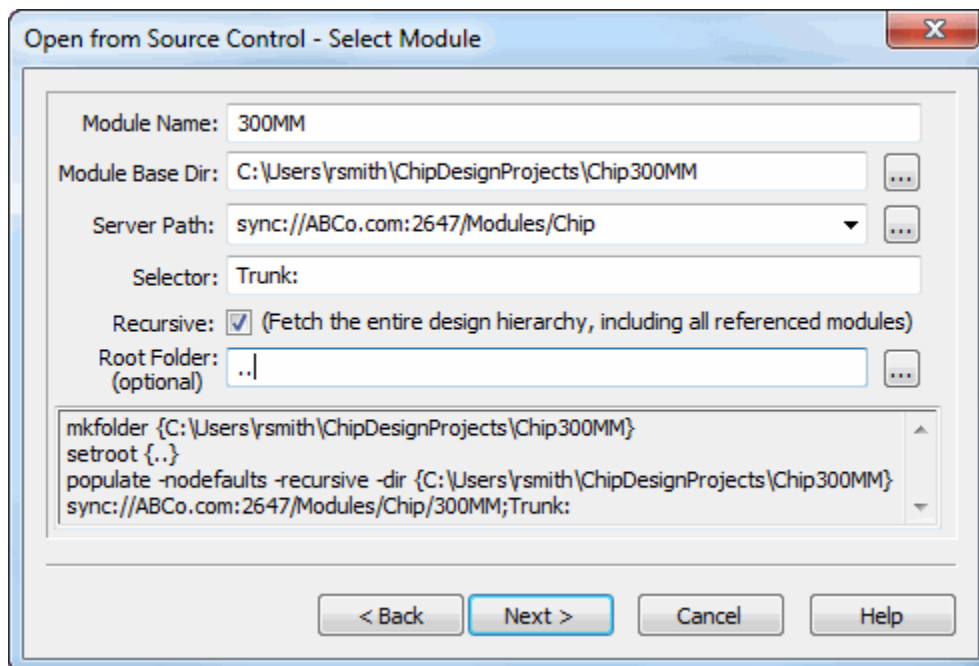
Selecting an Existing Module

When you connect a Visual Studio project, new or existing, to a DesignSync module, you must specify the location and name of the module on the server.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to DesignSync, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Project Name

The name of the Visual Studio project or solution file stored in the module. The Visual Studio project must be located at the top level of the module, not in a module sub-folder.

Note: This is the name of your project. It is not necessarily the same as the name of the module.

Module Base Dir

The workspace folder for the module being populated. By default, this is empty. Use the **Browse...** button to browse for a folder on your system.

Tip: To allow you to more easily differentiate between a module name and a workspace folder name, do not use an initial capital letter to begin the folder name and begin your module names with an initial capital letter.

Note: If the selected folder doesn't already exist, you are prompted to confirm creation.

Server Path

The server URL and category path for the module, in the form:

```
sync://<host>:[<port>]/Modules/[<category>...]/<ModuleName>
```

Note: The **Browse...** button uses the `sync_servers.txt` file to locate candidate modules. For information on setting up `sync_servers.txt`, see the DesignSync System Administration Help: SyncServer List Files.

Selector

The version designator for the workspace. You can specify:

- A branch selector, typically Trunk.
- A version selector, typically Latest.

Recursive

Specifies whether to populate the module recursively, following any hierarchical references contained in the module. By default, this is not checked, indicating the hierarchically referenced modules are not populated.

Root Folder (optional)

Specifies the module's root folder. This should not be confused with the module's base directory. The root folder is typically one folder above the module's base directory. If the root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the root folder is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created as a module.

Related Topics

Associating an Existing DesignSync module or vault with Visual Studio

Associating a Visual Studio Project with DesignSync

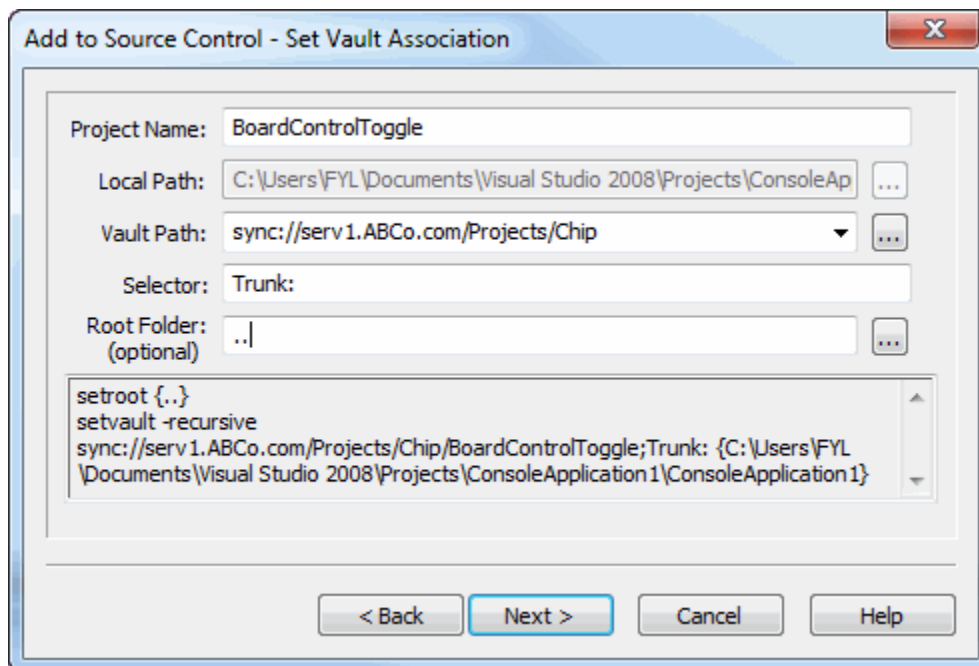
Set Vault Association

When you connect a Visual Studio project, new or existing, to a DesignSync vault using the Visual Studio Integration, you must set the vault association. You cannot set the vault association on a module directory.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to a DesignSync vault, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Project Name

The name of the top-level vault folder for the project.

Local Path

The directory path to the project on your local machine.

Note: If the path doesn't exist already, Visual Studio prompts you to confirm creation.

Vault Path

The vault path to the project on the DesignSync server.

Selector

The design objects version designator. You can specify:

- A branch selector, typically Trunk.
- A project configuration, as created using ProjectSync, that has been applied to the project vault you are accessing. See the ProjectSync Help for additional information about project configurations.

Root Folder

Specifies the workspace root folder. This should not be confused with the local path directory. The root folder is typically one folder above the local path directory. If the workspace root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the root folder is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created.

The results of the command appear in the output window.

