Help topic not found

There is no online Help for the current topic.

Administering Visual SourceSafe

The **Visual SourceSafe administrator** performs the activities required to maintain a **Visual SourceSafe database**.   As the Visual SourceSafe Administrator, you are responsible for:

- Installing Visual SourceSafe on a central network server.
- Running Visual SourceSafe Administrator, SSADMIN.EXE.
- Configuring the Visual SourceSafe installation by using the system SRCSAFE.INI file.
- Adding new users and maintaining the **user list**.
- Keeping the TEMP folder clean.
- Analyzing the DATA folder.
- Protecting Visual SourceSafe Administrator.
- Setting up security.
- Undoing user check outs.

The User List

The **user list** in Visual SourceSafe Administrator shows every user with **access rights** to the Visual SourceSafe database.   Users whose names don't appear in this list cannot access the database.   At installation of Visual SourceSafe, this list has only two entries: Admin and Guest.

The Admin user, of which there can only be one, cannot be deleted, and the Admin name cannot be changed−it is always Admin.   The Admin user has full access rights, which also cannot be changed. Admin has the right to undo the check out of a file that another user has checked out.   Admin is the only user who can run Visual SourceSafe Administrator and modify the Visual SourceSafe Administrator user list.

The Guest user has two purposes:

- To act as a default access rights template you can use with the Copy User Rights command to create default access rights for other users.
- To provide access to the Visual SourceSafe database for occasional or first-time users.

You can delete the Guest user from the user list.   You can also change the access rights of the Guest user as you can for any other user.

See Also

Maintaining the SRCSAFE.INI File

See Also

The SRCSAFE.INI file contains initialization variables that apply to all projects in the **Visual SourceSafe database**, and sets variables that are used for every user with access to the database. Use SRCSAFE.INI to set project-wide default values for variables, which individual users can override in their SS.INI files.   Variables are used in the SRCSAFE.INI file in the same way they are used in each user's SS.INI file.   All variables are listed in Microsoft Visual SourceSafe online Help.

Some variables work best when they are stored in SRCSAFE.INI.   For example, a **shadow folder** for a project can remain current with its project only if every user's **local copy** knows about it.

Other variables can be placed in either SRCSAFE.INI or SS.INI.   For example, if your entire structure is based on shadow folders, you might set Delete_Local to Yes in the SRCSAFE.INI file.   Individual users, however, may want to override that setting if they want a complete copy of the project being worked on.

See Also

Delete_Local Initialization Variable

Cleaning the TEMP Folder

See Also

The **Visual SourceSafe administrator** is responsible for cleaning the TEMP folder pointed to by the Temp_Path initialization variable in the SRCSAFE.INI file.

Visual SourceSafe usually places files in its TEMP folder while running, and deletes them before exiting.

Some circumstances, such as having to restart, can cause files to remain in the TEMP folder – a waste of disk space.   Once every few weeks, when no users are running Visual SourceSafe or Visual SourceSafe Administrator, delete all files from the TEMP folder.

See Also

Temp_Path Initialization Variable

Analyzing the DATA Folder

The Visual SourceSafe DATA folder contains the database of all your files and **projects**.   Although every precaution has been taken to ensure the integrity of all Visual SourceSafe files, file corruption due to network or operating system problems is possible.   The ANALYZE.EXE utility is provided to enable you to search for and fix any corruption or errors.

▶   **To run ANALYZE.EXE**
- At the command prompt, type

   Visual SourceSafe WIN32 folder path**analyze** *Visual SourceSafe DATA folder path*

The program goes through the Visual SourceSafe database, reporting any possible problems.   If errors or corrupted files are discovered, you can fix them using the -F, -C, and -D parameters.

We recommend that you run ANALYZE twice − first with the -F option (to fix any errors found), and the second time to check if any errors remain.   If you still have errors after the first run, you can check the Microsoft Knowledge Base article Q152807 for more information.

Example

```
c:\SS5\WIN32\analyze c:\SS5\Data
```

Note   Backing up your DATA folder on a regular basis is highly recommended.

See Also

ANALYZE.EXE Utility

Backing Up Visual SourceSafe

Microsoft Technical Support Services

Visual SourceSafe Administrator Keyboard Guide

Navigation Keys

The following table shows some of the keys you can use to move around Visual SourceSafe Administrator.

▶ **To navigate Visual SourceSafe Administrator**

Press	To
UP ARROW	Move up the **user list**.
DOWN ARROW	Move down the user list.
HOME	Move to the top of the user list.
END	Move to the bottom of the user list.

Menu Shortcut Keys for Commands

Each command has a keyboard combination associated with it that enables you to execute a command with keystrokes.   Typically, you press the CTRL or ALT key and simultaneously press the underlined letter in the menu command.

Shortcut Keys for Commands

Press	To
ENTER	Edit a user in the user list.
CTRL+A	Add a user to the user list.
CTRL+D	Delete a user from the user list.
CTRL+E	Edit a user in the user list.
F1	Get online Help on a specific item.

Visual SourceSafe Security Access Rights

<u>**See Also**</u>

Visual SourceSafe has two levels of security: default security and project security.   If security is enabled, project-level security becomes available.   Visual SourceSafe does not permit file-level security.

There is only one Admin user.   The Admin user has, in addition to the full set of **<u>access rights</u>**, the right to undo the check out of a file that another user has checked out.   The Admin user is also the only user who can run Visual SourceSafe Administrator and modify the Visual SourceSafe Administrator user list.   The Admin user's name cannot be changed−it is always Admin.

Note   The first time you run Visual SourceSafe Administrator, you will see a warning message reminding you to set an administrator password.   If you do not set an Administrator password, any user will be able to run the Administrator and grant or change project access rights.

Default Security

In the default security setup, in which the Enable Project Security option on the Project Security Options tab (Tools menu) is disabled, each user can have one of two types of database security rights:

- Read-write.   This is the default.   A user has full access rights: Read, Add, Check Out and Destroy.
- Read-only.   A read-only user can view all projects in the **<u>Visual SourceSafe database</u>**, but cannot change anything.   Such a user can perform commands such as View, Get Latest Version, and Show History, but cannot perform commands such as Check Out, Check In, or Share.

▶   **To change a user's default security level**
 1 Double-click the user's name in the Visual SourceSafe Administrator window.
−Or
Click a user in the Visual SourceSafe Administrator **<u>user list</u>**, and on the User's menu, click Edit User.
 2 Click the Read-Only check box to restrict a user to read-only rights.
−Or
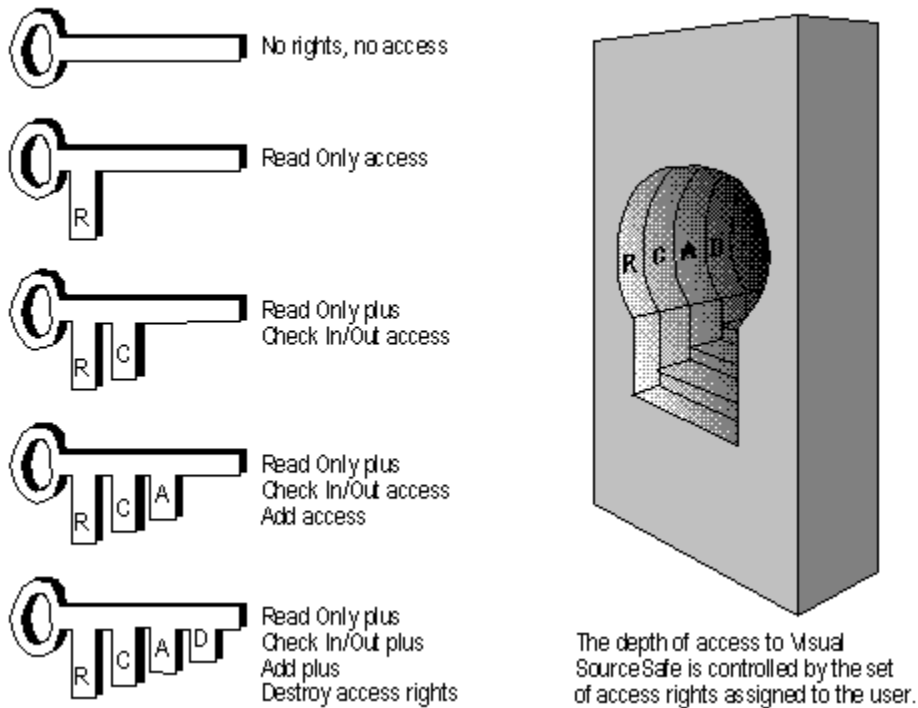Clear the Read-Only check box to grant read-write access rights.

Project Security

With security enabled, you can increase your security options.   With project security, you can control access rights for each user on a per-project basis.   Rights assigned for a **<u>parent project</u>** are automatically propagated down the project hierarchy unless you explicitly change them for a **<u>subproject</u>**.

Visual SourceSafe

No rights, no access

Read Only access

Read Only plus
Check In/Out access

Read Only plus
Check In/Out access
Add access

Read Only plus
Check In/Out plus
Add plus
Destroy access rights

The depth of access to Visual
SourceSafe is controlled by the set
of access rights assigned to the user.

▶ **To enable Visual SourceSafe project security**

1 On the Tools menu, click Options.

2 On the Project Security tab, click Enable Project Security.   Click the check boxes as appropriate to assign the user to one of the following levels of access rights:

• Read−The user can view or get files but cannot modify them.

• Check Out−Read and Check Out access rights.   The user can see and modify files.

• Add−Read, Check Out, and Add access rights.   The user can see and modify the files, and can also add and remove files in a project.

• Destroy−Read, Check Out, Add, and Destroy access rights.   The user has unlimited rights in the project, including the right to perform irreversible commands such as Destroy, Purge, and Rollback.

See Also

Rights Assignments for User Command

Rights by Project Command

Project Security Tab

Visual SourceSafe Administrator Naming Conventions

Visual SourceSafe has the following Administrator naming conventions and limitations.

- **User names** and label names can be up to 31 characters long.
- User passwords can be up to 31 characters long.
- Filenames can be up to 255 characters long.   Filenames can include spaces on operating systems that support them.
- Project paths can be up to 259 characters long (including the filename).
- Project or file **labels** can be up to 31 characters long.
- Search strings can be up to 63 characters long.
- Comments can be up to 63 characters long for the Check Out command, and 4K characters long for other comments.
- Lines in the Visual SourceSafe initialization files can be up to 511 characters long.
- Files and projects (added together) in the Visual SourceSafe database can total no more than 4 billion.
- Files and projects (added together) in any one project of the Visual SourceSafe database can total no more than 8191.
- The size of a file stored in the Visual SourceSafe database is limited to 2 gigabytes.
- The number of versions of a given file that Visual SourceSafe can store is limited to 32,767.

Most names in SourceSafe cannot begin or end with spaces or tabs.   Any characters can be used for names or labels, except the following:

- Dollar sign ($)
- At sign (@)
- Angle brackets (< >), brackets ([ ]), braces ({ }), and parentheses (( ))
- Colon (:) and semicolon (;)
- Equal sign (=)
- Caret sign (^)
- Pipe (vertical bar) (|)
- Asterisk (*)
- Exclamation point (!)
- Forward (/) and backward slash (\)
- Percent sign (%)
- Question mark (?)
- Comma (,)
- Quotation mark (single or double) (' ")
- Tab

Visual SourceSafe supports Universal Naming Convention (UNC) filenames.   If your network operating system supports UNC pathnames, you can use them anywhere you specify a path name.

Visual SourceSafe does not support the use of @ characters on Banyan Vines networks.   Banyan users can access all SourceSafe functionality by using mapped network drive letters.

Visual SourceSafe Administrator
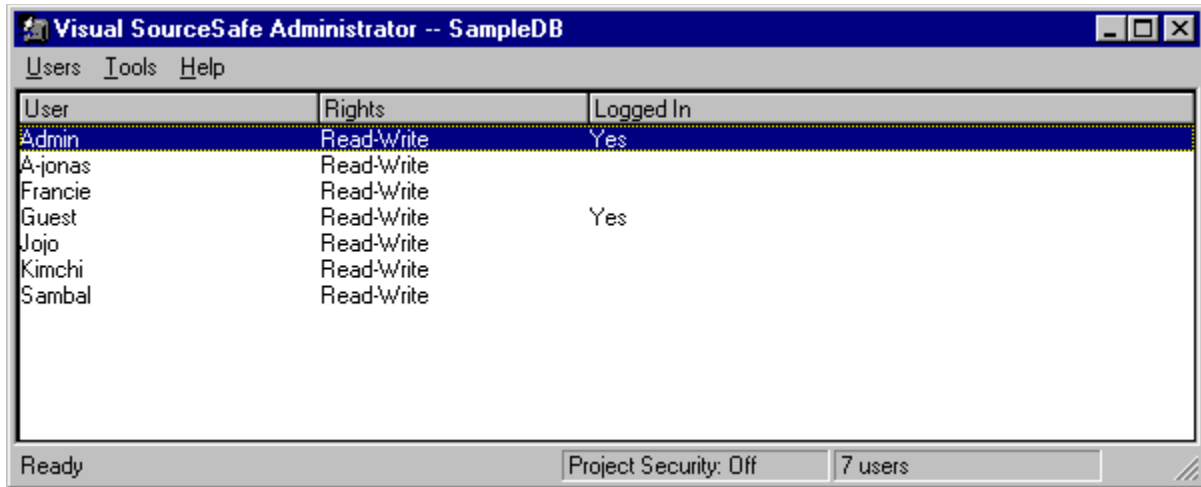
Visual SourceSafe Administrator is used for maintaining the **Visual SourceSafe database** and for setting global initialization variables.

