



Использование документации Altium
Последнее изменение: Rob Evans; 23.10.2018

One of the greatest strengths of an electronic authoring and editing environment is the ease with which you can create and modify a file. This capability means that ideas can be captured, explored and matured quickly, but it also means that it can be difficult to keep track of changes made to valuable files such as source code and electronic design data.

The need to keep track of changes made to a file, combined with the need for a systematic solution for managing source captured in an electronic form, has given rise to Version Control Systems (VCS). Version control systems are software tools that are not only able to maintain a history of the various versions of a file, they also support opening any revision of that file, as well comparing changes made between any two versions of the file. A VCS will generally integrate with the local Operating System (OS) by providing additional versioning functions and operations for folders and files.

Version Control Systems can be operated completely independently of the authoring and editing environment used to create a file. They offer an interface that allows you to place (or 'check') files into a central storage area called the repository, a check-out feature to copy a file from the repository to a working folder, an update feature to check back in any changes to the repository, and a method of logging information about a change. These features are also included in Altium Designer itself.

Typically a version control system will also handle the situation where a file has been checked-out and modified by multiple people, who could then check their changes in on top of each other, potentially losing someone's work from the latest revision. Dealing with this requires tools to compare versions of the file to detect differences, and an interactive tool to merge the differences back into a single version of the file, such as Altium Designer's [Collaborate, Compare and Merge](#) capabilities for PCB design documents.

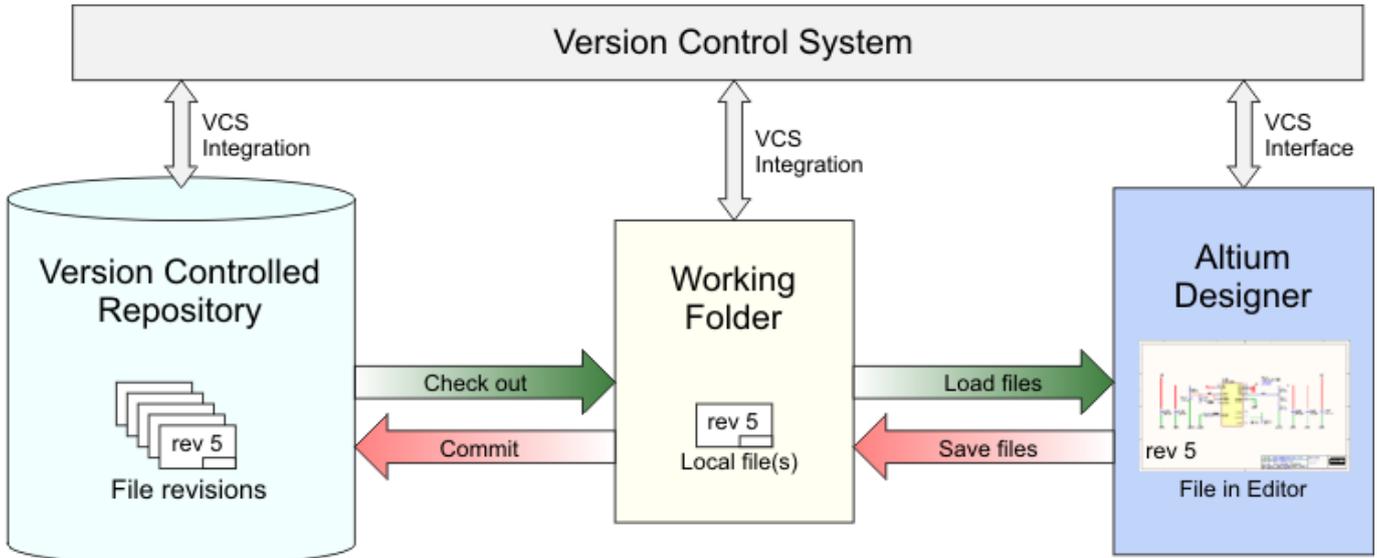
It is helpful to understand the [terminology](#) used with version control systems. Even though there are numerous systems available, they generally all use similar terms to describe their functionality.

Version Control Essentials

The basic approach to working with a version control system is to take a copy of, or 'check-out', the project files that you want to work on, from the repository into a working folder. These can then be modified and saved in Altium Designer, and when you decide it is appropriate, the updated files can be 'committed', or checked back in to the VCS repository. Check-out and check-in (commit) is performed through a version control system interface, which Altium Designer has built into its *Storage Manager* panel and *Projects* panel.