Related Topics

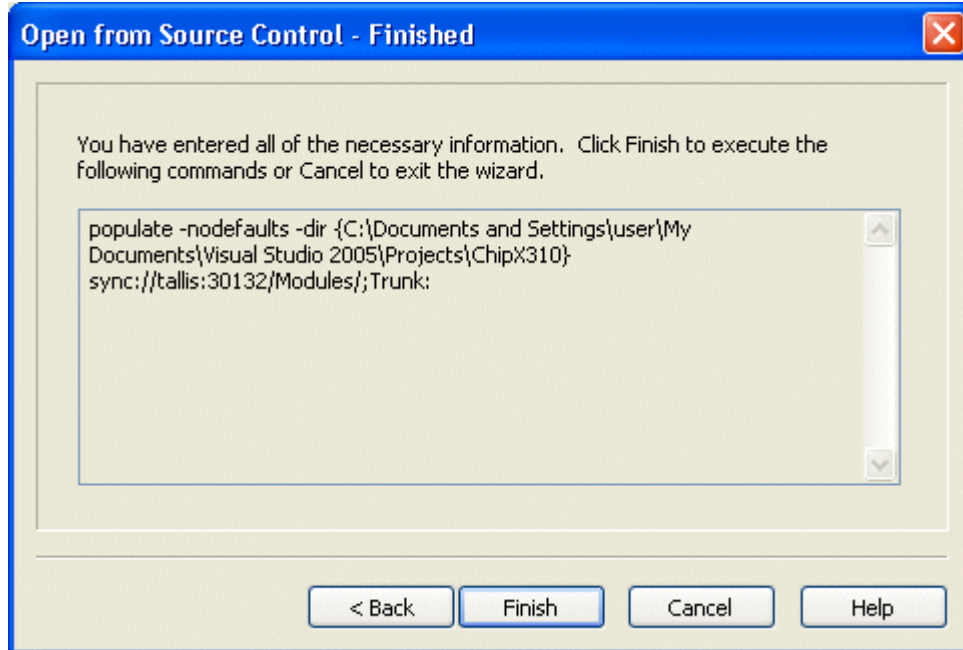
[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Check In](#)

Confirm Source Control Association

After you've defined the settings for associating the Visual Studio Project with the corresponding DesignSync module or vault location, you receive a confirmation screen

that displays the command you will run. If the command is correct, press finish to launch the



command.

Related Topics

[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Associating a Visual Studio Project with DesignSync](#)

[Associating an Existing DesignSync module or vault with Visual Studio](#)

Associating a DesignSync module or vault with Visual Studio

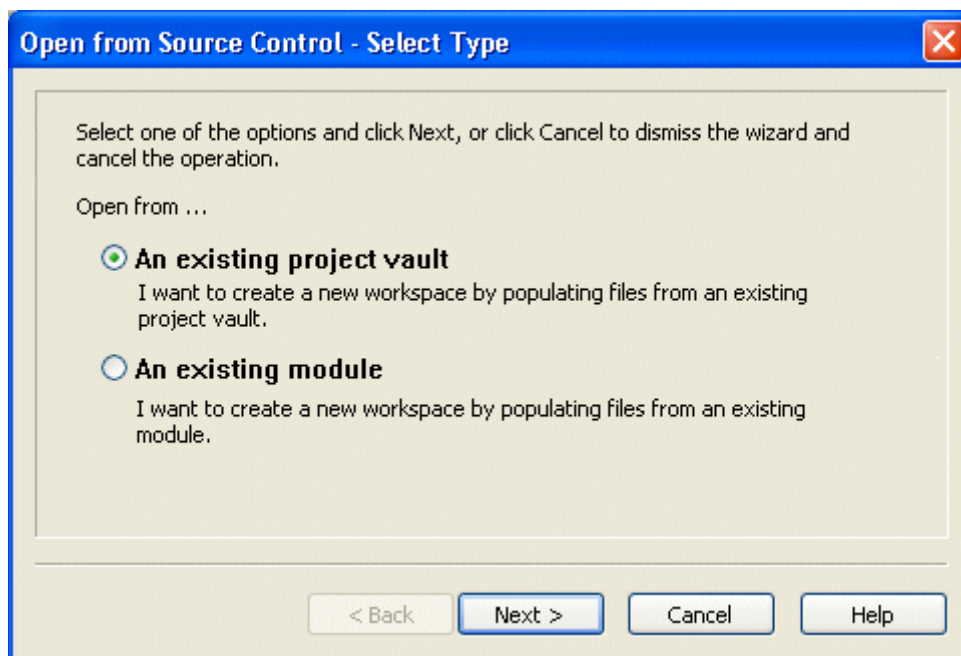
Associating an Existing DesignSync Module or Vault with Visual Studio

Visual Studio can automatically create a new workspace for a Visual Studio project that is already under source control with DesignSync, but has never been opened in your Visual Studio client.

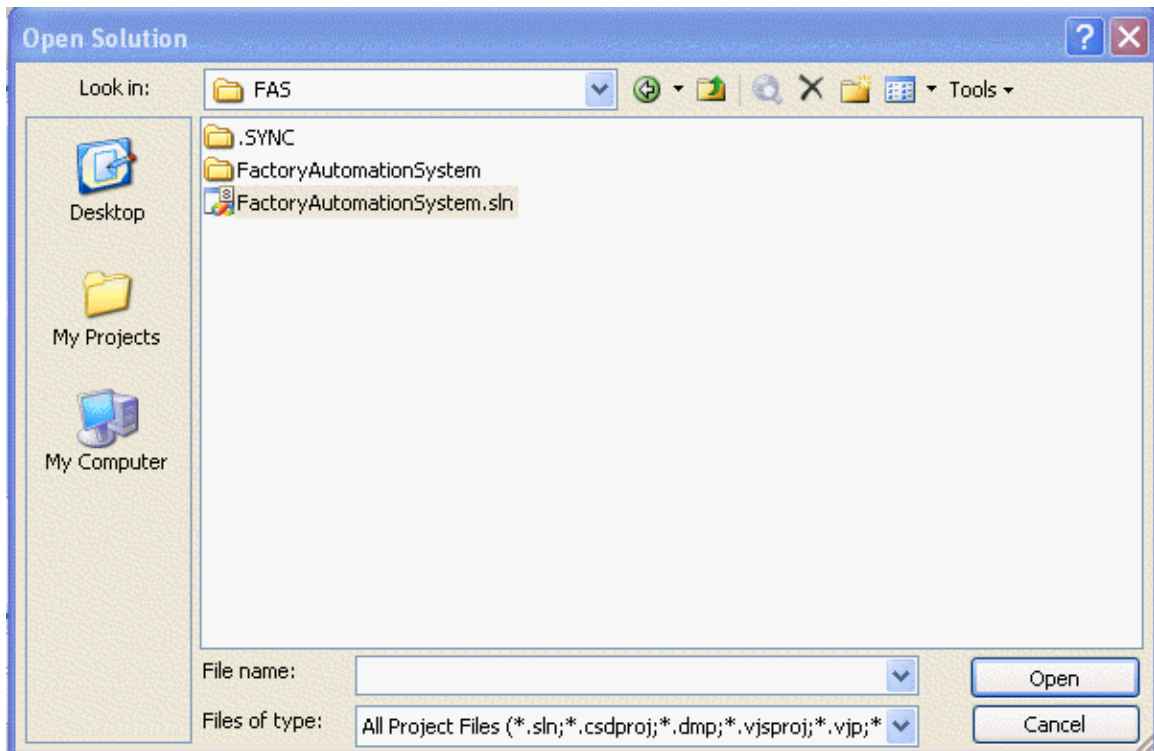
Important: After you've placed the project into DesignSync with Visual Studio, you should primarily use the DSVS integration to manage source control. If you are using the DesignSync GUI or other DesignSync clients, do not use these clients to change the vault association for the project workspace or the workspace selector.