The menu bar is at the top and provides access to Administrator commands.

Next is the user list, which lists all users, including Admin and Guest, and each user's security rights level.   You can size the columns of this display by dragging the column separator to the desired location.

At the bottom of Visual SourceSafe Administrator is the status bar, which indicates how many users (including the Guest user, but not the Admin user) are currently allowed to use Visual SourceSafe. The status bar also indicates if project security is on or off.

See Also

Visual SourceSafe Access Rights

Visual SourceSafe Naming Conventions

Backing Up Visual SourceSafe

See Also

Because your **Visual SourceSafe database** contains valuable source code and other important files, backing up Visual SourceSafe on a regular basis is highly recommended.

It is critical to back up the DATA folder and its subfolders.   If you recursively back up the DATA folder, you protect all of your files, projects, and histories.

Users' SS.INI files also contain valuable **working folders** and other setup information that should be backed up.

Note   Users can copy their SS.INI files to their hard disks; you can assign users the responsibility of performing their own backups.

See Also

Archiving Your Database

Restoring Your Archived Database

Visual SourceSafe Utilities

See Also

Several utility programs are installed in your Visual SourceSafe platform folder (WIN32, etc).   The following table lists each utility and its purpose.

Utility	Purpose
ANALYZE.EXE	Scans a **Visual SourceSafe database** for integrity and fix errors.
DDCONV.EXE	Updates a Visual SourceSafe database from an older format to the current format.
DELTA_SS.EXE	Converts Microsoft Delta projects or files to Visual SourceSafe **projects**.
MKSS.EXE	Creates an empty, old format Visual SourceSafe database. Runs automatically when Visual SourceSafe is installed for the first time.
PVCS_SS.EXE	Converts PVCS projects or files to Visual SourceSafe projects.
SS.EXE	Allows MS-DOS command-line syntax to be used in Windows.
SSARC.EXE	Archives part or all of your Visual SourceSafe database to a compressed archive file.
SSLOGIN.EXE	Finds or sets a logged-on user's username, and optionally his or her SS.INI file and **current project**, in integration scripts and macros.
SSRESTOR.EXE	Restores part or all of a Visual SourceSafe archived database file to a Visual SourceSafe database.
TESTLOCK.EXE	Tests file locks to help you determine the network **locking** mechanism.
UNLOCK.EXE	Frees database locks after a power failure or other unexpected loss.
UPDINI.EXE	Converts Visual SourceSafe .INI files that are in a format used prior to version 3.0, to the current format.

See Also

ANALYZE.EXE Utility

DDCONV.EXE Utility

DELTA_SS.EXE Utility

MKSS.EXE Utility

PVCS_SS.EXE Utility

SS.EXE Utility

SSARC.EXE Utility

SSLOGIN.EXE Utility

SSRESTOR.EXE Utility

TESTLOCK.EXE Utility

UNLOCK.EXE Utility

UPDINI.EXE Utility

PVCS_SS.EXE Utility

See Also

Converts a PVCS project or file into a Visual SourceSafe **project** or file.

Visual SourceSafe and PVCS use different internal file storage methods, so PVCS users cannot simply switch to using Visual SourceSafe; they must first extract all PVCS logfiles and enter them into Visual SourceSafe projects.

You can convert a project manually by getting a PVCS file and then adding it to Visual SourceSafe. There are disadvantages, however: It takes a lot of time and is error prone; it does not retain file histories and comments−only the latest version of each file is converted.

The PVCS_SS.EXE utility provides an automatic solution.   It converts PVCS logfiles into Visual SourceSafe files, with histories and comments intact.

Note   Only the **Visual SourceSafe administrator** can run the PVCS_SS.EXE utility.

−   **To prepare to run the PVCS_SS.EXE utility**
1 Back up your PVCS project files and **Visual SourceSafe database**.
2 Make sure both PVCS and Visual SourceSafe are fully installed on your system.

The PVCS_SS.EXE utility does not replace the installation process for Visual SourceSafe; it is run after Visual SourceSafe installation.

3 **Log on** to Visual SourceSafe as the Admin user.

The best way to do this is by setting environment (operating system) variables at the command prompt.   SSUSER and SSPWD are environment variables that indicate your username and password, respectively.   For example:

```
Set ssuser = admin
Set sspwd = hoof
```
By default, the Visual SourceSafe administrator is not assigned a password; specify a password only if you have assigned Admin a password.

4 Make sure that the PVCS folder (where executable files such as GET.EXE and VLOG.EXE are located), and the Visual SourceSafe folder, are both on your MS-DOS path.   While adding these folders to the path is not required, it makes running PVCS_SS.EXE easier.

You are now ready to run the PVCS_SS.EXE utility and convert all your files.

−   **To run PVCS_SS.EXE**
1 Change to the folder where the PVCS logfiles are located.

PVCS logfiles store all your data, comments, histories, and so on.   In general, they are named with the extensions that your filenames use, terminated with a V, so that TEST.C becomes TEST.C_V, and TEST.PAS becomes TEST.PAV.   The PVCS configuration setting LOGSUFFIX can, however, change the extension naming of logfiles.

2 Type **PVCS_SS** <*PVCS logfiles*> <*Visual SourceSafe project*>

The destination Visual SourceSafe project name must start with a dollar sign ($).   Otherwise, this variable is assumed to be a PVCS logfile.   A few examples are given below.

```
PVCS_SS *.??V $/Project
PVCS_SS TEST.C_V TEST.H_V $/Test
PVCS_SS *.PAV TEST.TXV $/
```
3 Repeat steps 1 and 2 for all PVCS logfiles you want to convert.

Remarks

- Filenames−The filename in Visual SourceSafe is not the name of the PVCS logfile; it is the name of the PVCS work file, that is, the real name of the file.   For example, TEST.C_V becomes TEST.C in Visual SourceSafe.

- Project–If the Visual SourceSafe project you specify exists, the converted files are added to it.   If the project does not exist, it is created.   If you do not specify a project, the **current project** for the Visual SourceSafe Admin user is used.

  - Histories and Version Numbers–If a file has gone through ten revisions in PVCS, there are ten versions of the file in Visual SourceSafe.   However, the numbering systems used by the two programs are different.   PVCS starts off with 1.0, followed by 1.1, 1.2, 1.3, and so forth.   Visual SourceSafe uses whole numbers: 1, 2, 3, and so on.   Therefore, what was 1.3 in PVCS becomes version 4 in Visual SourceSafe.

- Reporting and Logging–While the PVCS_SS.EXE utility is running, it displays all files that it is converting and all branches that it is ignoring.   This is logged in a file called PVCS_SS.LOG in the folder in which Visual SourceSafe is installed.

  - Null Revisions–A null revision in PVCS is a revision of a file that encodes no changes to the file contents.   PVCS_SS.EXE determines how to treat these revisions based on the Update_No_Change initialization variable in the Admin user's SS.INI file.   If the Update_No_Change variable is not set, PVCS_SS.EXE assumes the default value, Update, and converts all null revisions.

- **Branching**– Both PVCS and Visual SourceSafe support branches, but PVCS_SS.EXE does not convert PVCS branches to Visual SourceSafe branches.   You must manually convert the branches you want to keep.   PVCS_SS.EXE assists with this process by reporting and logging in PVCS_SS.LOG all the branches that it passes over.

See Also

PVCS_SS Command Line

Update_No_Change Initialization Variable

PVCS_SS (Command Line)

<u>**See Also**</u>

The following provides information on the PVCS command line.

Syntax

PVCS_SS *logfile* [*more logfiles*] [*project*] [**-P***PVCS folder*] [**-S***Visual SourceSafe folder*] [**-U***password*] [**-?**]

The order of the parameters is not important.   Options are recognized by a hyphen (-) or forward slash (/); the project is recognized by a dollar sign ($); any parameter without a hyphen, forward slash, or dollar sign is read as a PVCS logfile.

Remarks

- Logfiles—PVCS logfiles can be specified as individual filenames (TEST.C_V) or as masks (*.??V). You can specify as many as you want during one running of PVCS_SS, but you must specify at least one.   They can be in other folders, although the current drive and folder are used as the default.   Whatever logfiles you pass to PVCS_SS, and however you specify them, you should be able to pass the exact same parameters to the PVCS program, VLOG.EXE.

- Project—The project is optional, because PVCS_SS.EXE assumes the default is the <u>**current project**</u> for the Admin user.   If you do specify a project, use Visual SourceSafe project syntax.   For example, $/TEST/SUBPROJECT.   PVCS_SS.EXE attempts to create the project you specify, if it does not already exist.

- PVCS Folder—The PVCS folder stores PVCS executable files such as GET.EXE and VLOG.EXE. If you have the PVCS folder on your MS-DOS path, it is not necessary to specify the -P option.   If the PVCS folder is not on your path, use -P to enable PVCS_SS.EXE to locate PVCS.   For example:

  `-pd:\pvcs`

- Visual SourceSafe Folder—The Visual SourceSafe folder is where the Visual SourceSafe SRCSAFE.INI is located.   If you do not have SRCSAFE.INI on your path, use -S to enable PVCS_SS.EXE to locate Visual SourceSafe.   For example:

  `-sh:\ss3`

- Password—This is the password for the <u>**Visual SourceSafe administrator**</u>.   If you are already logged on to Visual SourceSafe as Admin, you do not have to specify the password.   (Use Whoami on the command line to determine if you are logged on as Admin.)   If you are not logged on as Admin, use -U to give the Admin password.   For example:

  `-uAdmin,R2D2nC3P0`

  By default, the Admin user has no password.   If you have not assigned one, you can skip this switch.

- -? (or -H)—If you use this option, PVCS_SS.EXE displays a description of its command line.   No files are converted.

See Also

PVCS_SS.EXE Utility

DELTA_SS.EXE Utility

Extracts files from Microsoft Delta libraries and converts them to Visual SourceSafe internal format.

If you have projects saved in Delta, you can convert them to Visual SourceSafe.   The Delta files themselves are not changed.   You can convert entire Delta projects or individual Delta files.   After the conversion, all of the files reside in a Visual SourceSafe **project** of the same name.

If you want to convert one or only a few files from Delta to Visual SourceSafe, you must run the Delta_SS.EXE utility for each file.   You can also use **wildcard characters** to match multiple files in a single project.

The file to be converted must not be locked or have "broken" status.   To prevent inconsistent versions of files when multiple files are being converted, make sure the project is not in active use. You can use the Lock Project command in Delta to deactivate the project while you are converting it.

DELTA_SS.EXE requires the current version of Visual SourceSafe and installation of the Delta command-line tools for Delta version 1.0.   It also requires sufficient free disk space on the Visual SourceSafe destination drive to accommodate the new files: Approximately two times the disk space used by your Delta log files is required for working space and for the new files.

Note   Before using DELTA_SS.EXE, be sure to back up your Delta project files and your **Visual SourceSafe database**.

−   **To convert a single file or folder**
- On the command-line, type **Delta_SS** [**-V**] **-D** Delta project location **-P** Delta project name **-S** Visual SourceSafe location **-U** Visual SourceSafe password [file | folder] [Visual SourceSafe project name].

    For example:
    ```
    Delta_SS -v -d c:\Delta -p ToolVer1 -s c:\SS -u drowssap FILE1.TXT
    ```

    This provides a verbose report of the conversion of FILE1.TXT from Delta to Visual SourceSafe. Because no Visual SourceSafe project was specified, the converted file is placed in a project with the same name as the Delta project.

−   **To convert an entire project**
- On the command-line, type **Delta_SS** [**-V**] [**-R**] **-D** Delta project location **-P** Delta project name **-S** Visual SourceSafe Project **-U** Visual SourceSafe password [Visual SourceSafe project name].

    For example:
    ```
    Delta_SS -r -d c:\Delta -p ToolVer1 -s c:\SS -u drowssap
    ```

    This converts the TOOLVER1 project (and recursively, all its **subprojects**) from Delta to Visual SourceSafe.   This conversion includes all history for each file.

Remarks

Only the **Visual SourceSafe administrator** can run the DELTA_SS.EXE utility.   The -U option is used to specify the Visual SourceSafe administrator password, if one exists.

If you don't supply the Visual SourceSafe database location on the command line with the -S option, DELTA_SS.EXE determines the Visual SourceSafe location as follows:

- If the SSDIR environment (operating system) variable is defined, then the specified location is used.
- If the parent folder to the folder containing DELTA_SS.EXE contains SRCSAFE.INI, then that folder is used.
- If the current folder contains SRCSAFE.INI, then the current folder is used.
- If none of the above is found, DELTA_SS.EXE terminates with an error message.

The -R option converts all projects under the specified Delta project recursively.

