File Management with Microsoft® Visual SourceSafe™ for Technical Writers

As technical writers become busier with increasing documentation projects, and as we try to maintain a fast pace in a rapid software development environment, it becomes increasingly important for us to stay organized. Organization not only helps technical writers on development teams remain efficient but also contributes to faster start-up times for new writers assigned to a team.

One way writers can maintain organization with their projects is through efficient file management. File management is helpful for technical writers working solo on documentation projects and vital for technical writers working within a documentation team. One great tool a writer can use to maintain an efficient file management system is Microsoft® Visual SourceSafe™ (VSS). VSS provides documentation teams, as well as their project teams, one secure, central location for managing files.

This article discusses using VSS for documentation file management. It assumes you have a basic knowledge of VSS and offers guidelines for keeping files organized within it. You can find specific information on using VSS in the product's documentation.

VSS for file management

VSS is a great tool for ensuring efficient file management. Many people view it as a tool used to store only application files; however, writers could make use of the tool to store their documentation files as well. Documentation files are, after all, a subset of the application files. They belong with the rest of the application.

File management with VSS offers technical writers the following benefits:

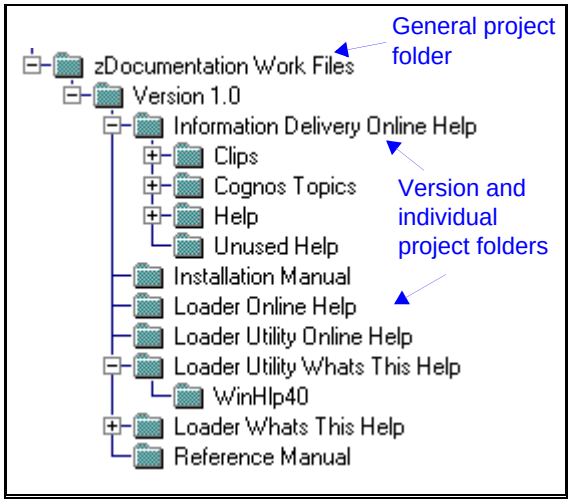
- an organized, central storage location for the project's documentation files
- a method for applying meaningful descriptions to every project folder and every documentation file
- security against multiple writers updating the same document simultaneously
- a share feature to avoid copying files to multiple locations
- a closer work environment with software developers
- added file protection with backup utilities

Organizing document files within VSS

In a current project, we use VSS to store our documentation project files. One master project folder, called *zDocumentation Work Files* (the "z" keeps the folder at the bottom of all the other application folders when sorted alphabetically), houses all of the projects documentation files and sits one level beneath the entire application project folder.

Within that documentation project folder, there are other project folders for each version of the application the documentation belongs to. We then have descriptive project folder names for each document in the set. The following examples are some of the folder names we use:

- Installation Manual
- Reference manual
- Information Delivery Online Help



We have many documentation folders in our project, but place them in a hierarchy that is organized and, more importantly, makes sense to us and future technical writers.

Within the documentation master folder (*zDocumentation Work Files*), we have a small text file that describes the structure of the project folders and documentation files in VSS. The text file, called *Documentation Read Me File*, ensures that a new technical writer can come into the team and have a reference from which to start.

We also have a master file, called *Documentation Project Master File*, that we place in the version-level folder. The master file contains all pertinent information about every document in that software version's documentation set.

These documents are designed to change as project needs dictate.