To open an existing Visual Studio Project already in source control with a new local workspace:

1. From the Menu bar, select: **File | Source Control | Open From Source Control...**
2. This opens the Add to Source Control - Select Type wizard, allowing you to decide whether the project should be placed under source control as a project vault or a new module. Select the appropriate type and press **Next>**.



3. If the Project is a project vault, you see the **Set Vault Association** screen. If the Project is a module, you see the **Selecting an Existing Module** screen. After you have finished specifying the source control information, press return to create the workspace settings and copy the project file to the specified local path.
4. After the workspace setup is complete, Visual Studio prompts you to select the local solution file. Select the appropriate solution file and press **Open**.



5. The project opens in the selected workspace. You can now perform all the available revision control operations to manage your project.

Related Topics

Creating a New Visual Studio Project and Associating it with DesignSync

Associating a Visual Studio Project with DesignSync

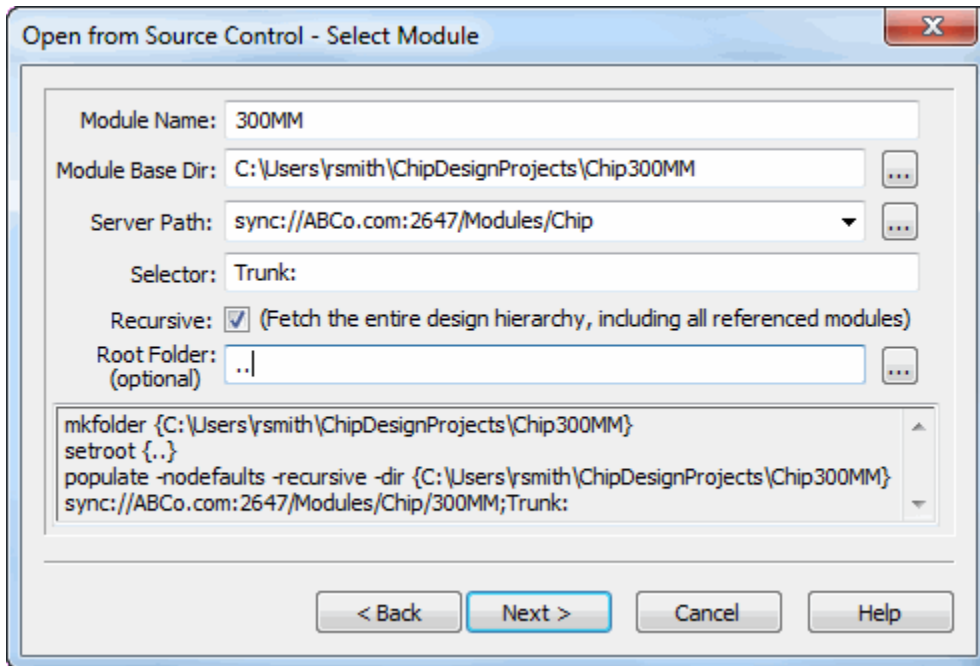
Selecting an Existing Module

When you connect a Visual Studio project, new or existing, to a DesignSync module, you must specify the location and name of the module on the server.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to DesignSync, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Project Name

The name of the Visual Studio project or solution file stored in the module. The Visual Studio project must be located at the top level of the module, not in a module sub-folder.

Note: This is the name of your project. It is not necessarily the same as the name of the module.

Module Base Dir

The workspace folder for the module being populated. By default, this is empty. Use the **Browse...** button to browse for a folder on your system.

Tip: To allow you to more easily differentiate between a module name and a workspace folder name, do not use an initial capital letter to begin the folder name and begin your module names with an initial capital letter.

Note: If the selected folder doesn't already exist, you are prompted to confirm creation.

Server Path

The server URL and category path for the module, in the form:

```
sync://<host>:[<port>]/Modules/[<category>...]/<ModuleName>
```

Note: The **Browse...** button uses the `sync_servers.txt` file to locate candidate modules. For information on setting up `sync_servers.txt`, see the DesignSync System Administration Help: SyncServer List Files.

Selector

The version designator for the workspace. You can specify:

- A branch selector, typically Trunk.
- A version selector, typically Latest.

Recursive

Specifies whether to populate the module recursively, following any hierarchical references contained in the module. By default, this is not checked, indicating the hierarchically referenced modules are not populated.

Root Folder (optional)

Specifies the module's root folder. This should not be confused with the module's base directory. The root folder is typically one folder above the module's base directory. If the root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the root folder is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created as a module.

Related Topics

Associating an Existing DesignSync module or vault with Visual Studio

Associating a Visual Studio Project with DesignSync

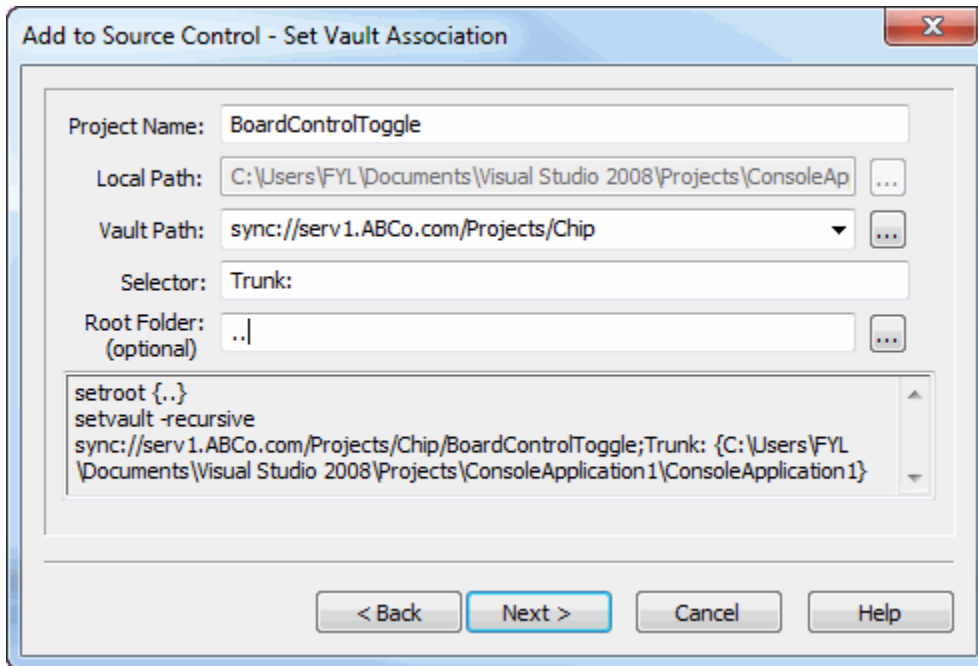
Set Vault Association

When you connect a Visual Studio project, new or existing, to a DesignSync vault using the Visual Studio Integration, you must set the vault association. You cannot set the vault association on a module directory.

After you've defined the settings described below and pressed **Next>**, you advance to the Confirm Source Control Association screen.

Note: To connect to a DesignSync vault, you may need to save your DesignSync server password. For more information on saving your DesignSync password, see the Synchronicity Command Reference: Password command.

Click on the fields in the following illustration for information.



Project Name

The name of the top-level vault folder for the project.

Local Path

The directory path to the project on your local machine.

Note: If the path doesn't exist already, Visual Studio prompts you to confirm creation.

Vault Path

The vault path to the project on the DesignSync server.

Selector

The design objects version designator. You can specify:

- A branch selector, typically Trunk.