The -P option specifying the Visual SourceSafe project name is not required.   If it is not specified, DELTA_SS.EXE uses the Delta project name as the default.   (The Delta project name must be specified with the -D option.)

Either -? or -H (from a Windows command line) can be used to get a listing of the DELTA_SS.EXE syntax.

After a successful conversion, the file(s) are placed into the named Visual SourceSafe project.   If the destination project does not exist, it is created during the conversion.   A log file (called DELTA_SS.LOG) describing the conversion is created and placed in the folder in which Visual SourceSafe is installed.

DELTA_SS.EXE saves the entire **history** of each selected file by default.   If you want to convert only the latest version, you can retrieve that version from Delta and add it to Visual SourceSafe manually using Visual SourceSafe's Add Files command.

Note   You must specify the location of the source Delta master project folder.   DELTA_SS.EXE does not operate on checked out Delta files in your working folder.

SSLOGIN.EXE Utility

Used by integration scripts and macros to find the logged-on user's username, and optionally to get .INI file and current project information for that user.   If you are running SSINIT.EXE, SSLOGIN.EXE can also be used to set the logon name stored by SSINIT.EXE.

Syntax

SSLOGIN [**-exitcode**] [**-E**] [**-O***filename*] [*username*] [*password*]

The following table describes command-line options available with this command.

Option	Description
-exitcode	Display a line of the format "Exit code: 0" so that calling programs can determine if SSLOGIN.EXE ran successfully. A zero indicates SSLOGIN.EXE ran successfully; anything else indicates it did not.
-E	Display the user's **current project** and path to their .INI file in addition to the username.
-Ofilename	Output the information to the specified file instead of to the screen.
username	Show information for the specified user. If the user does not exist, a non-zero exit code is displayed. If no username is specified, the current Visual SourceSafe user is used. If SSINIT is running, specifying a username passes that username to SSINIT, effectively logging in a new user.
password	Validate the user.

SS.EXE Utility

You can use the SS.EXE utility to integrate other applications with Visual SourceSafe by creating batch files or by calling SS.EXE commands from your applciation.   SS.EXE uses the following syntax.

Syntax

SS *command name <item(s) to operate on>* [*switches*]

Example

Issues a Visual SourceSafe Get Latest Version command on the file FILENAME.TXT:

```
ss Get filename.txt
```

All commands in the command set are described in Visual SourceSafe Help.

MKSS.EXE Utility

Creates an empty **Visual SourceSafe database** in an old format.   MKSS.EXE is installed in the ADMIN subfolder by default.

– **To create a new Visual SourceSafe database**
1 Create a folder.
2 Run the MKSS.EXE utility on the folder.
3 Run the DDCONV.EXE utility on the folder.

Syntax

MKSS *folder*

See Also

DDCONV.EXE Utility

DDCONV.EXE Utility

Updates a Visual SourceSafe database from an earlier format to the current format.

DDCONV.EXE is primarily used by installation scripts to make sure the format of an installed Visual SourceSafe database is correct after an update.   DDCONV.EXE is also one step in the process of creating a new Visual SourceSafe database with the MKSS.EXE utility.

If, during installation, Visual SourceSafe cannot find all of the installed database, the **Visual SourceSafe administrator** may have to run this utility manually.   DDCONV.EXE can also be used with the -S option to rebuild the status file.

Syntax

DDCONV [**-S**] [**-M***SSpath*]|[*data path*]

The following table describes command-line options available with this command.

Option	Description
-MSSpath	The path of a SRCSAFE.INI file from which DDCONV.EXE can determine the data folder.
Datapath	The path to a data folder to convert.
-S	Rebuild the Visual SourceSafe check out status index file.

See Also

MKSS.EXE Utility

ANALYZE.EXE Utility

<u>See Also</u>

Scans a **<u>Visual SourceSafe database</u>** for integrity and/or errors, displays them and, if set to do so, fixes them.   **<u>Visual SourceSafe administrators</u>** should run ANALYZE.EXE every week to verify that there are no problems in the Visual SourceSafe database.

Administrators can use the 32-bit command-line utility to repair version 4.0 Visual SourceSafe databases (and later).   For example, a database problem exists when you select a file that you know exists in the Visual SourceSafe database, use a command, such as GET, and receive the error message, "File does not exist."

Administrators can run ANALYZE manually from the command line, or as part of a script for unattended operation, to analyze or fix the problem.   The ANALYZE utility is in the WIN32 folder under the folder in which you installed Visual SourceSafe.folderfolder

We recommend that you run ANALYZE twice − first with the -F option (to fix any errors found), and the second time to check if any errors remain.   If you still have errors after the first run, you can check the <u>Microsoft Knowledge Base</u> article Q152807 for more information.

Syntax

ANALYZE [-**B<folder>**] [-**C**] [-**D**] [-**F**] [-**I-**] [-**V**] [-**X**] [-**?**] [-**H**] *<Full Database Path | @Response File |*
    List of Files>

Option	Description
-B<folder>	Specify a backup folder.
- C	Clean up and compress the database. This option can be slow.
- D	Delete any unused files.
- F	Attempt to repair any inconsistencies and corruption that are detected.
- I-	Continue without stopping for user input (allowing unattended operation). By default, ANALYZE leaves the results window on display.
- V*n*	Specify the verbosity of output. -V1 displays only critical errors, -V2 displays critical and significant errors, -V3 displays all errors, and -V4 shows all errors and warnings. The -V4 option shows its output in the lower pane of the Analyze Results window and contains detailed information on what the utility attempted and accomplished. The default is -V1. -V (without a number) is equivalent to -V4.
- X	Directs analyze to not lock files as it processes. Allows analyze to run against a live database. This option cannot be used with the -c, -d, or -f options and use of this option is not recommended unless absolutely necessary.
Full Database Path	Path of the Visual SourceSafe data folder to analyze. This parameter must be the first non-switch parameter. A data path, list of files, or response file must be present on the command line.
@Response File	File produced by running analyze, containing listing of problem files, named ANALYZE.BAD by default. Using this file as input directs ANALYZE to work on fewer files, thereby improving performance.
List of Files	List of files to analyze. List can include only log files, or

complete data paths, such as C:\VSS\DATA\A\AAAAAAAA.

-?, - H   Displays usage message containing descriptions of valid
                            options.

Order of options is not important, but the full database path, response file, or list of files must be included on the command line as the first non-option item.

Examples

```
ANALYZE -C -D -F -V4 C:\VSS\DATA
```

Analyzes and repairs the complete database and generates verbose output.

```
ANALYZE -BC:\TEMP C:\VSS\DATA
```

Analyzes the database, but does not fix it.   Creates a list of corrupted files in C:\TEMP.

```
ANALYZE -F -C -D @C:\TEMP\ANALYZE.BAD
```

Fixes any problems in the files listed in the ANALYZE.BAD file generated by the preceding example.

Remarks

The types of problems that the ANALYZE utility can correct include, among others, the following:

• Parent/child mismatch.

  Possible situations include those in which a project thinks it has a file or subproject but the child disagrees, a child thinks it has a parent but the parent disagrees, or child and parent don't reference each other but the child counts are off.

• Corrupt database files.

  These files include the following: NAMES.DAT (stores long filenames longer than 34 characters), RIGHTS.DAT (stores relationships between users and project rights), STATUS.DAT (stores checkout status for files), UM.DAT (stores all users in a Visual SourceSafe database) , and VERSION.DAT (stores the version of Visual SourceSafe).

• Removing unused items from the database.

Administrators should run this utility on a regular basis to verify that no problems exist in their Visual SourceSafe databases.   All results of Analyze/Repair are logged to the file, ANALYZE.LOG, which is placed in the \VSS\DATA\BACKUP folder and is replaced each time you run ANALYZE.   When you analyze the database, ANALYZE creates a file listing all corrupted files.   This file is called ANALYZE.BAD and is also placed in the \VSS\DATA\BACKUP folder.   In addition, the original versions of files that have been repaired are placed in the \VSS\DATA\BACKUP folder.

Under normal conditions, all users should be logged out of Visual SourceSafe before Administrators run ANALYZE.   Users must be logged out to run ANALYZE with the -   F option to repair problems. ANALYZE does not run if any users are logged into Visual SourceSafe unless the -X option is supplied.

ANALYZE writes output to a separate window it creates, Analyze Results.   This window may contain two panes.   You must close any open results windows before running ANALYZE another time.

The top (or only) pane contains summary information.   For example:

```
Analyze Version 4.00[Build 1110]
Database analysis in progress.
The Parent Project for item acctapp is corrupt and the link to the child
file has been lost until the project is recovered.
Analysis complete.
```

This is the information provided in the output window when you run ANALYZE using the -V or -V4 option for verbose output.   When you use the -V option, the information provided is written to the bottom pane of the Analyze Results window.   The following example shows verbose output:

```
Started analyzing user management system.
Successfully completed analyzing the user management system.
Rebuilding corrupted project for item aaaaaaaa.
Building the project list.
Checking cross file relationships.
Checking parent/child relationships.
Validating the Security System.
Writing a new copy of 'c:\vss\data\a\aaaaaaaa'.
The count of children of item 'aaaaaaaa', as given in its header, does not
match the number of children found on disk.  The count will be adjusted.
The count of sub-projects of item 'aaaaaaaa', as given in its header, does
not match the number of sub-projects found on disk.  The count will be
adjusted.
```

Note that the actual output in the Analyze Results window does not wrap; it scrolls within the pane.

When the RIGHTS.DAT database file is rebuilt by the ANALYZE utility, it turns on project security, even if security was disabled before the rebuild.   As a result, only the Administrator can log into Visual SourceSafe after the rebuild of RIGHTS.DAT.   Therefore, the Administrator must then run the SSADMIN program to turn off Project Security or re-assign all user rights.

ANALYZE Utility See Also

Analyze Utility Error Messages

Sample ANALYZE Log File

TESTLOCK.EXE Utility

See Also

Determines whether your network supports native **locking**.   TESTLOCK.EXE is installed in the platform-specific (i.e.   WIN32) subfolder by default.

The locking system is set in the SRCSAFE.INI file.   The primary variable is Lock_Mode, which can be set to either Native or Lockfile.   The default is Native.

−   **To determine the locking system in use on your network**

 1 Install Visual SourceSafe to a drive accessible from two or more computers.

 2 From one of the computers, go to the WIN32 folder and run:

```
TESTLOCK.EXE <folder name>
```
TESTLOCK.EXE locks a file, and displays the message "Press any key to continue."   Do not press a key yet; leave the program running, and go on to step 3.

If TESTLOCK.EXE fails and does not display the message, go to the specified folder and delete the file(s) TESTLOCK.*, then try again.   Make sure not to delete anything else.

 3 From another computer, run TESTLOCK.EXE again, providing the same folder path as before.

TESTLOCK.EXE attempts to lock the same records that the first computer has locked, and report the results.   If the lock fails, your network file system supports native locking.   If the lock succeeds, you must use the Lockfile variable setting.

 4 Press a key on both computers to end the program.

 5 If native locking is supported, repeat steps 2 through 4 on various computers on your network, testing each platform combination to make certain the necessary features are supported across your entire network.

To save time, you may leave TESTLOCK.EXE running on one computer, and repeat step 3 on every other computer, then switch to a different "first" computer and repeat the pattern, and so on.

See Also

Lock_Mode Initialization Variable

UNLOCK.EXE Utility

Clears orphaned locks by specifying the lock (from the error message) on the command line.

Visual SourceSafe uses different mechanisms to implement mutual exclusion within the database. (This has nothing to do with checking out files).   If Visual SourceSafe stops unexpectedly, it may leave some of these mutual exclusion locks behind.   The next time Visual SourceSafe tries to access the protected records, it reports an error.

Syntax

UNLOCK *lock*

Lock is the lock name displayed in the "time-out locking" error message.

See Also

TESTLOCK.EXE Utility

UPDINI.EXE Utility

Converts Visual SourceSafe's .INI files from an earlier version to the current version.

Visual SourceSafe's configuration is stored in a set of .INI files.   UPDINI.EXE is primarily used by installation scripts so that all .INI files are in the appropriate format after an upgrade.

This utility may also be necessary if you have moved your earlier version .INI files from their default installation location.   When you install Visual SourceSafe 3.1 or later as an upgrade of an earlier version of SourceSafe, the Setup program runs the UPDINI.EXE utility to convert your .INI files to the new format.   If your .INI files are in their default locations, Setup converts them automatically. However, if you have moved your SS.INI file, Setup may miss the file.   You then have to manually run UPDINI.EXE to update your SS.INI file after completing Visual SourceSafe installation.

Syntax

UPDINI [**-M***SSpath*]|[.*INI file*]

The following table describes command-line options available with this command.

Option	Description
-MSSpath	If this parameter is specified instead of an .INI file, UPDINI.EXE looks for the SYSTEM.INI file (the old name for SRCSAFE.INI). If it finds SYSTEM.INI, it opens it, finds all of the user .INI files, updates the contents of all .INI files, and then renames SYSTEM.INI to SRCSAFE.INI.
.INI file	Provide UPDINI.EXE with the name of an .INI file to update.