Applying descriptive comments

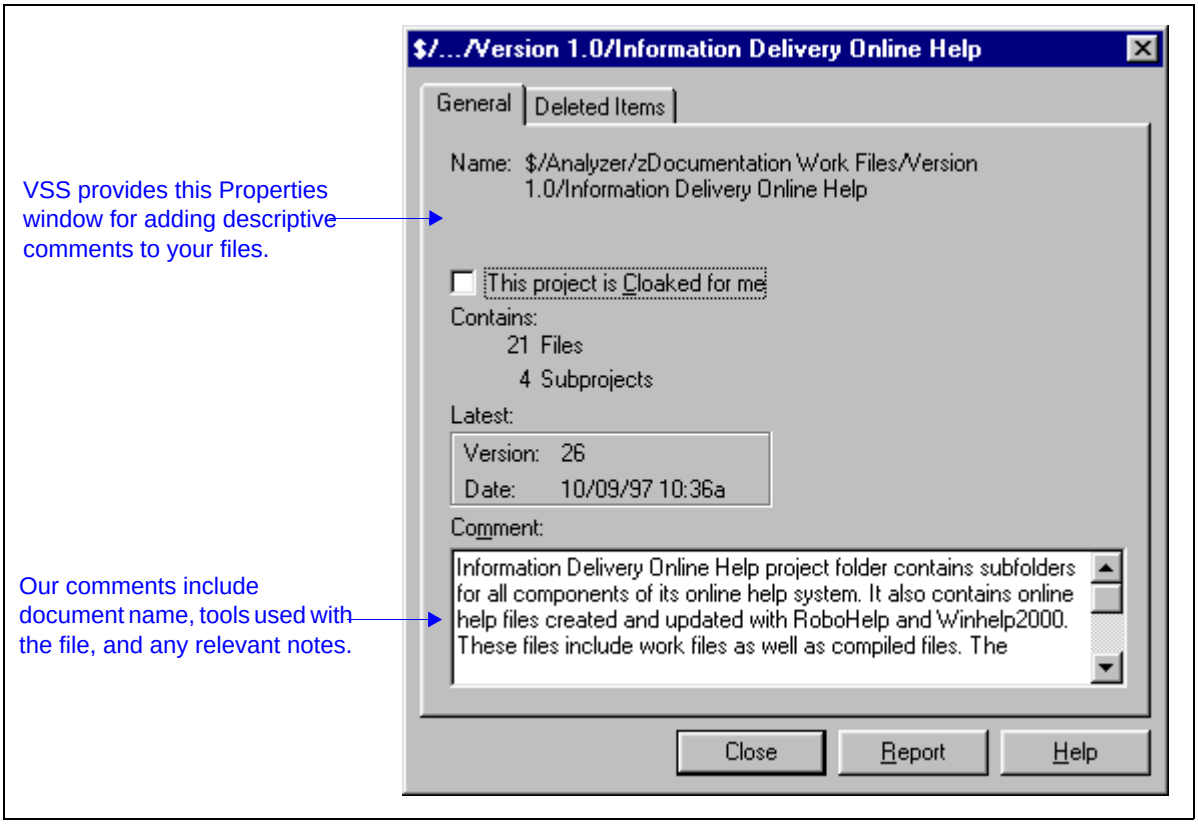
The best way for you to let people know the purpose and contents of a project folder or documentation file is to place a detailed description in its comments.

We place comments on every project folder and documentation file. The documentation files have a short comment that quickly describes the file. Project folders include more detailed information. We put the following information in the project folder comments:

- name of the document
- tools used to create or update the file
- notes pertinent to the file
- share information (I discuss sharing later in this article)

We also include this information in the documentation project master file. Although having the information only in VSS would avoid duplication, we decided a printable document with this information would be more useful for other technical writers. Therefore having the information in both VSS and the documentation project master file makes sense for everyone using these files.

Like the read me file and project master file, the information in the comments are designed to change as project needs dictate.



We use the properties feature in VSS to add descriptive comments to every project folder and source file.

Avoiding simultaneous updates

Many writers have made the mistake of duplicating work by performing an update that another writer had already performed.

The writers on our team made this mistake before we stored all of our documentation files in VSS. The mistake resulted in my reviewing every chapter of a book to ensure we didn't miss the update in any chapter.

When you use VSS to store your files, only one writer can update a file at a time by "checking out" that file. When you check out a file you download a working copy of the file from the VSS server to your computer. When you check out the file, VSS

prompts you for a description of the checkout. We type a message that describes the reason for the checkout. That way anyone can see the current changes a writer is making to a file.