- A project configuration, as created using ProjectSync, that has been applied to the project vault you are accessing. See the ProjectSync Help for additional information about project configurations.

Root Folder

Specifies the workspace root folder. This should not be confused with the local path directory. The root folder is typically one folder above the local path directory.

If the workspace root directory has already been defined, the interface displays the default value and does not allow you to change it.

If the root folder is not specified or already defined, DSVS uses the folder one level above the folder containing the specified files, solution, or project being created.

The results of the command appear in the output window.

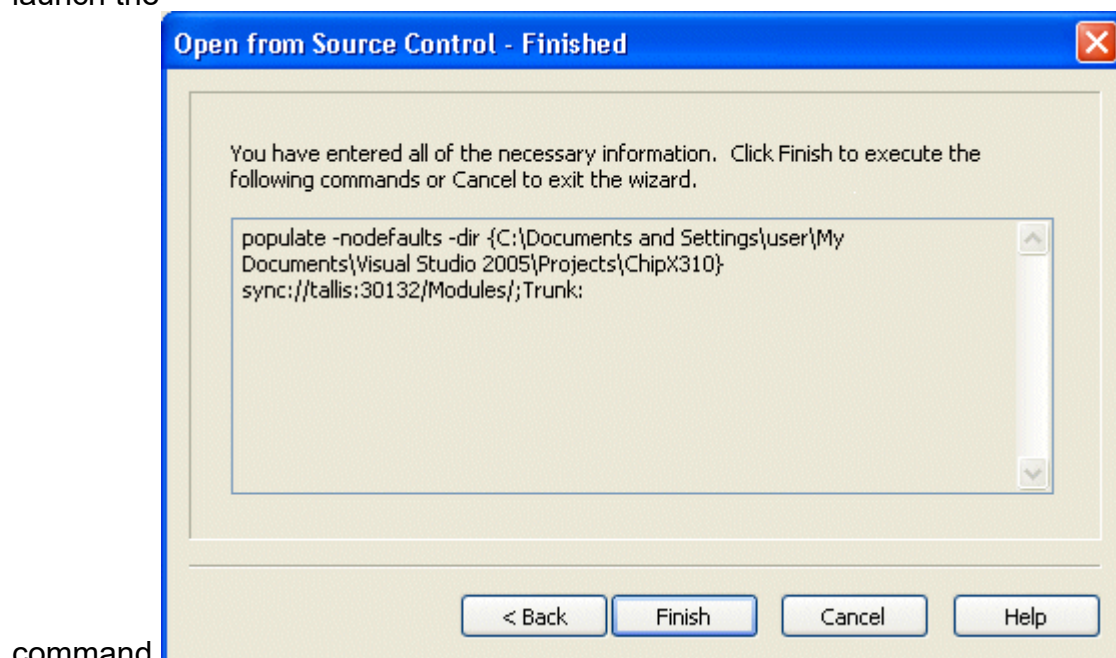
Related Topics

[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Check In](#)

Confirm Source Control Association

After you've defined the settings for associating the Visual Studio Project with the corresponding DesignSync module or vault location, you receive a confirmation screen that displays the command you will run. If the command is correct, press finish to launch the



Related Topics

[Creating a New Visual Studio Project and Associating it with DesignSync](#)

[Associating a Visual Studio Project with DesignSync](#)

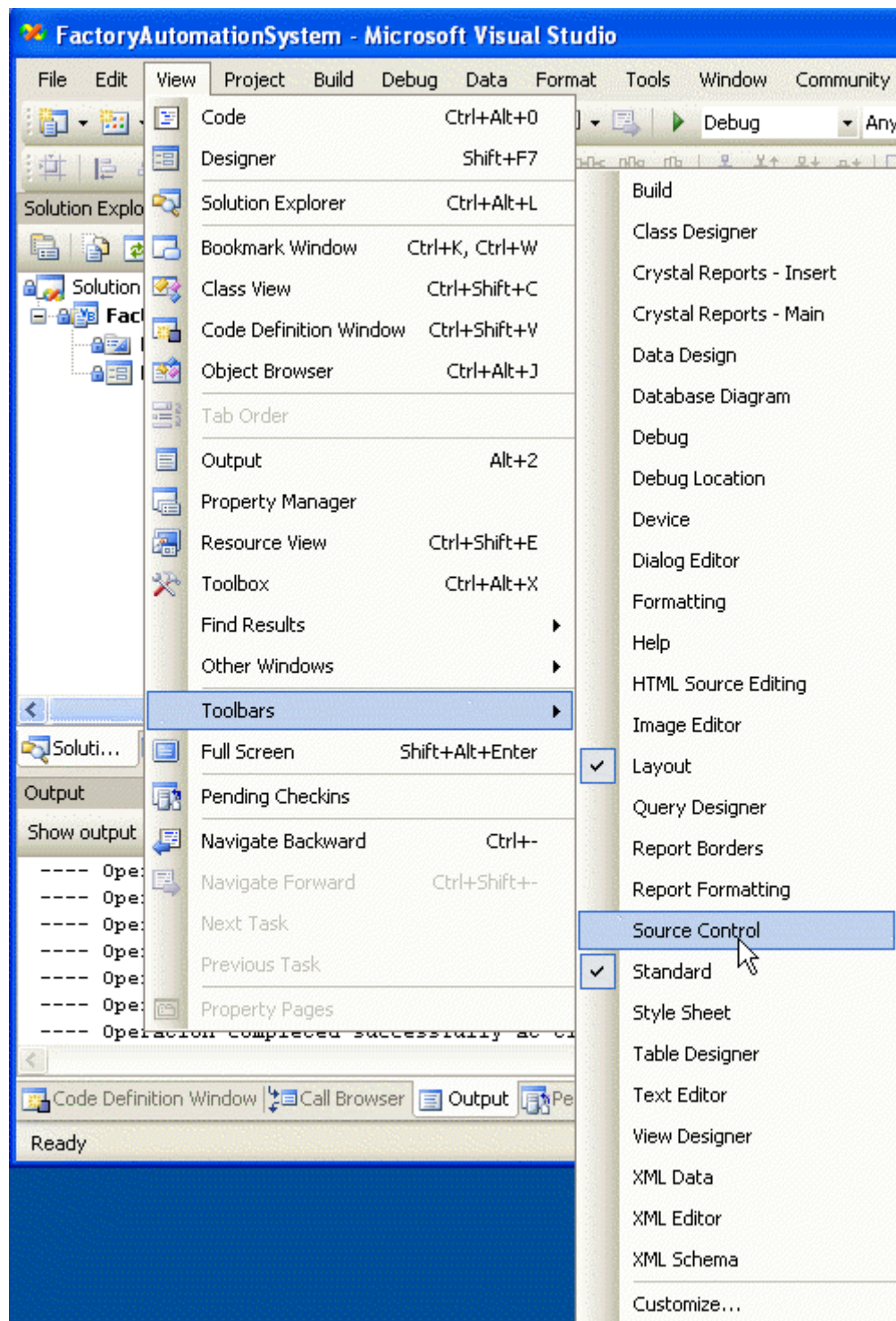
[Associating an Existing DesignSync module or vault with Visual Studio](#)

Configuring

Displaying the Source Control Toolbar

You can use the Source Control Toolbar to access DesignSync functionality.