Creating a Shadow Folder

The Visual SourceSafe administrator is responsible for creating **shadow folders**, because several variables must be set in the global SRCSAFE.INI file.

A shadow folder is optional.   It is generally used in one of two situations.   In one, there may be users who want to see the files, but who do not have access to Visual SourceSafe.   A shadow folder allows them to view but not modify files.

In the other, to make sure that every user has the most recent version of files, you may want all users to compile source code out of a centralized folder, instead of in **working folders**.   In this case, shadow folders are frequently used in conjunction with the Remove Local Copy After Add Or Check In option on the Local Files tab in the SourceSafe Options dialog box, which prevents your working folder from keeping its own compilable copy of the **project**.

– **To create a shadow folder**

1 In Visual SourceSafe Administrator, on the Tools menu, click Options, and then click Shadow Folder.   Enter the appropriate data for the Set Shadow Folder For Project and Set Shadow Folder To options.   Click OK to apply the changes, or click Set Another if you have more than one shadow folder to set up.

Remarks

A common error is to place the shadow variable in a user's SS.INI file instead of SRCSAFE.INI.   If you do this, only that user's installation of Visual SourceSafe has access to the folder; when other users make changes to the project, those changes are not shadowed, and the folder falls behind the project.

When you set a shadow folder for a particular project, you are actually making an assignment for the entire project tree, unless you explicitly override that assignment for particular **subprojects**.

Initialization variables can be placed in your SRCSAFE.INI file to customize your shadow folder. These variables, like the shadow folder itself, should always be placed in SRCSAFE.INI rather than SS.INI.   You can place any of these variables before any headers (so that they apply to all projects), or under an individual project header (so that a particular setting applies only to one project).

See Also

Shadow_Folder Tab

Shadow Initialization Variable

Shadow_DOSName Initialization Variable

Shadow_EOL Initialization Variable

Shadow_Extension Initialization Variable

Shadow_ReadOnly Initialization Variable

Shadow_SetTime Initialization Variable

Installing Visual SourceSafe

When you install Visual SourceSafe, you need two components – the **Visual SourceSafe Explorer** interface and the **Visual SourceSafe database**, which holds your project files and **version** information.   Both are needed to use Visual SourceSafe.

Visual SourceSafe can be installed in one of three configurations.   You can:

- Install Visual SourceSafe on your local computer and create a database there.   This makes limited use of Visual SourceSafe's version control features, because you are the only user.

- Install Visual SourceSafe on a network server and have each user run Visual SourceSafe Explorer from the server.   This may affect performance because each user is running the Visual SourceSafe executable file over the network.

- Install Visual SourceSafe on a network server and install the appropriate Visual SourceSafe executable files on each user's local computer.

Usually, your **Visual SourceSafe administrator** sets up the Visual SourceSafe database on a server (using the Server option in the Setup program), and expects you to set up the necessary client files on your computer.   To do this, connect to the Visual SourceSafe server and find the Visual SourceSafe root directory.   Run the NETSETUP.EXE program found there to install the client components on your computer.   This requires at least 2 megabytes of disk space on each local computer.

Note   If you intend to run Visual SourceSafe together with an integrated development environment (IDE) like Microsoft Visual Basic or Microsoft Visual C++, you must perform the client setup to properly register Visual SourceSafe on your local computer.   This client setup must occur after your administrator has finished installing SourceSafe on a network server.   If you want to have both the Visual SourceSafe database and the client on the same computer, use the Server setup button to install a SourceSafe database on your local computer rather than a network server.   Then run setup again and click the Client button to register your IDE properly.

The *Microsoft Visual SourceSafe User's Guide* contains detailed information on setting up Visual SourceSafe.   Chapter 8, "Administration," provides information on Visual SourceSafe database administration responsibilities.

Enabling Keyword Expansion

Keyword expansion refers to Visual SourceSafe's ability to place certain information directly into your file to create a file header for you.

To do this, you place certain keywords, in comments so that it does not affect your code, into the text of your file.   When you add or check in the file, Visual SourceSafe looks for these keywords, and places the relevant information after them.

This topic describes the traditional way Visual SourceSafe expands keywords.   With version 5.0, you now can also expand keywords in a different way that makes **HTML file keywords** much more useful.

SourceSafe Keywords

The following table is a list of all the Visual SourceSafe keywords.   Some of the keywords are redundant, either for convenience or for backward compatibility with other version-control systems, such as RCS.   Keywords are case-sensitive, so be sure to capitalize them properly:

Keyword	Description
$Archive: $	Visual SourceSafe archive file location
$Author: $	User who last changed the file
$Date: $	Date and time of last check in
$Header: $	Logfile, Revision, Date, Author
$History: $	File history, Visual SourceSafe format
$Log: $	File history, RCS format
$Logfile: $	Same as Archive
$Modtime: $	Date and time of last modification
$Revision: $	Visual SourceSafe version number
$Workfile: $	Filename
$NoKeywords: $	No keyword expansion for all keywords that follow. (The colon is optional.)
$JustDate: $	Date, without the time addendum.

A SourceSafe keyword has the following syntax:

```
$keyword: $
```

For example, you could place the following in a file:

```
$Revision: $
```

Upon check in, Visual SourceSafe could replace it with

```
$Revision: 23 $
```

The next time you check in the file, the 23 is replaced by a 24, and so on.   Without running Visual SourceSafe, you can always look at the top of your file to see, for example, what version your copy of the file is.

Enabling Keyword Expansion

Because keyword expansion requires Visual SourceSafe to scan each file for keywords, it can considerably slow the Check In and Add Files commands.   For this reason, keyword expansion is by default disabled for all files.   Your Visual SourceSafe administrator must indicate which files Visual SourceSafe should scan for keywords in the Administrator program.

1. To make keyword expansion work, use either of the methods described below:

In the Visual SourceSafe Administrator, point to Tools and click Options and in the "Expand keywords in files of type" option, enter the file types for the keyword expansion.   For example, type *.txt.

- or -

Add the following two lines to the SRCSAFE.INI file on the server:

```
Keyword_Masks = *.TXT
Expand_Keywords_Locally = YES
```

You could add them to a user's SS.INI file, but it is better to put them in the SRCSAFE.INI file on the server so that keywords are expanded when a user updates or checks in a file.

Note   The file may not have updated keywords if the user chooses to keep the file checked out on the update (that is, Keep_Checkedout=YES).   Also, the "Keyword_Masks =" line can have any extension or combination of extensions.   For example:

```
Keyword_Masks = *.C, *.CCP, *.H, *.HPP, *.BAS
```

1. Add keywords to files.   Keywords are case sensitive.

2. Format the keywords correctly.   There are two ways to format keywords in files.   The most frequently used method is to align the keywords on the left margin:

```
$Archive: $
$Author: $
$Date: $
```

When expanded, they look like this:

```
$Archive: /KeywordAlignTest/KeyAlign1.txt $
$Author: Guest $
$Date: 11/23/95 9:31a $
```

The other way is to format the keywords so that when they are expanded, the values are aligned in a column.   Note the use of two colons and the amount of space between the colons and the second dollar sign.

```
$Archive::                                           $
$Author::                                            $
$Date::                                              $
```

When expanded, they look like this:

```
$Archive:: /KeywordAlignTest/KeyAlign1.txt          $
$Author:: Guest                                     $
$Date:: 11/11/95 5:57p                              $
```

If you don't leave enough space between the colons and the dollar sign, SourceSafe truncates the value so it would look like this:

```
$Archive:: /KeywordAli$
$Author:: Guest       $
$Date:: 11/11/95 5:$
```

1. Place comment characters in front of keywords so the keywords are not compiled.   SourceSafe sets the following comment characters by default.

```
*.ASM = "; "
*.BAS = "rem "
*.BAT = "rem "
*.S = "; "
<all other file types> = "* "
```

If you want to use a different comment character in a file that has a specific file extension, add the following lines to the SRCSAFE.INI file on the server:

```
[Keyword Comments]
*.C = "/*","*/"
```

The comment character is especially important in dealing with $Log: $ and $History: $ keywords. Unlike the other keywords, $Log: $ and $History: $ keywords add more information to the file each time keywords are updated − instead of just replacing the previous information with the current information.   A comment character, as defined by the SRCSAFE.INI file, is added to the beginning of each line created by $Log: $ and $History: $ keywords.   For example, if the following is in a .txt file:

```
     *$Log: $
     *$History: $
```

and the default comment characters are in effect, it would look like this after keyword expansion took place:

```
*$Log: /KeywordAlignTest/KeyAlign1.txt $
*
* 3     11/23/95 10:47a Guest
*
* 2     11/23/95 10:45a Guest
*
* 1     11/11/95 5:57p Guest
*$History: KeyAlign1.txt $
*
* *****************  Version 3  ****************
* User: Guest      Date: 11/23/95   Time: 10:47a
* Updated in $/KeywordAlignTest
*
* *****************  Version 2  ****************
* User: Guest      Date: 11/23/95   Time: 10:45a
* Updated in $/KeywordAlignTest
*
* *****************  Version 1  ****************
* User: Guest      Date: 11/11/95   Time: 5:57p
Updated in $/KeywordAlignTest
```

Effect of Keyword Expansion

When you check in or add a file with keyword expansion enabled, Visual SourceSafe places a modified copy of your file directly into the current project.   It then immediately gets the file, which refreshes your **working folder** with the checked-in file.

The Get Latest Version command slows down the updating process.   You can avoid it by not setting the Copy Keyword Expanded-Files Into Working Folder option on the Local Files tab of the Options dialog box.   When you clear this option Visual SourceSafe still expands keywords, and places the modified file into the project.   However, it does not copy the file into your working folder.

Customizing the SS.INI and SRCSAFE.INI Files

Visual SourceSafe provides two initialization files that contain settings you can use to customize your Visual SourceSafe environment:

- SS.INI, for individual user variables.
- SRCSAFE.INI, for the **Visual SourceSafe database** global variables.

Each user has an SS.INI file that can be edited.   Only the **Visual SourceSafe administrator** should edit SRCSAFE.INI.

Important   Manually editing the initialization files is not recommended, unless you are very familiar with Visual SourceSafe initialization variables and their behavior.   Instead, you should set most initialization variables directly in Visual SourceSafe in the SourceSafe Options dialog box on the Tools menu.   You can set the SRCSAFE.INI variables in the Visual SourceSafe Administrator SourceSafe Options dialog box, on Administrator's Tools menu.   A list of all initialization file variables is available in the Visual SourceSafe Explorer Help file.

When you use the SourceSafe Options dialog box, Visual SourceSafe saves your changes in memory, but does not write the changes to the SS.INI file until you exit Visual SourceSafe. Therefore, if you start a second Visual SourceSafe session before exiting the first, your changes are not yet in effect.

You can read and modify initialization files with any text editor.   There is a limit of 511 characters on an initialization file line, although this maximum may be restricted by specific variable requirements (such as project path limitations).

Usually, SS.INI is located in the USERS subfolder of the folder in which Visual SourceSafe is installed, for example, SS\USERS\USERNAME.   You can move your SS.INI file from this location, providing you tell Visual SourceSafe of the change by editing the USERS.TXT file (see your Visual SourceSafe administrator).

The SRCSAFE.INI file is usually placed in the main SourceSafe folder on a server.

Note   Initialization variables set in SS.INI override settings in SRCSAFE.INI.

The Format

There are three kinds of lines in SS.INI and SRCSAFE.INI.

- Initialization variables, which are the functional lines, take the following format:

  ```
  Initialization variable = value
  ```

  Boolean variables can be set either with "Yes" or "No," "True" or "False."
- Comments begin with a semicolon:

  ```
  ; Here I am, JP.
  ```

  Visual SourceSafe ignores all comments.
- Headers are provided in brackets:

  ```
  [$/Word]
  ```

  Most headers specify particular project paths.   Other headers, such as [Keyword Comments], are used for specific purposes.

The order of variables inside an initialization file does not matter; however, the header under which a variable is placed matters a great deal.   If a variable is placed under the wrong header—for instance, if a general-purpose variable is placed under [Keyword Comments]

—it is ignored.

SS.INI on Multiple Platforms

If you are running a multi-platform Visual SourceSafe installation, you may want certain variables to have different values on the different platforms.   You can therefore place a platform in parentheses after the variable, to indicate that the variable applies only on that platform, for example:

```
Use_ReadOnly (Win) = Yes
```

The setting in the example applies only under Microsoft Windows.   Other settings include UNIX, Win, NT, MAC, and PC−the last is a catch-all for personal computer operating systems.

See Also

Using Project Headers in .INI Files

Visual SourceSafe Naming Syntax and Conventions

Using Project Headers in .INI Files
<u>See Also</u>

Project headers allow you to specify variables on a project-by-project basis.   Because almost every variable can be placed under a project header, you can customize Visual SourceSafe so that different projects act very differently.