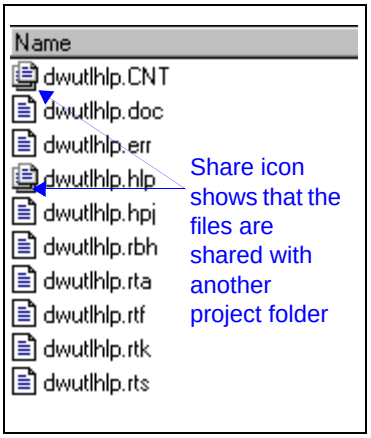
When you have finished updating the file, you check it back in. Like the checkout, the checkin prompts you for a description. We enter a description of what we actually did to the file since it is often different from what we planned to do. Then, everyone knows which updates we made to which files and there is no question as to whether or not a writer performed a specific update.

Sharing files to avoid copying to multiple locations

A useful feature with VSS is its sharing capability. Sharing offers a way to have one master file that is shared between different project folders. The best way to describe the benefits of sharing is through an example.

Our documentation project folders are organized by the way we work with them. For an online help document, we keep all of the related files together in the same folder. When we need to work on the online help file, we check out the entire folder. Then all of the current work files are together on the working computer and ready for updates.

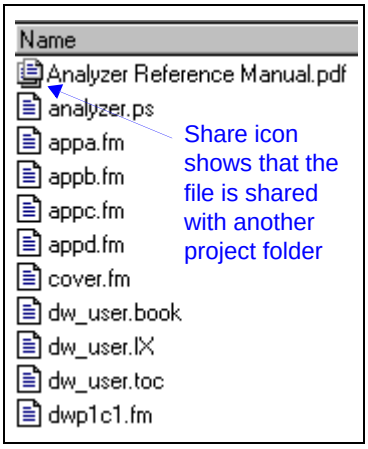
The actual help (.hlp) file and contents (.cnt) file are shared with the application project folder so that developers can build them into the application. Sharing prevents us from having to copy the compiled help and contents file to the application folder every time we recompile the help file.



Sample of shared help files as they appear in VSS.

You can benefit from sharing more than just online help files. We also share electronic documents created from FrameMaker® source files. We share the electronic document files with a project folder we call *Online Books*.

Just like with the help project folders, we check out an entire FrameMaker® project folder to generate the book and create a postscript file. The difference with our FrameMaker® folder is with re-creating the electronic document (the .pdf file). Since we do not re-create the .pdf file in the same folder we check it out to, we have to replace the original document with the new one every time we re-create it. However, once the original document is replaced, we can check the new document back in to VSS as the updated file.



Sample of shared .pdf file as it appears in VSS.

To share a file with any other project file in VSS, simply drag the original file to the new folder where you want the file to be shared. Then when you check out the working file to perform updates, you automatically check out the shared file as well. When you check the working file back in, the same changes are reflected in the shared file.

Working closely with developers

One benefit of using VSS for documentation files that is not so obvious is that doing so provides a closer working relationship with software developers on the team. Because the documentation files are stored in the same system as the application files, developers are reminded that the documentation is part of the application.

On our team, the developers always check with the writers to determine if the documentation files that are in VSS are ready whenever they want to make a new distribution set. Also, the developers know at any given time whether or not you have files checked out. On our team, if we have documentation files checked out, the developers coordinate with us to make a distribution set when our files are checked back in.

Protecting files with backups

Every night our project team backs up the entire server on which our VSS files are located. This backup ensures that we always have a recent copy of our application files should the server crash. Storing our documentation files in VSS means that our files are part of this backup. We no longer worry about backing up our own copies of the documentation files because every night they are backed up with the application files.

Drawback of VSS

There really is only one drawback in using VSS for storing documentation files. It is easy to forget to check a file back in. If a file is checked out, no one else can check it out to work with it. We have forgotten many times to check files back in, and usually it results in tracking down the writers with the files checked out and reminding them to check the files back in.

I have read about other technical writers at different companies who have complained that they run into VSS problems because of their file sizes. Personally, I have never experienced this problem.

As long as writers remember to check files back in when they are finished working with them, VSS offers an organized and safe method for storing, managing, and working with your documentation files.

WRITTEN 10/97 BY KEVIN LEWIS