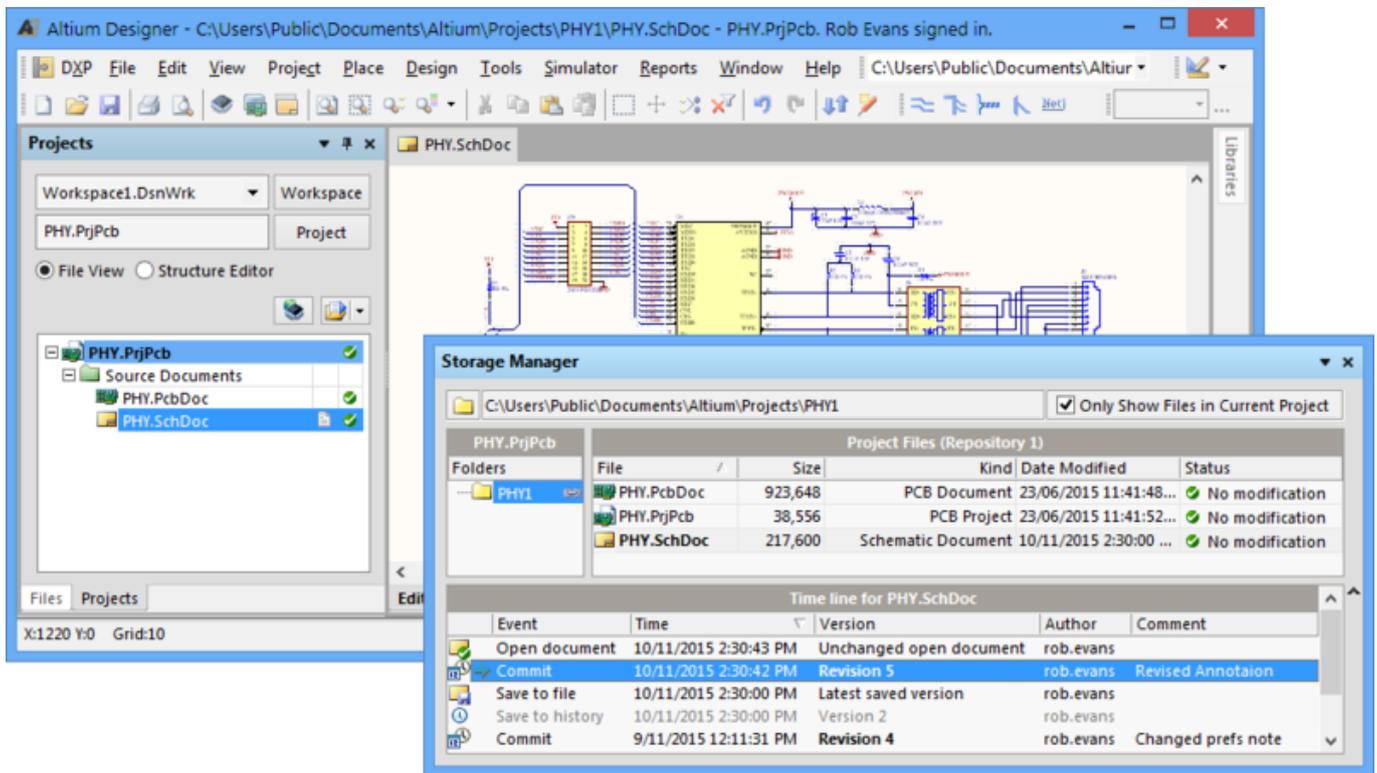
The key to how a version control system works is that it monitors the status of files that have been checked-out of the central repository. This means that the VCS knows that a file has been checked-out into your working folder, what revision was checked-out, and if it has been modified since it was

checked-out. Each VCS uses its own approach to monitoring files, refer to the specific VCS documentation if you need further information.



Simple diagram of how Altium Designer interfaces to a Version Control System (VCS)

The figure above illustrates the concept of a VCS repository that holds a sequence of design file revisions (up to revision 5), with the latest copy checked out in the working folder. When the file in the working folder is opened in the Altium Designer environment, Altium Designer recognizes that the project file is under version control, with its current version control status being shown in both the *Storage Manager* and *Projects* panel.



Altium Designer's built in Version Control interface can be accessed directly from the *Projects* panel or from the *Storage Manager* panel, where more option are available.