If a variable occurs in the SS.INI file before any header, it applies to all projects.   If a variable is placed under a project header, it applies only under that project.   Consider the following example, in which the Diff_Format variable is set three times:

```
;the following variable applies to all projects
Diff_Format = Visual
;the following variables apply under $/Reports
[$/Reports]
Diff_Format = SS
Dir = C:\DEV
;the following variable applies under $/Reports/JournalLogs
[$/Reports/JournalLogs]
Diff_Format = UNIX
```

The first setting of Diff_Format occurs before any header; consequently, it sets the default for all projects.   The second setting of Diff_Format overrides this setting for the project $/REPORTS.   The setting also, by default, is **<u>inherited</u>** by all **<u>subprojects</u>** under that project.   However, under $/REPORTS/JOURNALLOGS, the variable is explicitly set to UNIX; this setting blocks other inheritance, and applies only to JournalLogs and its subprojects.

The Dir variable behaves differently, because it specifies a folder path.   Inheritance in projects creates a folder tree that mirrors the project tree.   In the example, for $/REPORTS/JOURNALLOGS, the Dir variable is not C:\DEV, but rather C:\DEV\JOURNALLOGS.

See Also

Customizing the SS.INI and SRCSAFE.INI Files

Undoing Check Outs

Sometimes a user inadvertently leaves a file checked out.   This keeps other users from checking out and editing the file.   If the user cannot be located, the Visual SourceSafe administrator can undo the user's check out.

–   **To undo a user's checkout**

1 Log on to **Visual SourceSafe Explorer** as Admin, using the administrator password if there is one.

2 Click the **checked-out file** in the proper project.

3 On the SourceSafe menu, click Undo Check Out.

ANALYZE Utility Error Messages

This topic reviews common messages produced by the Analyze utility.   Although usually echoed to the screen, the ANALYZE messages are also written to the DATA\BACKUP\ANALYZE.LOG file by default.

The following is a listing of most common ANALYZE messages.   These are messages that appear if ANALYZE is run with or without the -F switch.   The messages that appear when the -V switch is used are not documented here.

In order to understand some of the messages it is crucial to understand how SourceSafe stores files and projects.   There are two physical files created for each file and project in SourceSafe.   These two files are stored in a subfolder of the DATA folder.   The subfolder is the same as the first character of the SourceSafe file name, also known as the physical file name.   One of the two files is called the 'Log' file.   This is the one without an extension.   This is where SourceSafe information and differences between one version of the file and the next are stored.   The other file is called the 'Data' or 'Tip' file.   This is the file with an extension of either .A or .B.   This is where SourceSafe stores the most recent version of the file or project.

There are two ways to identify what the physical file is:

1. The following creates a physical.txt file that indicates all the non-deleted and non-corrupt files in the SourceSafe database:

   ```
   SS physical $/ -r -ophysical.txt
   ```

   Then search physical.txt to find the file you are looking for.   If you are searching by the file name (MyFile.txt), be sure to scan the whole file as there may be multiple instances of a file.

1. The other option to identify a file name from the physical file name is to open the physical log file, the one without an extension, in a text editor and search for the file name.   First, a word of caution, don't write anything to this file and definitely don't save any changes to this file!!! If you do you will have corrupted your file.   The name of the file is usually on the first line.   It will say 'SourceSafe', then some garbage characters, then the name of the file or project, then some more garbage characters, then the extension of the data file that it's looking for (.A or .B).   Although it is harder to determine, you may also be able to identify what project this file is in if you look further in the file.

Conventions

The following conventions are used in the messages below:

- <Physical Log File Name> is the file in the DATA folder without an extension, i.e.   BKAAAAAA
- <Physical Data File Name> is the file in the DATA folder with an extension of .A or .B, i.e. BKAAAAAA.A
- <Project Name> is the name of the project in the Visual SourceSafe interface, i.e $/MyProject/Sub-Project
- <File Name> is the name of the file in the Visual SourceSafe interface, i.e.   MyFile.txt
- # is a number, i.e.   1

Messages

Database analysis in progress.

This message is informational only to let the user know the ANALYZE process has started.

Incompatible database version.

The version.dat file has the wrong information in it.   This usually occurs when a user is upgrading

from an older version of Visual SourceSafe.   It occurs because of a problem in the conversion process.   The problem is usually documented in the ddcerr.log file that is located in the DATA folder. When the problem is cleared up, usually re-running the ddconv utility will update the version.dat file. The syntax for re-running the ddconv utility is:

```
ddconv <Path to Data>
```

Version.dat is a two byte file with the version number of the database.   SourceSafe version 3.x has a database version of 05 and Visual SourceSafe has a database version of 06.   This message occurs because the version.dat file still says 05.   If the database can't be accessed with the 5.0 executables then don't manually change the version.dat file.   If the database can be accessed with the SSEXP executables, then it is probably okay to hack this file and change the value.   To do this you can, use a Hex editor or create a blank empty database with MKSS and DDCONV, copy the created version.dat over to the live database and then delete the newly created blank empty database.

Unable to open project <Path Used as Analyze Parameter>\a\aaaaaaaa.   Continue?

Example:

```
Unable to open project f:\vss\a\aaaaaaaa
Continue?
```

ANALYZE cannot find the main project in the database.   This is usually due to pointing to the wrong subfolder when running ANALYZE.

Encountered a bad CRC in <File Name>; record type <Record Header Type>.

Example:

```
Encountered a bad CRC in status.dat; record type SH.
```

Each record in the Database has a record header that contains the size of the record, the record type, and a CRC (Cyclic Redundancy Check), which is accumulated for the records data block.   Analyze reads in the header and computes the CRC for the data that is currently there.   If it does not match the CRC in the header, this message is displayed.   The key information is the File Name and Record Header Type, if this is unrecognizable then the header is also bad.   The impact of this corruption is that this record is lost.   Depending on the type of the record, Analyze may leave alone, fix it, or just throw it away.   If the File Name and Record Header Type are recognizable, the chances are increased that ANALYZE -F can recover the file.

This usually only occurs once at the beginning of the log file.   It occurs most often for users who are upgrading from 3.x.   There is corruption in the status.dat file.   The status.dat file is used to store whether a file is checked out or not.   Usually running ANALYZE -F <Path to Data>, fixes the problem in the status.dat file.

Project log '<Project Physical Name>' has a <Log Type> record for item '<File Name| Project Name>' but that item Was|Wasn't found in the project.

Example:

```
Project log 'DGEAAAAA' has a create record for item 'MyFile.txt', but that
item wasn't found in the project.
```

Analyze takes the log records from a project and the current list of children records in the project and plays the log records backwards until it gets to the beginning of the list of log records.   The list of children should be empty corresponding to the creation of the project.   This error just says that the history of the project does not line up with the contents of the project.   This is just an internal check. This message may or may not be fixed by ANALYZE -F depending on the problem in the log.   This is NOT a dangerous error and can be left alone.   This message almost always appears with the next message.

Using the -pss switch of ANALYZE you can determine what log entries are either incorrect, missing or

should not be there and properly correct them.

This message is believed to be caused by a bug in older builds of DDCONV (811 and 831). When the conversion process reached a log entry that was a rename, it would go back to the create record and change the create record to refer to the new name instead of the old name.

The project contents as rebuilt from the log '<Physical Log File Name>' does not match the project's actual contents.

Example:

```
The project contents as rebuilt from the log 'DGEAAAAA' does not match the
project's actual contents.
```

This message almost always appears after one or more of the previous messages. It is because of the same problem as documented in the previous message. This is just an internal check. This message may or may not be fixed by ANALYZE -F depending on the problem in the log. This is NOT a dangerous error and can be left alone.

No parent(s) or branch(es) were found for file '<Physical Log File Name>'.

Example:

```
No parent(s) or branch(es) were found for file 'BRGAAAAA'.
```

This means that the file currently has no parent or branch records, and this file will be put on the delete list. If this file was rolled back or branched from another file then it will be taken off the delete list later. Usually running ANALYZE with the -F and the -D switches will fix the references or destroy these files if appropriate. This message is not an error in your source code.

The CRC for data file 'File Name' (Physical Data File Name) does not match the stored CRC. The file may be corrupt. The file was last checked in on   "Date; Time" by user "User Name" in project 'Project name'.

Example:

The CRC for data file 'MyFile.txt' (YBGAAAAA.a) does not match the stored CRC. The file may be corrupt. The file was last checked in on '10/15/96; 11:26a" by user 'Guest' in project '$/MyProject'.

Explanation:

This message usually occurs without the last sentence about the last checked in date and time. This is a message that could potentially mean the loss of older versions of the file if not caught right away. Visual SourceSafe issues this message because the log file records the CRC for the last updated copy of the file and the last recorded CRC does not match the current CRC. To fix the error, first validate that the copy in the working folder of the user who last checked in the file, matches the Physical Data File Name. If it does, make sure that the 'Check in unchanged files' is set to 'Check In'. This option is under the Tools Menu/Options option /General Tab. Then check out the file and check it back in. This will correct the stored CRC.

No parent project for subproject file '<Physical Log File Name>'

Example:

```
No parent project for subproject file 'ABBAAAAA'.
```

Explanation:

This means that this project has been left around from the deletion of its parent, or that the parent project has somehow been lost. This error is usually fixed by ANALYZE -F <Path to Data> by either reconstructing the parent or removing the sub project if it is no longer needed.

The data file for "File Name(Physical Data File Name)" was not found.

Example:

```
The data file for 'EQUATES.INC' (aiaaaaaa.b) was not found.
```

Explanation:

SourceSafe keeps the last copy of each file in the database as an individual file (.A or .B), also known as the data file.   Analyze noticed that the data file that corresponds to the filename is missing from the database.   It knows whether to look for a .A or a .B file based on an entry in the log file (AIAAAAAA.) This is usually caused by a network or server problem with file creation.

If <File Name> is a project, you can run ANALYZE -F <Path to Data> to fix it.

If <File Name> is a file, there is no easy fix.   The best thing to do is to access the history of the file and find out the user who last checked in the file.   Then obtain the latest version of that file from the working folder of the last user to check in the file.   Copy that file from to the physical data file name (with the .A or .B extension as specified in the error message) to the correct subfolder in the database.   The correct subfolder can be determined from the first letter of the file.   Be sure to rename the file as part of the copy process.

If you have multiple instances of this file in different projects in your database, you will have to determine which project this file is in.   To do this you can run the following command:

```
SS physical $/ -r -ophysical.txt
```

Then search in the file for the Physical Data File Name without the extension.   In the example, above this would be aiaaaaaa.   If you move up in the file from occurrence of aiaaaaaa you will see the project name that the file is in.

There is a diff chain size mismatch in file 'File Name'(Physical Log File Name)' at version # (versions earlier than that version can no longer be retrieved from the database).

Example:

```
There is a diff chain size mismatch in file 'MyFile.txt' (FYIAAAAA) at
version 12 (versions earlier than that version can no longer be retrieved
from the database).
```

This message is an error and means that versions older than the one specified can't be retrieved.   This is usually caused by a corruption in a log entry record which causes the difference chain (or delta) to be unusable to properly regenerate older versions of the file.   If this occurred very recently, you may be able to retrieve the file (FYIAAAAA and FYIAAAAA.a (or .b) from backup.   Other than backups, there is no fix for this.

File '<Physical Log File Name including path>' is not the correct SourceSafe version.

Example:

```
File 'f:\vss\data\H\HACKAAAA' is not the correct SourceSafe version.
```

Each physical log file maintains its database version.   The setup process of Visual SourceSafe runs a utility called DDCONV on the database.   DDCONV runs through the database and converts files to the current version.   This message is caused because the file did not get converted to the correct version.   This message can occur for many reasons like someone was using SourceSafe, the file couldn't be written to or the file was so corrupt in the earlier version of SourceSafe that DDCONV couldn't recognize it.   If there is just a lock on the file you can safely run DDCONV against the database and it will convert this file.   The syntax to run DDCONV is:

```
ddconv <Path to Data>
```

If there is corruption in the file/project, it will have to be fixed before DDCONV can convert it.   Unfortunately, since the rest of the database has already been upgraded to 5.0 it is hard to determine what that corruption is.   You can recover your 3.x database and run the 3.x ANALYZE on the 3.x data

to determine the corruption.   Once it is fixed you can run DDCONV on the data again.

If the file/project is not needed then it and it's corresponding data file (.A or .B) can be moved out of the DATA subfolder and the new ANALYZE run with the -F switch to clean up any links to the file. See the directions in the SUMMARY section of this article to get the new ANALYZE.

The file '<Physical Log File Name including path>' appears to be corrupt.   Unable to read the format or header.

Example:

```
The file 'f:\vss\data\O\ORLAAAAA' appears to be corrupt.  Unable to read
the format or header.
```

Files in Visual SourceSafe have format and header records to identify the file.   One or both of these is corrupted.   This is very serious because this usually means the rest of the log file is corrupt.

The best solution for this is to get the file from backup.

If no backups are available and it is a project, delete the files from the DATA\?\ folder (Be sure to delete the file with the extension of .a or .b and the file with no extension.) Then run ANALYZE -F <Path to Data> to clean up the links.   All files that were in that project are likely to be lost.

If it is a file, then make a copy of the data file (the one with the extension of .a or .b).   Then delete the files from the DATA\?\ folder.   Delete the file with the extension of .a or .b and the file with no extension.   Then rename the .a or .b file to it's real name.   Finally, add the file back into SourceSafe. You will have to identify the file.   See directions for identifying a file at the beginning of this article.