1. Right click on the Visual Studio toolbar or select **View > Toolbars** to view the list of available toolbars and select **Source Control**



The Source Control toolbar is now visible in the application.

Note: You can customize the Source Control toolbar using **View > Toolbars > Customize**. If, after customizing the toolbar, you reset the toolbar to its default state, the toolbar does not display properly until you also reset the DesignSync Add-in.

Changing Source Control

When your module project has branched, or, your non-module project vault has branched or moved to a different DesignSync server, you can change the definition of your project workspace to point to the new location.

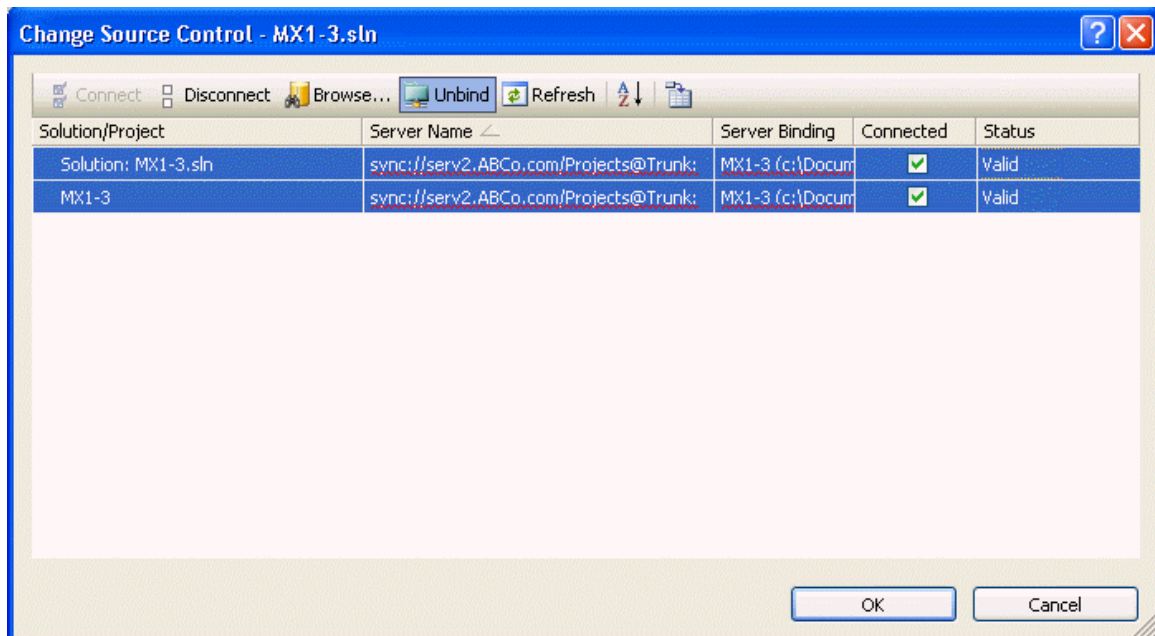
Important: When you use unbind a project from source control with the Change Source Control dialog, you must use the Change Source Control dialog to bind it to the new source control location. Do not try to add the project to source control using other procedures, wizards, or dialogs.

Note: The screens for the procedure below show changing the branch for the Visual Studio Project.

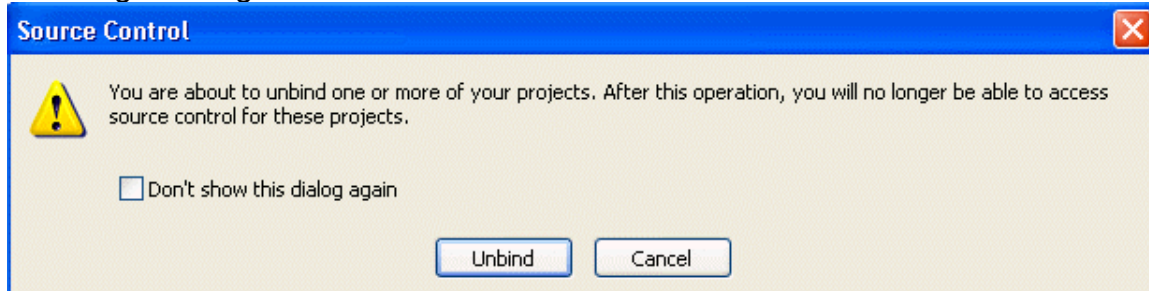
To Change Source Control:

1. Open the Visual Studio Project.
2. From the Menu bar, select: **File > Source Control > Change Source Control...**

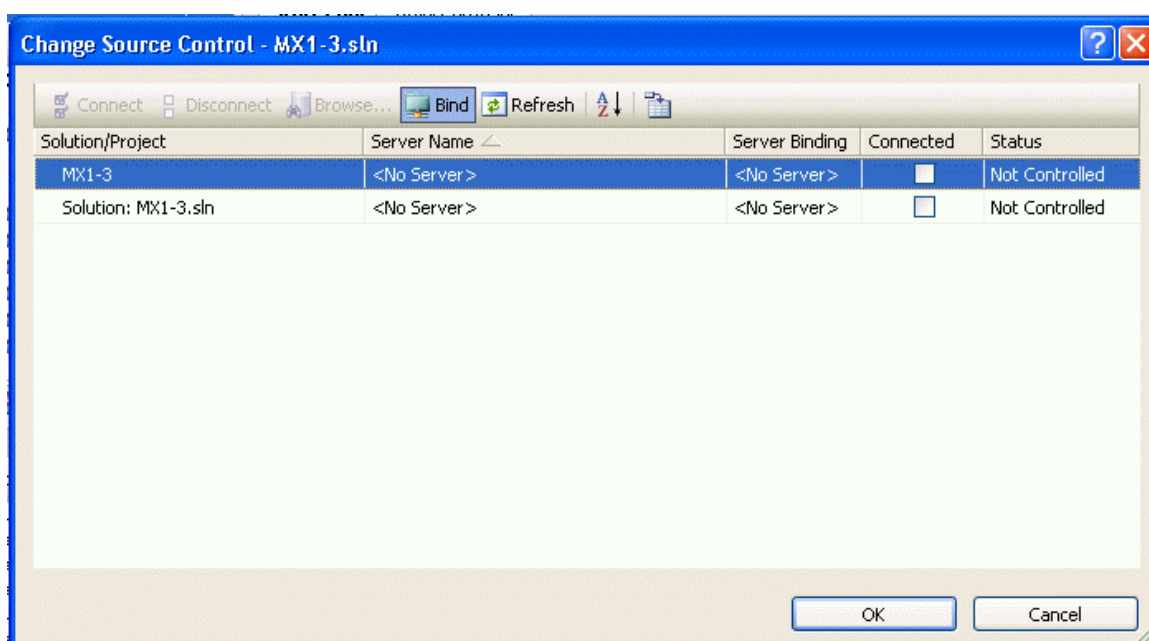
This shows the current project bindings.