The right-click menu in the *Storage Manager* panel (and the *Projects* panel) allows you to a perform standard VCS actions, such as checking back in a changed file. When you right-click and select **Commit** the VCS takes a copy of the file from the working folder, increments the revision number (stored internally in the VCS), and stores the copy of the file in its repository, along with a log

message that you can enter to describe the changes made to the file. Most VCS systems do not store the entire file with every commit, instead they store the differences from the previous version.

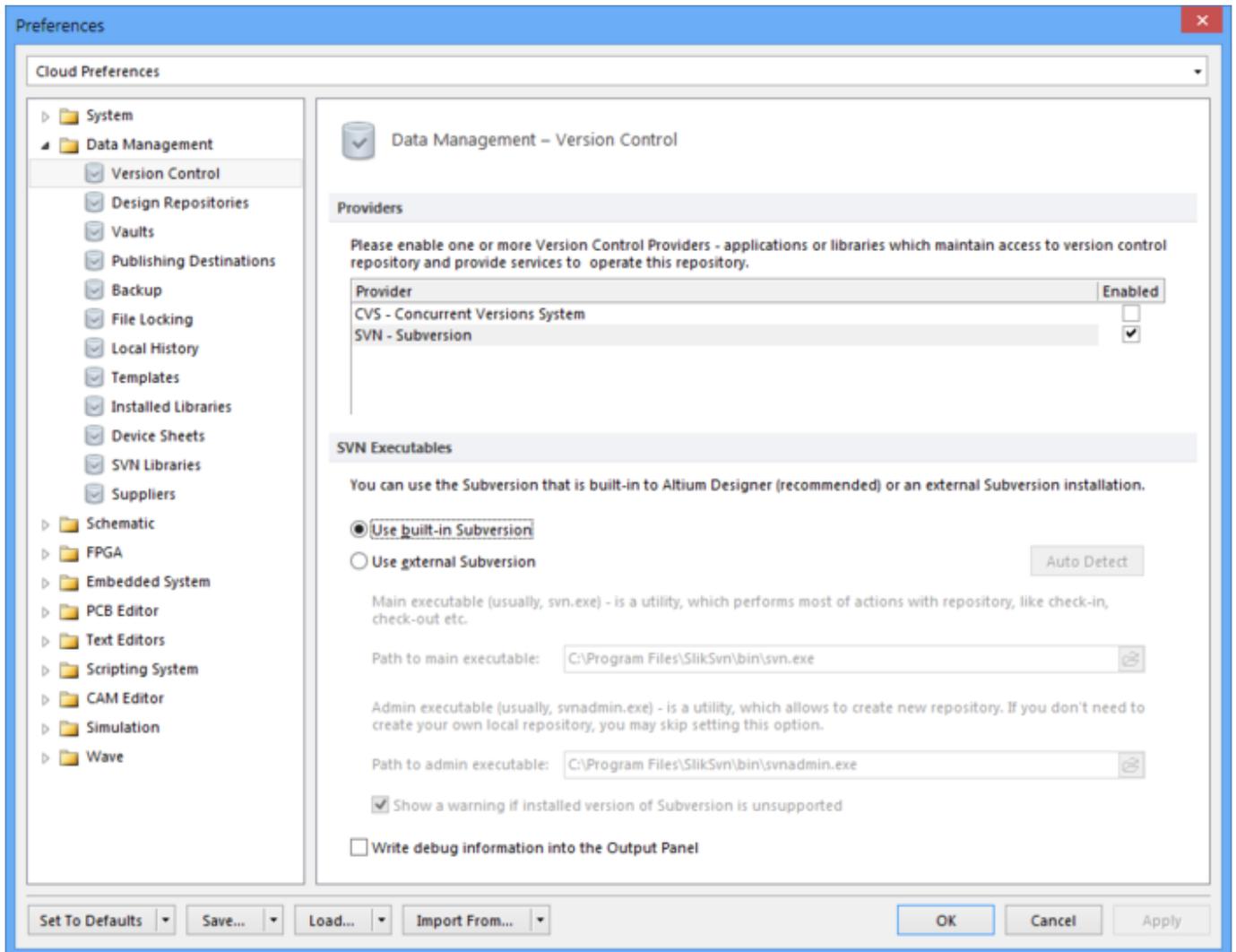
If required, external version control systems can be managed and used independently of Altium Designer with an appropriate VCS client application. In the case of Subversion, a suitable client is the [Tortoise SVN interface](#), which also provides additional VCS commands to those available in the *Storage Manager* panel.