The file <File Name> was branched from <Physical Log File Name> which is missing a branch reference.   A reference will be added.

Example:

```
The file CBLIST.ASM was branched from NNAAAAAA which is missing a branch
reference.  A reference will be added.
```

The file this one was branched from, doesn't remember that it branched to this file.   Analyze is letting you know that it is adding a reference of that branching to the original file.

The file <File Name> was branched from <Physical Log File Name> which is now corrupted and the early versions will be unaccessible.

Example:

```
The file MyFile.txt was branched from FIRAAAAA which is now corrupted and
the early versions will be unaccessible.
```

This is an informational message to let you know that versions before the file was branched are not available due to corruption in the file that it was branched from.   Ideally you would fix the corruption in the branched from file, the physical log file name in the message.   There is probably another message in the ANALYZE.log file about the corruption in the branched from file.

The item <File Name> has an extra parent relationship which will be removed.

Example:

```
The item MyFile.txt has an extra parent relationship which will be removed.
```

This is an informational message to let you know that the file listed has a parent that is not needed. Analyze -F is removing the extra parent record.

The Header information in the rights system is corrupt.

There is corruption in the rights.dat file.   This is the file that stores project security information.   This

can usually be fixed by running ANALYZE with the -F switch.

A rights setting mismatch was found.

There is a pointer off in the rights.dat file.   This is the file that stores project security information.   This can usually be fixed by running ANALYZE with the -F switch.

Found a reference to an invalid rights block.

There is corruption in the rights.dat file.   This is the file that stores project security information.   This can usually be fixed by running ANALYZE with the -F switch.

The nameset information for <File Name> is corrupt.

Example:

```
The nameset information for Myfile.txt is corrupt.
```

There is corruption in the names.dat file for the listed file name.   The names.dat file is where long file/project information is stored.   Up to 33 characters can be recovered by running ANALYZE with the -F switch.

Sample ANALYZE Log File

The following is a sample log file generated by the ANALYZE utility.   Many redundant messages were deleted from this sample, to save space.

```
Database analysis in progress.
Encountered a bad CRC in status.dat; record type SH.
Project log 'ADLAAAAA' has a share (from another project) record for item
'gervcs.cfg', but that item wasn't found in the current project.
Project log 'AEMAAAAA' has a share (from another project) record for item
'qb_tax_f', but that item wasn't found in the current project.
Project log 'BEMAAAAA' has a share (from another project) record for item
'atg_debu.lib', but that item wasn't found in the current project.
No parent project for subproject file 'BSHAAAAA'.
No parent project for subproject file 'CSHAAAAA'.
Project log 'DDLAAAAA' has a share (from another project) record for item
'cindexpp.¹', but that item wasn't found in the current project.
No parent project for subproject file 'DSHAAAAA'.
No parent project for subproject file 'ESHAAAAA'.
No parent project for subproject file 'FSHAAAAA'.
Project log 'FNHAAAAA' has a share (from another project) record for item
'macintax', but that item wasn't found in the current project.
Project log 'GCLAAAAA' has a share (from another project) record for item
'quickbo1', but that item wasn't found in the current project.
Project log 'GDLAAAAA' has a share (from another project) record for item
'cindexpp.¹', but that item wasn't found in the current project.
No parent project for subproject file 'GSHAAAAA'.
Project log 'HKGBAAAA' has a share (from another project) record for item
'svrgloba.h', but that item wasn't found in the current project.
Project log 'HCLAAAAA' has a share (from another project) record for item
'quickbo1', but that item wasn't found in the current project.
No parent project for subproject file 'HSHAAAAA'.
Project log 'JKGBAAAA' has a share (from another project) record for item
'qb_tax_f', but that item wasn't found in the current project.
No parent project for subproject file 'JVKAAAAA'.
Project log 'KCLAAAAA' has a share (from another project) record for item
'ppcbutto.¹', but that item wasn't found in the current project.
Project log 'LBABAAAA' has a share (from another project) record for item
'testmac.mak', but that item wasn't found in the current project.
Project log 'LHFAAAAA' has a share (from another project) record for item
'quill.rsr', but that item wasn't found in the current project.
No parent project for subproject file 'LRHAAAAA'.
Project log 'MCLAAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
Project log 'MAFAAAAA' has a share (from another project) record for item
'quill.rsr', but that item wasn't found in the current project.
Project log 'NUFAAAAA' has a share (from another project) record for item
'pictbut0.rsr', but that item wasn't found in the current project.
Project log 'NTSAAAAA' has a share (from another project) record for item
'svrgloba.h', but that item wasn't found in the current project.
Project log 'NEGAAAAA' has a share (from another project) record for item
'macintax', but that item wasn't found in the current project.
Project log 'NOFBAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
```

Project log 'NTNAAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
Project log 'OTSAAAAA' has a share (from another project) record for item
'qb_tax_f', but that item wasn't found in the current project.
Project log 'OBABAAAA' has a share (from another project) record for item
'qb_print', but that item wasn't found in the current project.
Project log 'POFBAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
Project log 'QTNAAAAA' has a share (from another project) record for item
'undiverg', but that item wasn't found in the current project.
No parent project for subproject file 'QRHAAAAA'.
No parent project for subproject file 'RRHAAAAA'.
Project log 'RTNAAAAA' has a share (from another project) record for item
'mwstartu', but that item wasn't found in the current project.
Project log 'RIGBAAAA' has a share (from another project) record for item
'quill.rsr', but that item wasn't found in the current project.
Project log 'STNAAAAA' has a share (from another project) record for item
'quill_up', but that item wasn't found in the current project.
Project log 'TOFBAAAA' has a share (from another project) record for item
'testmacp.¹', but that item wasn't found in the current project.
Project log 'TTNAAAAA' has a share (from another project) record for item
'testmacp.¹', but that item wasn't found in the current project.
File 'e:\ssnebula\data\T\TTBBAAAA' is not the correct SourceSafe version.
Project log 'VOLAAAAA' has a share (from another project) record for item
'quillppc.¹', but that item wasn't found in the current project.
Project log 'VXSAAAAA' has a share (from another project) record for item
'qb_tax_f', but that item wasn't found in the current project.
Project log 'VBLAAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
Project log 'WVFAAAAA' has a share (from another project) record for item
'unchecko', but that item wasn't found in the current project.
Project log 'WBLAAAAA' has a share (from another project) record for item
'qb_tax_Õ', but that item wasn't found in the current project.
Project log 'XBLAAAAA' has a share (from another project) record for item
'quickboo', but that item wasn't found in the current project.
Project log 'XTGBAAAA' has a share (from another project) record for item
'taxtypep', but that item wasn't found in the current project.
Project log 'YCLAAAAA' has a share (from another project) record for item
'toolrout.h', but that item wasn't found in the current project.
Project log 'YBLAAAAA' has a share (from another project) record for item
'qb_tax_Õ', but that item wasn't found in the current project.
Project log 'YAEBAAAA' has a share (from another project) record for item
'sourceto', but that item wasn't found in the current project.
Project log 'YDMAAAAA' has a share (from another project) record for item
'taxtypep', but that item wasn't found in the current project.
Project log 'YTGBAAAA' has a share (from another project) record for item
'qb_tax_Õ', but that item wasn't found in the current project.
Project log 'ZDMAAAAA' has a share (from another project) record for item
'macintax', but that item wasn't found in the current project.
No parent(s) or branch(es) were found for file 'AYSAAAAA'.
No parent(s) or branch(es) were found for file 'AATAAAAA'.
No parent(s) or branch(es) were found for file 'ABTAAAAA'.
There is a diff chain size mismatch in file 'formmgr.c' (BFDAAAAA) at
version 29 (versions earlier than that version can no longer be retrieved
from the database).

There is a diff chain size mismatch in file 'el_todo.c' (BXEAAAAA) at
version 6 (versions earlier than that version can no longer be retrieved
from the database).
There is a diff chain size mismatch in file 'cformif.h' (BOFBAAAA) at
version 2 (versions earlier than that version can no longer be retrieved
from the database).
There is a diff chain size mismatch in file 'toolbar.c' (BXDAAAAA) at
version 15 (versions earlier than that version can no longer be retrieved
from the database).
There is a diff chain size mismatch in file 'goto.c' (BWEAAAAA) at version
9 (versions earlier than that version can no longer be retrieved from the
database).
There is a diff chain size mismatch in file 'windows.c' (BAEAAAAA) at
version 38 (versions earlier than that version can no longer be retrieved
from the database).
No parent(s) or branch(es) were found for file 'BQNAAAAA'.
No parent(s) or branch(es) were found for file 'BRNAAAAA'.
No parent(s) or branch(es) were found for file 'TWSAAAAA'.
No parent(s) or branch(es) were found for file 'TOTAAAAA'.
No parent(s) or branch(es) were found for file 'TPTAAAAA'.
No parent(s) or branch(es) were found for file 'TQTAAAAA'.
The Header information in the rights system is corrupt.
Found a reference to an invalid rights block.
Found a reference to an invalid rights block.
Found a reference to an invalid rights block.
Found a reference to an invalid rights block.
Found a reference to an invalid rights block.
Found a reference to an invalid rights block.
Analysis complete.

Setting up a Web Site Project

Visual SourceSafe includes several features which make managing World-Wide Web sites easier. For example, you can designate a specific Visual SourceSafe project as a Web site project, and then quickly create a site map of the HTML files in the project (and subprojects), Check Hyperlinks in the project files, and deploy the files to an internet server location when they are ready.

The first step in taking advantage of these features is to set up your Visual SourceSafe database to properly handle web site projects.   You do this by setting the options on the Web tab of the Tools Options dialog box.   Then, using the Web Projects tab in the same dialog box, you designate an existing project as a web site project.   After that tab is completed, exit from the Administrator and have all your clients exit and restart.   You should see a special icon () in the Visual SourceSafe Explorer on that project.

From there, check the topics on the following commands for more information on using these powerful features:

- Create Site Map
- Check Hyperlinks
- Deploy

Web Project Initialization Variables

There are three generalized initialization variables for handling Web projects.   Two apply to Web projects that are deployed via FTP through a firewall, and one applies to the Check Hyperlinks command.   You set these with the Web tab

A typical set of initialization variables for Web site deployment might look like the following:

```
Deploy_Path = \\mysrvr\bar, FTP://www.microsoft.com/baz
.
.
.
Deploy_Proxy = itgproxy

Deploy_Host_Local = www.microsoft.com
```

The first variable lists the deployment paths for a particular Web site project, some of which can be FTP sites.   The second specifies the proxy you use for deploying Web sites via FTP through the firewall at your site.   The third says that for any projects you deploy to your local server, in this example **www.microsoft.com**, the proxy should not be used.

If no **Deploy_Proxy** is specified, no proxy is ever used.   If a **Deploy_Proxy** is specified, it is used for all FTP deployment, except for the server(s) specified in **Deploy_Host_Local**.   Note that **Deploy_Host_Local**, like **Deploy_Path**, can be a list.   However, **Deploy_Proxy** can only be set to one value.

Locking the Visual SourceSafe Database

When you need to archive, backup, or run a utility on the Visual SourceSafe database, it is safer and more convenient to lock the database so that new users trying to log into the database are prevented from doing so while your operation is underway.   Note that the act of checking the checkbox in the Lock SourceSafe Database dialog box is what actually locks users out.   Once this box is checked, no one is allowed to log into the database.   The users listed who are already logged in are unaffected – you need to notify them manually to log out.

When the database is locked, a small key is displayed in the right hand corner of the Administrator status bar.

Locking the Visual SourceSafe Database See Also

Lock Visual SourceSafe Database Command

Lock Visual SourceSafe Database Dialog

HTML Keyword Expansion

<u>See Also</u>

Keyword expansion refers to Visual SourceSafe's ability to place certain information directly from the Visual SourceSafe database into your file for you.   This can be handy, for example, if you want to place the string "Last modified on: *date*" in your web pages.

To do this, you place certain keywords, in comments so that they do not affect your HTML code, into the text of your file.   When you add or check in the file, Visual SourceSafe looks for these keywords, and places the relevant information after them.

SourceSafe Keywords

The following table is a list of all the Visual SourceSafe HTML keywords.   Note that these HTML keywords are identical to traditional SourceSafe keywords − except that they have an extra dollar sign in front of them.   Keywords are case-sensitive, so be sure to capitalize them properly:

Keyword	Description
$$Archive: $	Visual SourceSafe archive file location
$$Author: $	User who last changed the file
$$Date: $	Date and time of last check in
$$Header: $	Logfile, Revision, Date, Author
$$History: $	File history, Visual SourceSafe format
$$Log: $	File history, RCS format
$$Logfile: $	Same as Archive
$$Modtime: $	Date and time of last modification
$$Revision: $	Visual SourceSafe version number
$$Workfile: $	Filename
$$NoKeywords: $	No keyword expansion for all keywords that follow. (The colon is optional.)
$$JustDate: $	Date, without the time addendum.

A SourceSafe HTML keyword has the following syntax:

```
$$keyword: $
```

For example, you could place the following in a file:

```
Last updated <!--$$JustDate:--!>  <!--$-->
```

Upon check in, Visual SourceSafe could replace it with

```
Last updated <!--$$JustDate:-->5/16/96<!--$-->
```

The next time you check in the file, the 5/16/96 is replaced by the current date, and so on.   This automates the process of updating your HTML pages each time you modify them.