- Click Unbind to remove the current server binding settings. You will see the following warning:

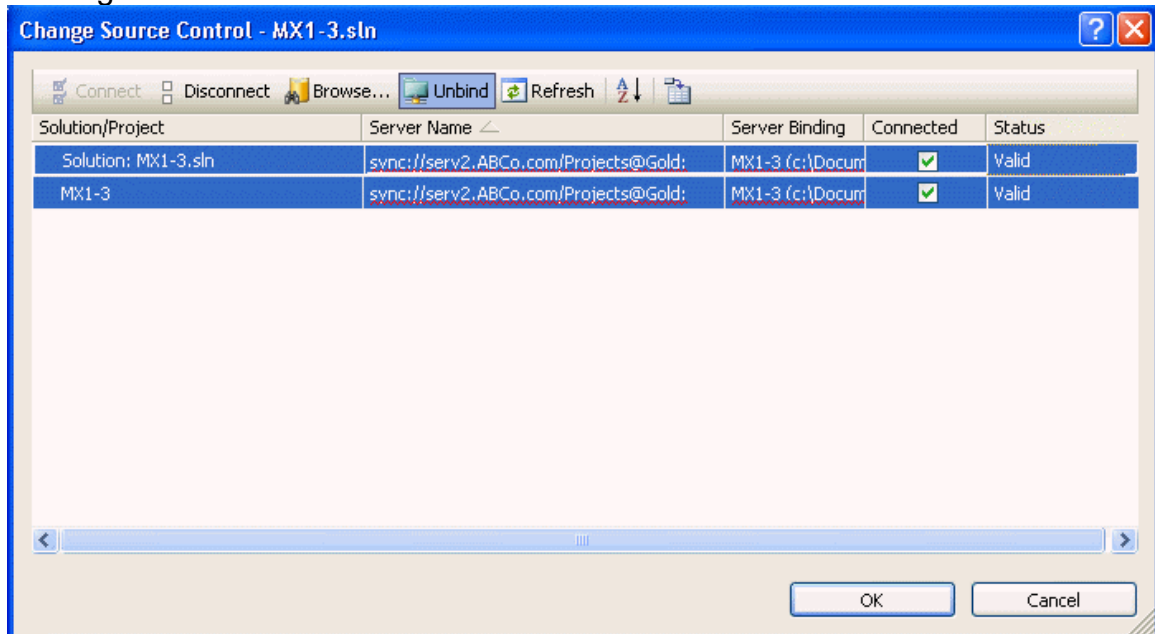


- When the project has been disconnected from source control, you see the Unbind button has become a Bind button.



- Click Bind to open the either the Change Source Control screen and define the new location.
- Visual Studio automatically validates the server binding. During validation, the status of the specified Project, Solution or File is "Unknown." After the binding has been confirmed, the status reads "Valid." If you receive a status of "Invalid", verify the spelling of the server or branch tag selected or check that the server is

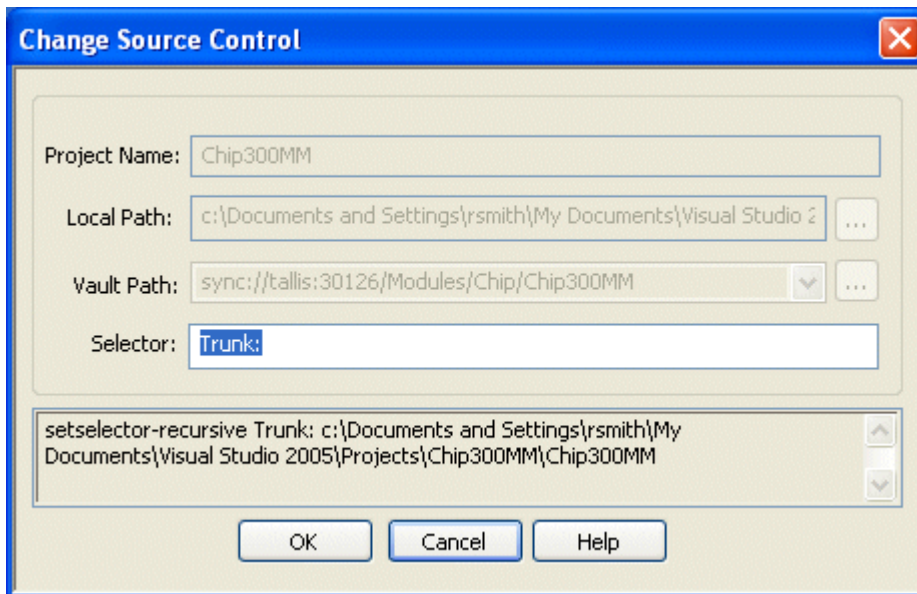
running.



7. Click OK to close the Change Source Control Window and confirm your changes.

The Change Source Control Dialog Box

Click on the fields in the following illustration for information.



Project Name

The name of the top-level vault folder for the project. (non-module vaults only)

Local Path

The directory path to the project on your local machine. (non-module vaults only)

Note: If the path doesn't exist already, Visual Studio prompts you to confirm creation.

Vault Path

The vault path to the project on the DesignSync server. (non-module vaults only)

Selector

The design objects version designator. You can specify:

- A branch selector, typically Trunk.
- A project configuration, as created using ProjectSync, that has been applied to the project vault you are accessing. See the ProjectSync Help for additional information about project configurations.

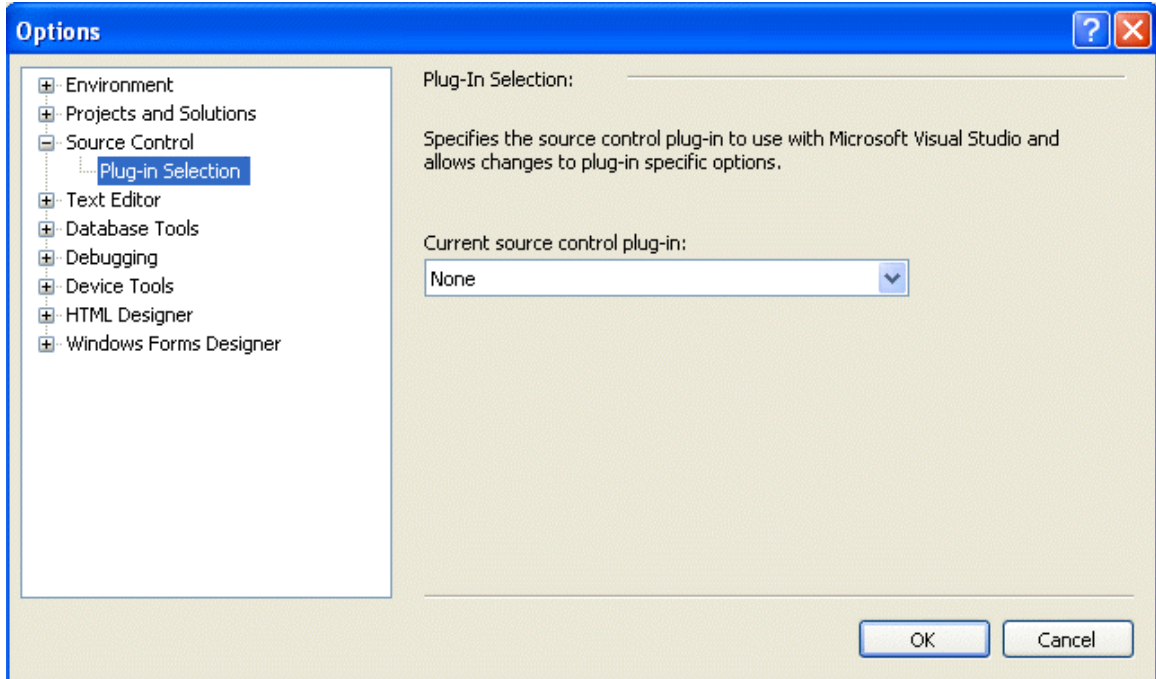
The results of the command appear in the output window.