Version Control installations

Altium Designer allows for three main approaches to interfacing with a Version Control System, all of which are configured in the [Data Management - Version Control](#) page of the *Preferences* dialog (**DXP » Preferences**). Altium Designer's interface can be configured to work with the following VCS applications:

- Altium Designer's internal copy of the Subversion (SVN) version control system.
- An external installation of the Subversion (SVN) version control system.
- An external installation of the Concurrent Versions System (CVS) version control system.
- The Altium Vault's built in Subversion (SVN) system, when using a centralized version control approach to access both vault-based and external Design Repositories – see [Vault VCS Repository](#), below.

The most common and recommended approach is making use of the SVN system included with Altium Designer, which is enabled by selecting the **Use built-in Subversion** option in *Preferences*. Note that the actual SVN system executables are located in the `\Subversion Client` folder of the Altium Designer installation.



Use the SVN application supplied with the Altium Designer installation (recommended), or direct the VCS interface to an 'external' CVS or SVN system installed on the local machine.

Alternatively, where a compliant company SVN or CVS system is already available (its [application](#) is locally installed), the *external* connection options in the Version Control preferences are used to locate and apply its main executable(s).

With Subversion (SVN), a path can also be specified to its *admin* executable, usually `svnadmin.exe`, which will allow Altium Designer to create a repository. This option can be skipped if a repository is already available, and there is no need to create more. Note that with some version control installations you may need to restart Altium Designer to register the VCS selection.

Multiple user access

There are essentially two different approaches to providing multiple team members with access to the same set of files that are under Version Control. Depending on the VCS that is in use, you can either:

- Allow multiple check-outs (referred to as the unreserved check-out model). Note that if necessary, a file can be manually locked by a user, which will prevent other users committing a new version to the repository.
- Restrict file access to one person at a time (referred to as the lock-modify-unlock, or the

exclusive check-out model)

Allowing multiple check-outs is the ideal approach as it allows different team members to continue working independently, without having to wait for someone else to check a file back in before they can work on it. However, it does require that the tools and techniques are available to resolve the inevitable situation where two people have modified the same file. When these capabilities are available, the foundation is in place for true multi user design collaboration and its associated benefits.

To support this situation, Altium Designer includes schematic and PCB comparison (or 'diff') capabilities, available through the [Storage Manager](#) and [Collaborate, Compare and Merge](#) panels. Altium Designer offers sophisticated PCB design collaboration features that allow file merge differences to be viewed and resolved within the editor, to ultimately create a new Head revision in the VCS. The advanced collaboration functionality also allows for live design collaboration between multiple users, with the ability to define and assign sections of a design as user Work Regions.

- See [Collaborative PCB Design](#) for more information on multiple user design collaboration.
- See [Using Version Control](#) for a guide to applying version control to project design in Altium Designer.

VCS Repositories

Before using version control, the project files must be recognized by both the VCS and Altium Designer as being under version control. This process can be different for the various VCS methods and applications, but it essentially entails creating and/or connecting to a design repository and adding design project files to that repository.

Design repositories are based on a database structure, and internally store information in a hierarchy of files and directories, referred to as a file tree. The actual repository that you connect can be an existing company 'central' design repository, or one you have created in an accessible location such as on the local PC or a shared network location.

Depending on the type of repository, it will be accessible via a range of protocols that include:

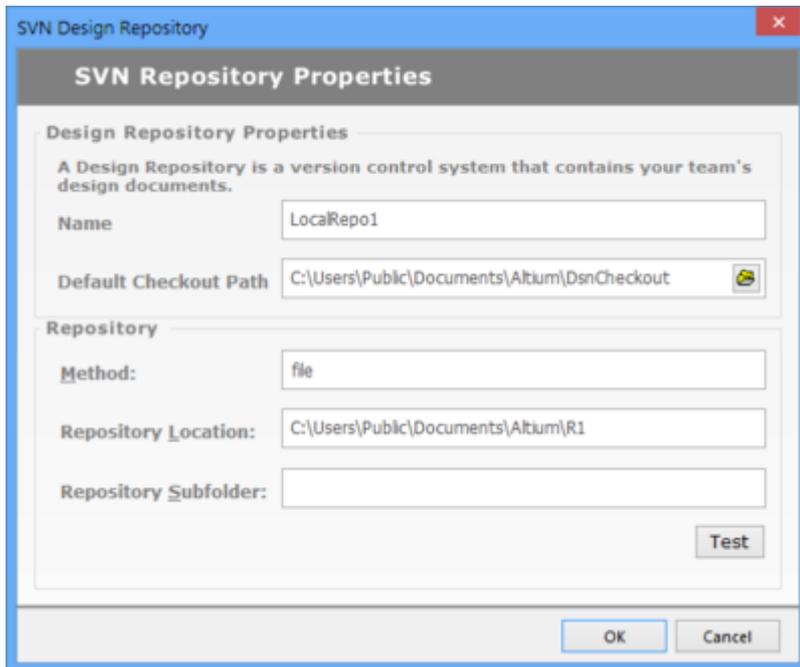
- Conventional file access, which is usually for local or network -based repositories.
- The svn protocol to a server-based repository, using plain text or TCP/IP.
- The http method to a server-based repository, generally using [WebDAV](#) over http.