Enabling Keyword Expansion

Because keyword expansion requires Visual SourceSafe to scan each file for keywords, it can considerably slow the Check In and Add Files commands.   For this reason, keyword expansion is by default disabled for all files; you must indicate which files Visual SourceSafe should scan for keywords.

− **To enable and view keyword expansion**
- Set the Copy Keyword-Expanded Files Into Working Folder option on the Local Files tab of the Options dialog box (Tools menu) to view the effects of keyword expansion.

After a Check In or Add command, Visual SourceSafe copies the keyword-expanded versions of the files to your **working folder**.

Effect of Keyword Expansion

When you check in a file with keyword expansion enabled, Visual SourceSafe places a modified copy of your file directly into the current project.   It then immediately gets the file, which refreshes your working folder with the checked-in file.   The Get Latest Version command slows down the updating process.   You can avoid it by not setting the Copy Keyword Expanded-Files Into Working Folder option on the Local Files tab of the Options dialog box.   When you clear this option, Visual SourceSafe still expands keywords, and places the modified file into the project.   However, it does not copy the file into your working folder.

See Also

Expanding Keywords

Visual SourceSafe's Web Site

Visual SourceSafe maintains a <u>web site</u> where you can get the latest information, free utilities to use with Visual SourceSafe, pricing information, helpful hints, and more.   Check it out!

Help Menu Commands

The Help menu commands provide Help on using Visual SourceSafe Administrator and Help on specific Visual SourceSafe Administrator commands.   In addition, you can press F1 at any time to see Help on the currently selected menu item or dialog box.

Help Command	Action
Contents	Runs Help and displays Visual SourceSafe Administrator's Table of Contents.
Search For Help On	Displays the Help Search dialog box so you can quickly find the information you need.
Technical Support	Displays information on Microsoft Technical Support Services.
About Administrator	Displays a dialog box with information on your copy of Microsoft Visual SourceSafe Administrator, including the version number and the copyright, legal, and licensing notices.

Remarks

A quick way to find information is to use Search, an automated index with hundreds of index entries and extensive cross-referencing.

You can also look for Help topics in Visual SourceSafe Administrator's Table of Contents.

As you read a topic in Visual SourceSafe Administrator Help, you can scroll through the topic with the scroll bar, the arrow keys, or the PAGE UP and PAGE DOWN keys.
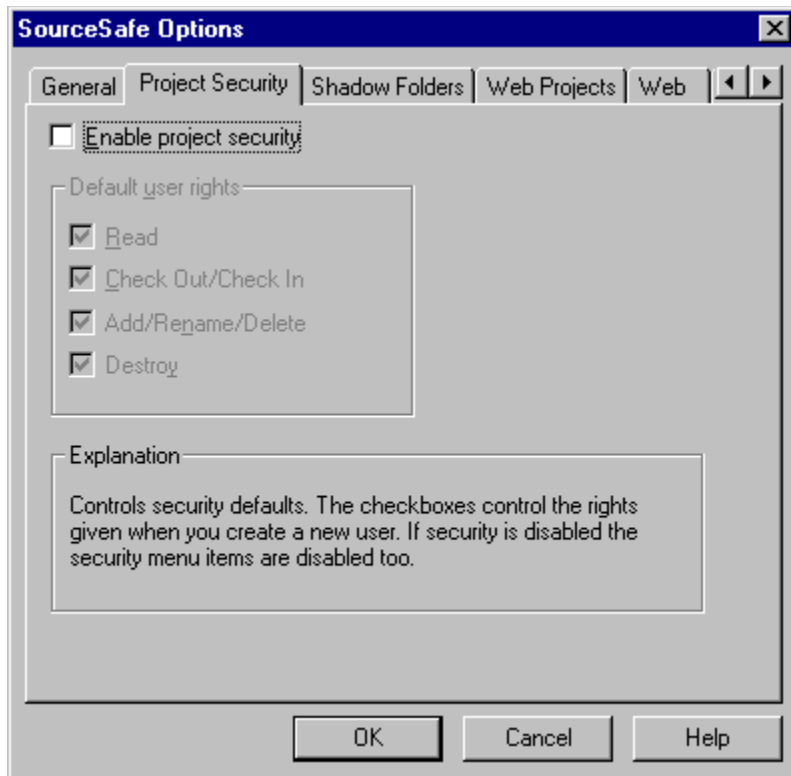
In the Help window, click Search to open Visual SourceSafe Administrator's index.   Click Back to go to the last Help topic viewed.

Help Menu Commands See Also

Technical Support

Project Security Options Tab

Controls Visual SourceSafe project **security**.   Use this tab to turn security on and off; if security is enabled, you can set default **access rights** for users.

Remarks

Your first decision is whether to enable project security.   If the Enable Project Security check box is clear, project-level security is disabled (Visual SourceSafe's default security is always operating).   If the Enable Project Security check box is selected, you can set default access rights for new users added to the Visual SourceSafe Administrator **user list**.   The Add Users command on the Users menu adds users to the user list.

For instance, if the Read, Check Out/Check In, and Add/Rename/Delete check boxes are selected, new users are given the Add access right in the **root project** (and therefore in all projects) by default. You can then change these rights in any project using the Rights by Project or Rights Assignments for User commands.

− **To display the Project Security tab**
- On the Tools menu, click Options, and then click the Project Security tab.

Tab Options

Enable Project Security

Enables the project security system, which provides the four levels of access rights shown below.

Read

Permits a user to view **read-only** copies of files.

Check Out/Check In

Permits a user to use the Check Out and Check In commands, but does not permit renaming, adding and deleting of files or projects, and so forth.

Add/Rename/Remove

Permits a user to use the Add, Rename, **<u>Delete</u>**, Check Out, and Check In commands, but does not permit actions that permanently destroy files or projects.

Destroy

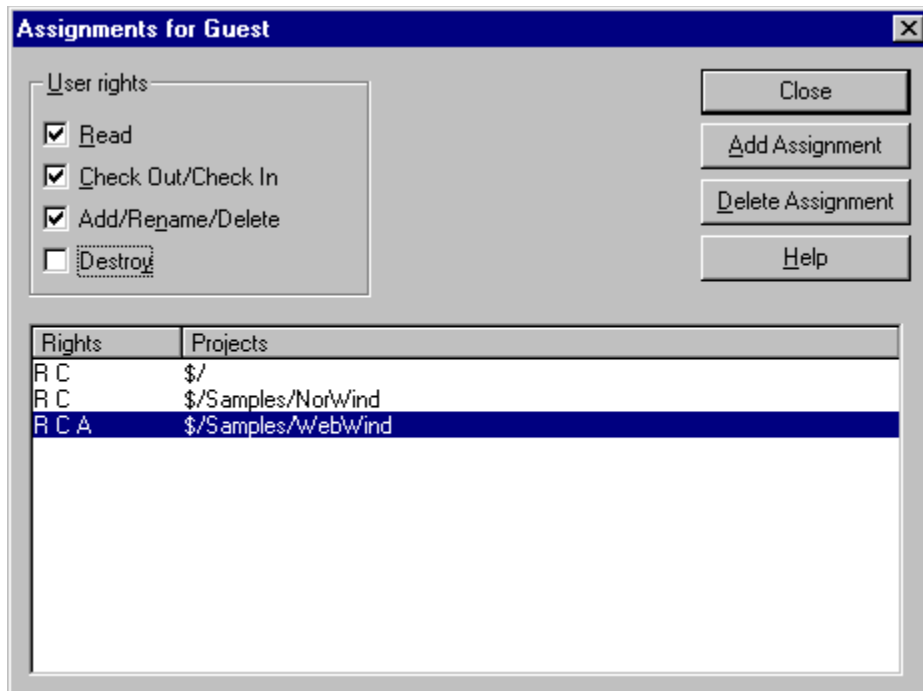Permits a user to use all Visual SourceSafe commands.   Equal to Read-Write access rights under the default security system.

See Also

Rights Assignments for User Command

Rights by Project Command

Visual SourceSafe Security Access Rights

Rights Assignments for User Command (Tools Menu)

Displays and changes project **access rights** to the **Visual SourceSafe database** explicitly assigned to the user selected in the **user list**.

Remarks

Default user access rights are assigned to each new user under default security or on the Project Security tab in the SourceSafe Options dialog box (Tools menu).

Note   To use the Rights Assignments for User command, you must first check Enable Project Security on the Project Security tab in the SourceSafe Options dialog box (Tools menu).

The Rights Assignments for User dialog box displays only those access rights assignments that were explicitly changed; the effect of **rights propagation** is not shown.   To see the propagation of access rights, use the Rights by Project command (Tools menu).

Dialog Box Options

User Rights

Specifies the level of access rights－Read, Check Out, Add or Destroy

－held by the user in the project selected under Projects.   To change these assignments, click the project you want and then click the appropriate access right check box.

Rights

Level of access rights assigned to the selected user for each project.   You can change the column size of this column by dragging the column separator after the word "Rights."

Project

Projects in which the selected user has been explicitly assigned access rights.

Close

Closes dialog box and returns to Visual SourceSafe Administrator.

Add Assignment

Displays the Add Assignments for User dialog box.   To add projects and assign or change access rights for the selected user, select a project in the project list, click the level of access rights you want to assign, and then click OK.

Delete Assignment

Deletes a user's access rights in the selected project(s).

See Also

Add Assignments for User Dialog Box

Project Security Options Tab

Rights by Project Command

Visual SourceSafe Access Rights

Add Assignment for User Dialog Box

Changes **access rights** for the user selected in the **user list**.

－   **To display the Add Assignment for User dialog box**

1 On the Tools menu, click Rights Assignments for User.

2 In the Rights Assignments for User dialog box, click Add Assignment.

Remarks

To assign or change access rights for the selected user, under Project, select a project in the **project list**, and then click the User Rights check box with the level of access rights you want to assign.

Note   To use the Add Assignment for User dialog box, you must first check Enable Project Security on the Project Security tab in the SourceSafe Options dialog box (Tools menu).

Dialog Box Options

Project

Project list for the current **Visual SourceSafe database**.

User Rights

Specifies the level of access rights－Read, Check Out, Add, or Destroy

－held by the user in the selected project.
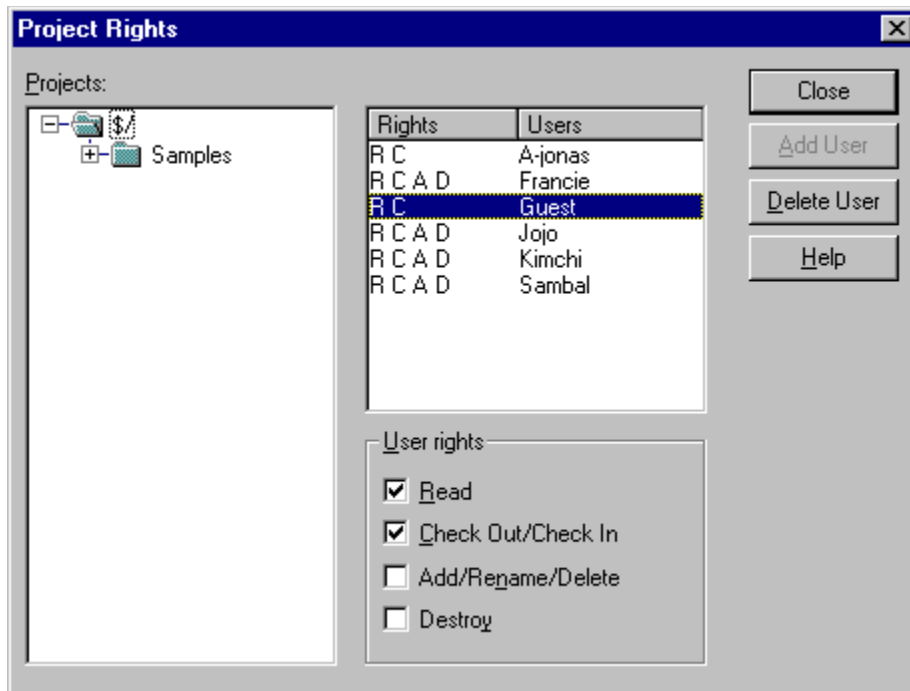
OK

Confirms selections.

Cancel

Cancels selections and returns to the Rights Assignments for User dialog box.

See Also

Rights Assignments for User Command

Rights By Project Command (Tools Menu)

Displays and changes __access rights__ for selected users in selected projects.

Remarks

Default user access rights are assigned to each new user under default security or on the Project Security tab in the SourceSafe Options dialog box (Tools menu).

This command shows the effect of __rights propagation__ in the Project Rights dialog box.   For example, if a user has the Destroy access right in the __root project__, then under Projects, you see that the user has the Destroy access right in all subprojects−unless you explicitly change the access rights.   Explicit changes are displayed in the Rights Assignments for Users dialog box, available from the Tools menu.

__Note__   To use the Rights by Project command, you must first check Enable Project Security on the Project Security tab in the Visual SourceSafe Options dialog box on the Tools menu.