DSVS Options

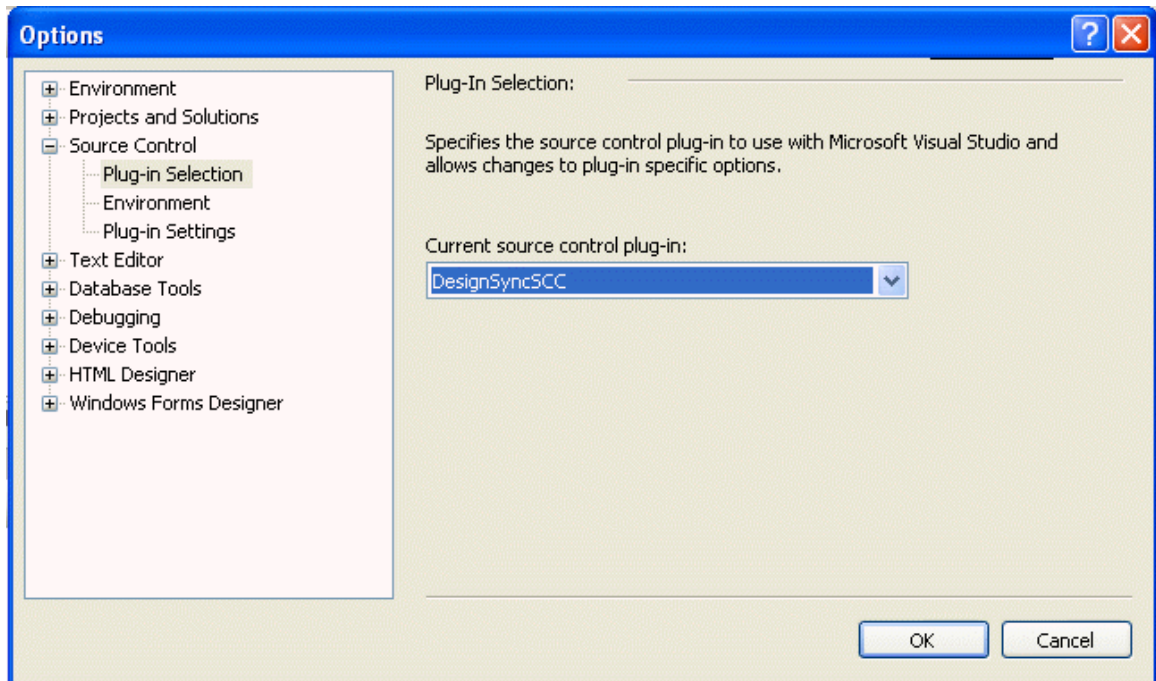
Visual Studio 2005 allows you to select DesignSync as your source control solutions, view your default login name, and set certain source control options.

To Open the DSVS plug-in Options for Visual Studio 2005:

1. From the Menu bar, select: **Tools > Options > Source Control**.



2. From the **Current source control plug-in** list, select **DesignSyncSCC**.



Troubleshooting

About DSVS

DSVS version information is displayed as an About box attached to the Source Control definition. The About box also includes contact and copyright information for Dassault Systemes and DSVS.

To Launch the About Dialog Box

1. Select the **Tools > Options >** to open the Visual Studio options dialog.
2. On the Options dialog, select **Source Control > Plug-in Settings**.
3. Click the Advanced... button to launch the **About** dialog.

Note: The About box is also displayed as part of the **Get Options** dialog box as the last tab.

Related Topics

[Get Options](#)

[About the DesignSync Visual Studio Integration](#)

DSVS log files

In addition to displaying the revision control operations output to the Visual Studio Output window, DSVS maintains a set of log files. The log files are stored in the same location as the DesignSync client logs. The log location is specified in SyncAdmin. For information on choosing on log location, see the SyncAdmin Help: The Logging Tab.

DSVS uses a daily log, regardless of the number of DSVS clients used in a day. If a DSVS session spans more than a single calendar day, there is a new DSVS log created for each day named:

```
DSVS_<LogBeginDate>.log
```

While the command is in process, DSVS logs the command messages to a temporary log in the same location as the DSVS log file called <command>.log. After the command completes, the results of the command log file are appended to the daily DSVS log file. The command log file is overwritten the next time the command is run.

Related Topics

[SyncAdmin Help: The Logging Tab](#)

Could Not Find Key Error

When reviewing the log file, you may see errors such as "Could not find <RegistryKeyName> under key <RegistryKeyPath>." These errors are non-critical and result from DesignSync checking the registry keys for any user-defined default values.

If there isn't a user defined value for the key, DesignSync uses the default value and echoes the "error" that it was not able to find a registry key to the log.

Related Topics

DSVS log files

System Administration Guide: Overview of Registry Files

Reset DesignSync Add-in

Visual Studio can cache information from the DesignSync Add-in. If you update your DesignSync client or reset the Source Control Toolbar, you may need to reset the Visual Studio Add-in to re-enable the Add-in functionality.

To reset the Visual Studio Add-in (Visual Studio 2005 only):

1. With Visual Studio closed, open the Visual Studio 2005 command prompt.
2. Type the following command

```
devenv /resetaddin DesignSyncSCC.Connect
```