VCS repositories are connected to, and created when necessary, in the [Data Management - Design Repositories](#) page of the *Preferences* dialog (**DXP » Preferences**).

- See [Securing Design Source Inside a Design Repository](#) for additional information.

Connect to a Repository

To connect to an existing SVN design repository for example, use the  button to open the *SVN Design Repository* dialog. This provides **Properties** settings that allow you to define the name of the repository connection and its target folder path for checked out design files.



Locate an existing VCS repository to be registered in Altium Designer, and therefore be available for including design files under version control.

The dialog's **Repository** options need to be configured to match the available repository's location, an optional subfolder target, and a compatible protocol method for the connection. Note that the `svn` and `http` access methods require additional information that relates to the host server and its access.

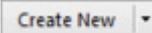


This connection process is not required for a repository that has been created within the Altium Vault. Altium Designer will automatically connect to a [Vault-based SVN repository](#) when a user signs into the Vault that hosts the repository (**DXP » Sign in to Vault**).

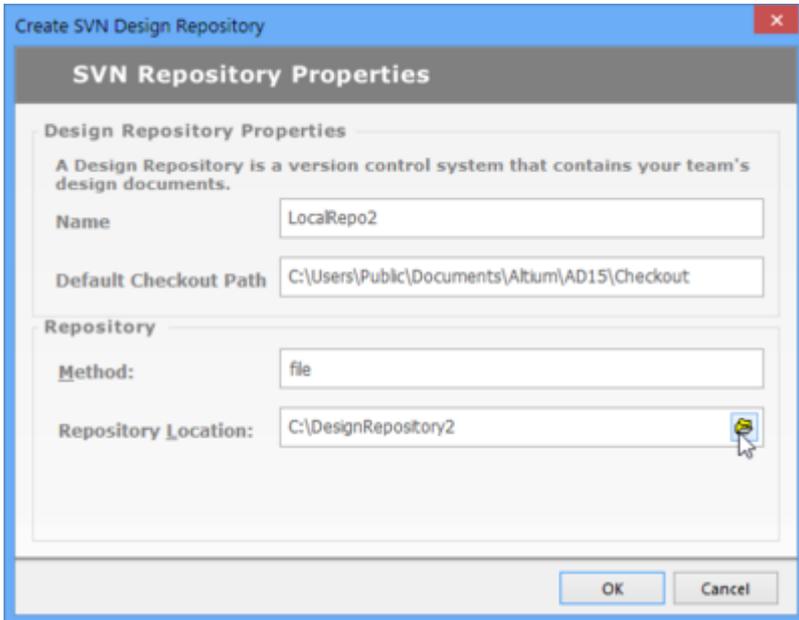
For more information, see:

- The [Data Management - Design Repositories](#) page of the *Preferences* dialog.
- The [SVN Design Repository](#) dialog page.

Create a Repository

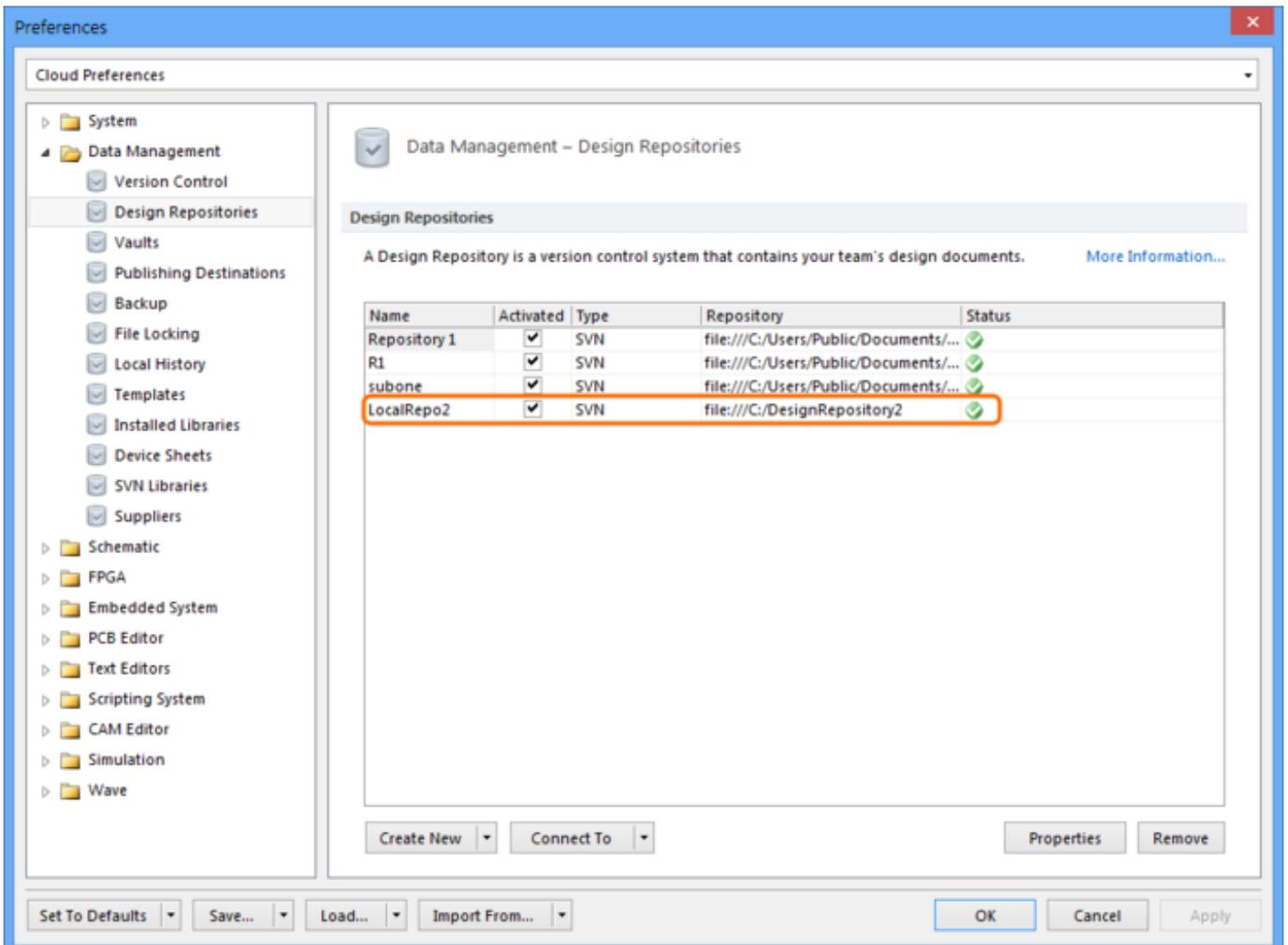
To create a local SVN design repository, go to the [Data Management - Design Repositories](#) page of the *Preferences* dialog (**DXP » Preferences**), and use the  button to open the *Create SVN Design Repository* dialog.

The dialog provides a range of configuration options that allow you to define the repository's folder location and method (connection protocol), and the local connection name and target folder path for checked out design files. Note that the `svn` and `http` access methods require additional information that relates to the host server and its access. For more information, see the above links.



Nominate or create a local folder that will be configured as a named VCS repository.