3. This opens Visual Studio.

Related Topics

Enabling the DesignSync SCC Add-in

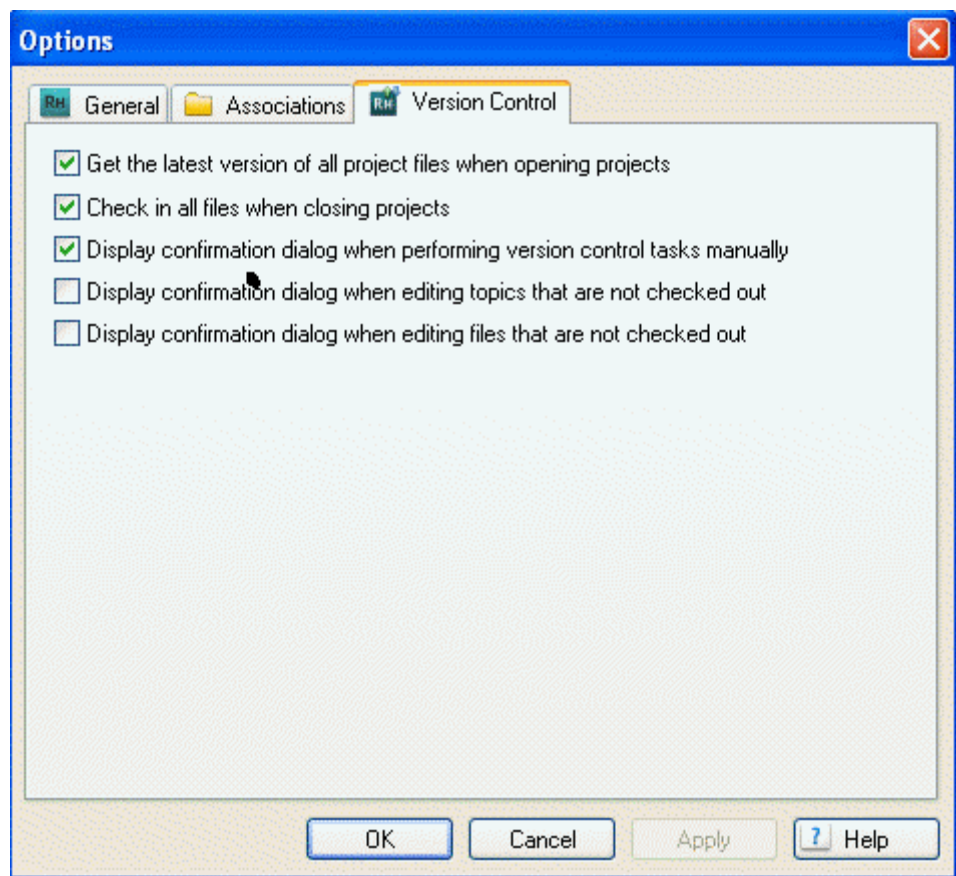
Displaying the Source Control Toolbar

Unable to specify options or comments when running DesignSync commands

DesignSync for Visual Studio User's Guide

If you are unable to specify command options or comment when performing DesignSync actions from within DSVS or using the SCC interface, you may need to adjust the options of Visual Studio, or your application to display confirmation, optional, or advanced dialog boxes for revision control operations.

This option is generally located under Tools | Options on the tab that control revision control operations.



Additional Information

Module Context Field

This field is only available when a module folder is being operated on, when a module context is required. Specifying a module context enables the operation to be run on a workspace folder that is below multiple modules. Or on a sub-folder of a module on a server.

The default module context value is an empty string. You can type the server URL of a module, or select an existing client module to restrict the scope of the operation. Or select from among the module instances for the folder being operated on, which are listed alphabetically in the pull-down. However, if more than one folder is being operated on, then you can only select from the available module instances (you cannot type in a value).

For dialog boxes in which a `Browse...` button appears next the **Module context** field, you can use the `Browse...` button to navigate to and select a module on a server.

Select Vault URL Browser

Facilitates entering a vault URL by using the vault associated with a public project (as defined by your project leader) or an entry in your site or local **SyncServer lists**. You can expand the listed project or server folders to select subfolders if desired.

Command Buttons

These buttons appear on many of the forms in the DSVS integration. Not all buttons appear on all forms.

| Button | Description |
|---------------|--|
| Save Settings | <p>Click this button to save the option settings that you have selected. The saved settings are displayed the next time you bring up the dialog box, and the settings persist from one DesignSync invocation of the dialog box to the next invocation.</p> <p>Note: These settings are only saved as presets in the dialog box. If you do not open the options dialog, you will not see these settings. You can use the command defaults feature to set defaults for the command. Those defaults are used when the command is run without explicitly specifying options and are overridden by options specified from the options dialog box.</p> <p>Most saved settings apply only to future invocations of the</p> |

| | |
|--------|--|
| | same operation. However, some saved settings, such as for Exclude Filter and Key Substitution , apply to all operations that support the options. You typically want to apply some options consistently across all commands. For example, if you exclude *.log files during checkins, you likely want to exclude *.log files for all operations. |
| OK | When you click on the OK button, your settings are executed and the dialog closes. |
| Cancel | When you click on the Cancel button, the dialog closes without executing any of the settings in the dialog. |
| Help | This button invokes help information for the dialog. You can also invoke help by pressing F1 at any time. |

Getting Assistance

About DSVS

DSVS version information is displayed as an About box attached to the Source Control definition. The About box also includes contact and copyright information for Dassault Systemes and DSVS.

To Launch the About Dialog Box

1. Select the **Tools > Options >** to open the Visual Studio options dialog.
2. On the Options dialog, select **Source Control > Plug-in Settings**.
3. Click the Advanced... button to launch the **About** dialog.

Note: The About box is also displayed as part of the **Get Options** dialog box as the last tab.

Related Topics

[Get Options](#)

[About the DesignSync Visual Studio Integration](#)

Using Help

ENOVIA Synchronicity DesignSync Data Manager Product Documentation provides information you need to use the products effectively. The Online Help is delivered through WebHelp®, an HTML-based format.

Note:

Use SyncAdmin to change your default Web browser, as specified during the ENOVIA Synchronicity DesignSync Data Manager tools installation. See SyncAdmin Help for details.

When the Online Help is open, you can find information in several ways:

- Use the **Contents** tab to see the help topics organized hierarchically.
- Use the **Index** tab to access the keyword index.
- Use the **Search** tab to perform a full-text search.

Within each topic, there are the following navigation buttons:

- **Show** and **Hide**: Clicking these buttons toggles the display of the navigation (left) pane of WebHelp, which contains the Contents, Index, and Search tabs. Hiding

the navigation pane gives more screen real estate to the displayed topic. Showing the navigation pane gives you access to the Contents, Index, and Search navigation tools.

- << and >>: Clicking these buttons moves you to the previous or next topic in a series within the help system.

You can also use your browser navigation aids, such as the **Back** and **Forward** buttons, to navigate the help system.

Related Topics

Getting a Printable Version of Help

Getting a Printable Version of Help

The *DesignSync Visual Studio User's Guide* is available in book format from the ENOVIA Documentation CD or the DSDocumentationPortal_Server installation available on the 3ds support website (<http://media.3ds.com/support/progdir/>). The content of the book is identical to that of the help system. Use the book format when you want to print the documentation, otherwise the help format is recommended so you can take advantage of the extensive hyperlinks available in the DesignSync Help.

You must have Adobe® Acrobat® Reader™ Version 8 or later installed to view the documentation. You can download Acrobat Reader from the Adobe web site.

Contacting ENOVA

For solutions to technical problems, please use the 3ds web-based support system:

<http://media.3ds.com/support/>

From the 3ds support website, you can access the Knowledge Base, General Issues, Closed Issues, New Product Features and Enhancements, and Q&A's. If you are not able to solve your problem using this information, you can submit a Service Request (SR) that will be answered by an ENOVIA Synchronicity Support Engineer.

If you are not a registered user of the 3ds support site, send email to ENOVIA Customer Support requesting an account for product support:

enovia.matrixone.help@3ds.com

Related Topics

Using Help

Index

B

Branch

- creating 46

C

Check In

- interface 17

- working offline 4

Checkout

- checkout for edit 12

- undoing a checkout 22

Command

- buttons 85

D

DesignSync Visual Studio 1, 81, 87

- enabling DesignSync Add-in 78, 82

- getting started 2

- launching SyncAdmin 37

Diff

- comparing files 32

Disconnected Mode 4

G

Get

- getting latest version 7

- operation 7

H

Help

- contacting ENOVIA 88

- printing 88

- using 87

History

- viewing 24

L

Log File

- DSVS 81

- reviewing error 82

M

Module

- displaying status 29

O

Objects

- comparing 32

- moving and renaming 38

- working with DSVS objects 2

P

Properties

interface 35

R

Reset

resetting DesignSync Add-in 82

resetting the Source Control toolbar 75

S

Source Control

changing 75

excluding from 48

icons 5

T

Tag

interface 41

Toolbar

displaying 73

Troubleshooting

about DSVS 81, 87

could not find key error 82

DSVS log files 81

reset DesignSync add-in 82

unable to specify options when running commands 82

V

Vault Browser

selecting URL 85

Vaults

setting the vault association 54, 62, 68

Versions

populating 7

properties 35