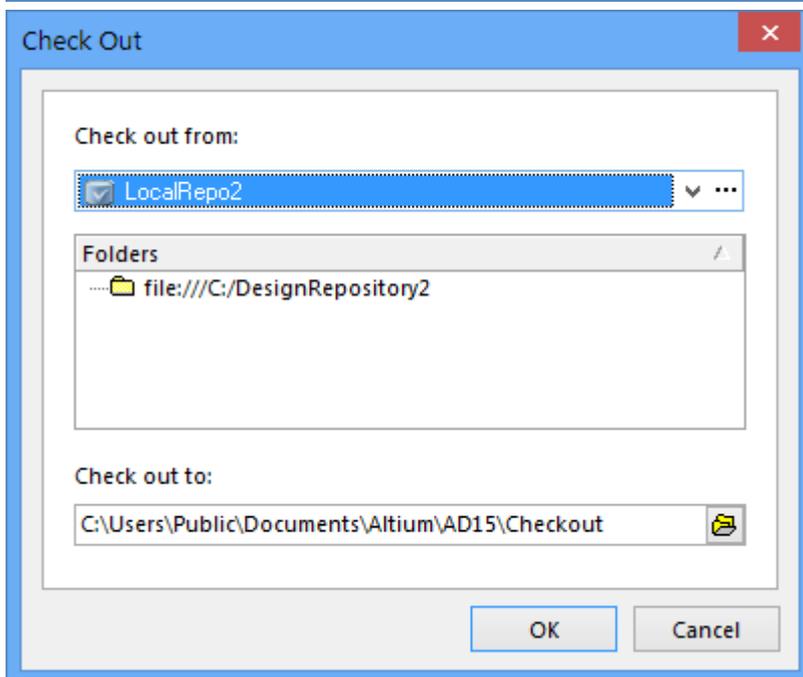
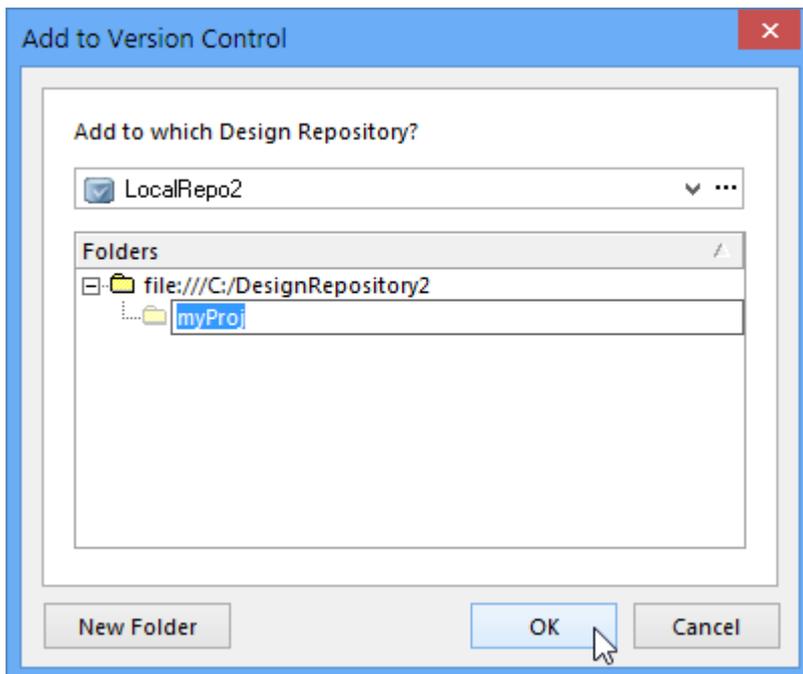
The installed VCS system will subsequently create the correct version control file structure and database in the nominated repository folder. Once created and registered, the new repository entry will be connected and listed in the Design Repositories page of the *Preferences* dialog. Click or to confirm the changes.



The process of creating a VCS repository - traditionally completed using an external (separate) VCS client - can be done directly in Altium Designer's *Preferences* dialog.

With the new or existing design repository available to Altium Designer, project design files can be added to and retrieved (checked-out) from the repository using the *Storage Manager* and *Projects* panels, and commands from the main menu. As the repository files exist under version control, all design revisions are tracked and accessible from within Altium Designer.

Use the right click context menus in the *Storage Manager* or *Projects* panels to add a complete project folder (and its constituent design files) to version control, or add individual files as required. The design files under version control in the repository can be subsequently checked out and loaded in Altium Designer using the **File » Check Out** command, and ultimately Committed back to the repository (checked in) as new revisions when the edits are complete.



Once a compatible VCS repository is accessible in Altium Designer, design files can be added to (left dialog image) and checked out from (right dialog image) the repository.

- See [Using Version Control](#) for a guide to applying version control to project design documents in Altium Designer.

Vault VCS Repository

As an alternative to conventional VCS repositories hosted locally or on the network, fully managed, a multi-user design repository can be created in the Altium Vault. Because of the Vault's tight integration with its VCS service (specifically SVN) and Altium Designer, user and client access can be centrally configured through common credentials.

This approach allows an organization to allocate multiple Vault-based design repositories to groups of users, who simply use their standard Vault sign-in credentials to access the Vault and any assigned VCS repositories. All Vault-based repositories are centrally configured through the vault's [browser-based interface](#) (see the **ADMIN » VCS** page), thereby simplifying the task of managing multiple version controlled repositories that have different client access requirements.

- See [Centralized Version Control Service](#) for more information about using Vault-based SVN Repositories.

Additional information

Version Control Terminology

[Version Control Terminology](#)

<i>VCS</i>	Version Control System: A generic term applied to any tool that is capable of managing file version history and file retrieval.
<i>Repository</i>	The master storage of all the files under version (or source) control - can also be known as the database.
<i>Database</i>	The master storage of all the files under version (or source) control - also known as the Repository in practice.
<i>Check-in</i>	To save your working copy of the file into the repository. Referred to as Commit in Altium Designer.
<i>Check-out</i>	To take a copy of a file from the VCS repository into a working folder. This is generally the the latest revision of the file, but all earlier revisions can also be checked out. Depending on the VCS, the file can be flagged as simply checked-out, or checked-out exclusively (locked).
<i>Commit</i>	Save the working copy of the file back into the repository. Referred to as Check-in in some version control systems. In Altium Designer, the normal Save command saves an edited file to the working folder, whereas Commit saves that folder file to the repository as a new revision (version).

<i>Conflict</i>	The situation when two Altium Designer users attempt to commit changes that affect the same region of the same file. These must be resolved, either using a Merge tool, manually, or by determining which version will dominate (become the new revision).
<i>Update</i>	The act of checking for and 'pulling in' changes from the repository version of a file to a working copy (the compliment of Commit, or Check-in). The process of merging in any differences requires a Merge tool, or manual updating.
<i>SVN</i>	Subversion is an open source version control system. Altium Designer has a direct SVN interface, allowing revisions to be directly tracked and accessed from its Storage Manager panel. This makes it easy to compare different revisions of schematic and PCB files.
<i>CVS</i>	The Concurrent Versions System is an open source version control system. Altium Designer has a direct CVS interface, allowing revisions to be directly tracked and accessed from its Storage Manager panel. This makes it easy to compare different revisions of schematic and PCB files.
<i>Working folder</i>	The folder where files are checked out to from the repository, so they can be worked on - also referred to as the Sandbox Folder. Files checked out from Altium Designer are automatically loaded.
<i>Sandbox</i>	The folder where files are checked out to from the repository, so they can be worked on - also referred to as the Working Folder. Files checked out from Altium Designer are automatically loaded.
<i>Log message</i>	A comment about changes made to a revision when it is checked back in (committed) to the repository. The log messages can be used as a summary of the progress of changes in a file.
<i>Working copy</i>	The 'local' copy of a file that changes are made to - normally resides in the Working Folder.
<i>Revision</i>	A committed change in the history of a file or set of files. This is the numerical reference supplied by the VCS to track the different editions (versions) it is holding of the file.
<i>Version</i>	The term version is normally used to refer to the external reference number allocated by a human to the controlled files, or their output (such as in the case of source code). Most often this is considered as a revision.
<i>Head</i>	The latest revision committed to the version control system.
<i>Base</i>	The revision in the repository that you checked out to be your local Working Copy. Also called the checked out revision.
<i>SCC</i>	Source Code Control, a software interface standard defined by Microsoft® that allows an editing application (such as Visual Studio) to interface to a version control system. The interface supports actions such as launching the VCS, checking files out, checking changed files in, logging information about the change and examining the revision history. NOTE: No longer supported by Altium Designer.
<i>Project</i>	Many version control systems support the concept of a project. A VCS project is a set of related files which can be checked-in/out as a set. The VCS may also support other project-type features, such as setting a version number on all files in a project. This is distinct from the concept of an Altium Designer project, which can be added to Version Control using the 'Add Project Folder to Version Control' command.

References

[References](#)

SVN community website: <http://subversion.apache.org/>

SVN online manual: <http://svnbook.red-bean.com/>

SVN download: <http://subversion.apache.org/packages.html>

The Tortoise SVN Client: <http://tortoisesvn.net>

CVS help: <http://www.march-hare.com/library/default.asp>

CVS manual (known as the Cederqvist, after the author): <http://cvsbook.red-bean.com/cvsbook.html>

Alternate CVS download site: <http://www.cvsnt.com>

Source Control How-to, by Eric Sink: http://www.ericSink.com/scm/source_control.html

Source URL:

[https://www.altium.com/documentation/ru/display/ADES/\(\(Version+Control+and+Design+Repositories\)\)_AD](https://www.altium.com/documentation/ru/display/ADES/((Version+Control+and+Design+Repositories))_AD)