Dialog Box Options

Projects

Project list of the current __Visual SourceSafe database__.   Select the project for which you want to view or change users' rights.

Rights

Access rights assigned to the user.

Users

Users who have access rights in the selected project.

User Rights

Specifies the level of access rights－Read, Check Out, Add, or Destroy

－that you want to assign the user selected in the User column.   To change the selected user's access rights for the selected project, click the appropriate check box.

Close

Closes this dialog box and returns to Visual SourceSafe Administrator.

Add Users

Displays the Add Users for Project dialog box, which adds users to the project selected under Projects.

Delete Users

Deletes the selected user's access rights to the selected project, after a warning message.

See Also

Add Users Command

Add Users for Project Dialog Box

Project Security Options Tab

Rights Assignments for User Command

Visual SourceSafe Access Rights

Add User for Project Dialog Box

Adds users to the project selected under Projects in the Project Rights dialog box.

Remarks

The users displayed in this dialog box are users in the **Visual SourceSafe database** who do not have **access rights** in the project selected in the Project Rights dialog box.

To display the Add Users for Project dialog box
- On the Tools menu, click Rights by Project, and then in the Project Rights dialog box, click Add User.

Note   To use the Add User for Project dialog box, you must first check Enable Project Security on the Project Security tab in the SourceSafe Options dialog box (Tools menu).

Dialog Box Options

Rights

Access rights assigned to the user.

Users

Users who don't have access rights in the selected project.

User Rights

Specifies the level of access rights－Read, Check Out, Add, or Destroy

－that you want to assign the user selected in the Users column.   After you click a check box, the access rights are displayed under Rights.

Note   If the user you want to add to the project has the Read-Only access right and you want to assign additional access rights, on the Users menu, click Edit User, and then clear the Read-Only check box.

OK

Confirms selections.

Cancel

Cancels selections and returns to the Project Rights dialog box.

See Also

Edit Users Command

Rights by Project Command

Visual SourceSafe Access Rights

Copy User Rights Command (Tools Menu)

<u>See Also</u>

Acts as a template and copies one user's **<u>access rights</u>** to another user.

Remarks

Instead of adding a user's access rights to every project a user is working on, you can simply copy all access rights from another similar user.

−   **To copy user rights**

1 In the **<u>user list</u>**, click the user whose access rights you want to modify.

2 On the Tools menu, click Copy User Rights.   The Copy Rights Assignments to User dialog box is displayed, providing a list of all Visual SourceSafe users.

3 Click a user to use as an access rights template, and then click Copy.

After you copy rights, the two users have identical access rights in every project.   You can, however, individually change their access rights in specific projects.   Future changes to one user are not duplicated for the other, unless you use this command to copy user rights again.

Dialog Box Options

Copy From

User whose access rights you want to copy.

Copy

Copies access rights from the user selected under Copy From to the user selected in Visual SourceSafe Administrator.

Cancel

Cancels the Copy User Rights command.

See Also

Add Users Command

Visual SourceSafe Access Rights
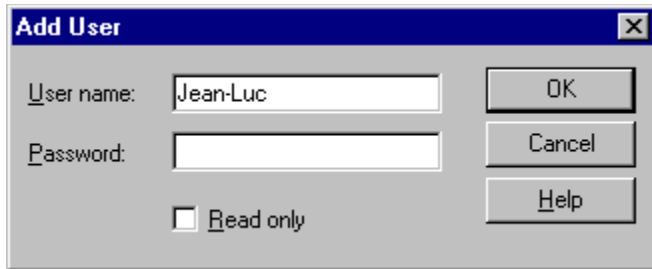
Font Command (Tools Menu)

Changes the font used in the **user list** in Visual SourceSafe Administrator.

Remarks

This dialog box enables you set the font name, style, and size for your Visual SourceSafe Administrator display.   The Sample box provides a preview of the currently selected font.

Add User Command (Users Menu)

Adds a new user to Visual SourceSafe Administrator's **user list**.

Remarks

The **username** and read-only access rights can be changed later with the Edit Users command, and the password can be changed with Visual SourceSafe Administrator's Change Password command. Users can also change their own passwords using Visual SourceSafe Explorer's Change Password command.

Dialog Box Options

User Name

Type the valid username of the user you want to add to the system.

Password

Type an optional Visual SourceSafe password.   As you type, the characters are displayed as asterisks.

Read Only

Assigns the Read **access right**.   Check if you don't want the user to have Read-Write access rights.

OK

Adds the user.

Cancel

Cancels the Add Users command and returns to Visual SourceSafe Administrator.

See Also

Change Password Command

Edit Users Command

Visual SourceSafe Access Rights

Delete User Command (Users Menu)

Deletes a user from Visual SourceSafe Administrator's **user list**.
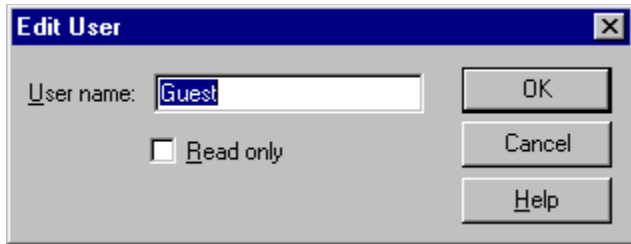
Remarks

–   **To delete a user**

- Click the user you want to delete in Visual SourceSafe Administrator's user list, and then on the Users menu, click Delete.   Click OK to confirm the deletion.

The user is deleted from the list, and is no longer able to run Visual SourceSafe.

If the deleted user has files checked out, a warning is not generated.   To unlock those files, run Visual SourceSafe as the Admin user and use the Undo Check Out command.

Edit User Command (Users Menu)

<u>**See Also**</u>

Changes the <u>**username**</u> and <u>**access rights**</u> of the user selected in Visual SourceSafe Administrator.

Remarks

Use the Change Password command on the Users menu to change a user's password.

Note   A shortcut for displaying the Edit User dialog box is to select the username in Visual SourceSafe Administrator's <u>**user list**</u>, and then press ENTER.

Dialog Box Options

User Name

Type a new name to change the name of the user selected in Visual SourceSafe Administrator.

Read Only

Assigns the Read <u>**access right**</u>.   Check if you don't want the user to have Read-Write access rights.

OK

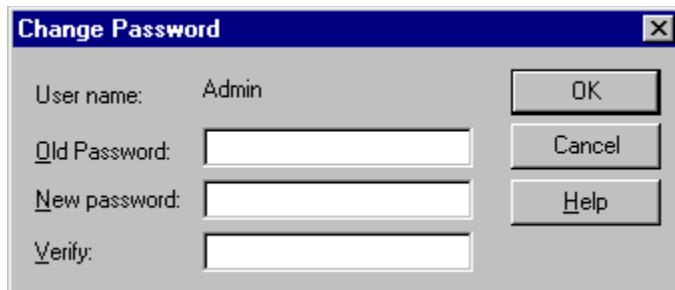Confirms your changes.

Cancel

Cancels the Edit Users command.

See Also

Change Password Command

Visual SourceSafe Access Rights

Change Password Command (Users Menu)

Sets the **password** for users, including the Admin user.

Remarks

Changing a user's password is something the administrator rarely has to do, because users can change their own passwords in Visual SourceSafe Explorer (Change Password command on the Tools menu).   When users forget their passwords, however, they can't run Visual SourceSafe.   It is not possible to find out what the user's password is.   You can, without knowing the password, use the Change Password command to change it.

Note   When you run Visual SourceSafe Administrator, you are prompted for the Admin user's password.   If you want to provide a password for the Admin user, use the Change Password command to give the Admin user a password.   Once you have done this, make sure not to forget the Admin user's password, as there is no way to discover it; you must call Microsoft Technical Support Services for assistance.

Dialog Box Options

User Name

Name of the user selected in Visual SourceSafe Administrator.

New Password

Type the new password.   A password can consist of any combination of 1 to 15 characters.   As you type, the characters are displayed as asterisks.   Visual SourceSafe passwords are not case-sensitive.

Verify

Retype the new password to verify it.

OK

Confirms the new password.

Cancel

Cancels the Change Password command.

See Also

Visual SourceSafe Naming Syntax and Conventions

Microsoft Technical Support Services

Exit Command (Users Menu)

Exits Visual SourceSafe Administrator.

Options Command (Tools Menu)

<u>See Also</u>

Enables you to customize the Visual SourceSafe environment for all users.

Remarks

Click the tab with the information you want.   Each tab contains groups of options you can set to alter Visual SourceSafe's behavior in certain situations.   The tabs in the SourceSafe Options dialog box change Visual SourceSafe's behavior for all users in Visual SourceSafe Administrator's **<u>user list</u>**. (Individual users can override some settings by using the Options command on **<u>Visual SourceSafe Explorer's</u>** Tools menu.)

Dialog Box Tabs

General

Sets general options applicable to a number of commands in Visual SourceSafe.

Project Security

Sets security options, such as whether project security is enabled, and the default user rights assignment.

Shadow Folders

Sets **<u>shadow folder</u>** information for all users.

Web Projects

Designates a project as a web project.

Web

Sets how certain internet information is transmitted for web projects.

File Types

Sets how file type information is determined for all users.
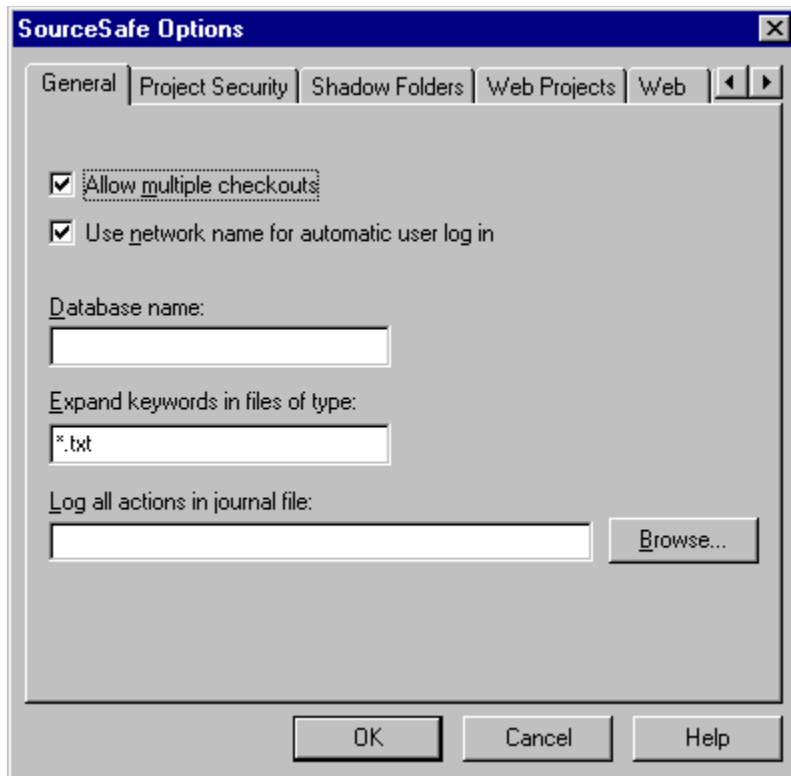
See Also

File Types Options Tab

General Options Tab

Project Security Options Tab

Shadow Folders Options Tab

General Options Tab (Tools Menu)

Sets general user options.

Remarks

Settings affect all users in Visual SourceSafe Administrator's **user list**.

–   **To display the General Tab**
- On the Tools menu, click Options, and then click the General Tab.

Dialog Box Options

Allow Multiple Check Outs

Controls whether many people can **check out** one file simultaneously.   Under ordinary circumstances, only one person can have a file checked out of Visual SourceSafe at one time.   If you check this box, however, Visual SourceSafe allows many people to check out the same file.   When the first user **checks in** the file, Visual SourceSafe updates the file; thereafter, Visual SourceSafe **merges** in all subsequent changes.

Use Network Name For Automatic User Log In

Sets whether Visual SourceSafe reads users' **logon** names from the operating system.   If you click this box, Visual SourceSafe automatically attempts to get **usernames** from the operating system.   Clear this box if you want users to use logon names different from their network logon names−Visual SourceSafe displays a Login dialog box unless users supply a username in some other way (with an environment variable, for example).

Database Name

Sets the name of the current **Visual SourceSafe database** as displayed in the Visual SourceSafe Explorer title bar.   If undefined, the Visual SourceSafe Explorer title bar reads "Visual SourceSafe Explorer," with no database name.

Tip   If you have a lot of different Visual SourceSafe databases, you can often end up running Visual SourceSafe without knowing which one you're connected to.   You can use the Database Name option to uniquely identify the database, and if you want to know what folder the copy of Visual SourceSafe is in, you can include the folder or share name in the database name.   There is a 31-character limit for the database name.

Expand Keywords In Files Of Type

Indicates on which files to use **keyword expansion**.   By default, Visual SourceSafe does not scan files for keyword expansion when it performs Check In or Add Files commands.

Log All Actions In Journal File

Sets the file where all Visual SourceSafe commands that change stored data are logged. (Commands that don't change data in the SourceSafe database are not logged.)   If you select this option, the specified file is a text file listing all Visual SourceSafe actions that users take.
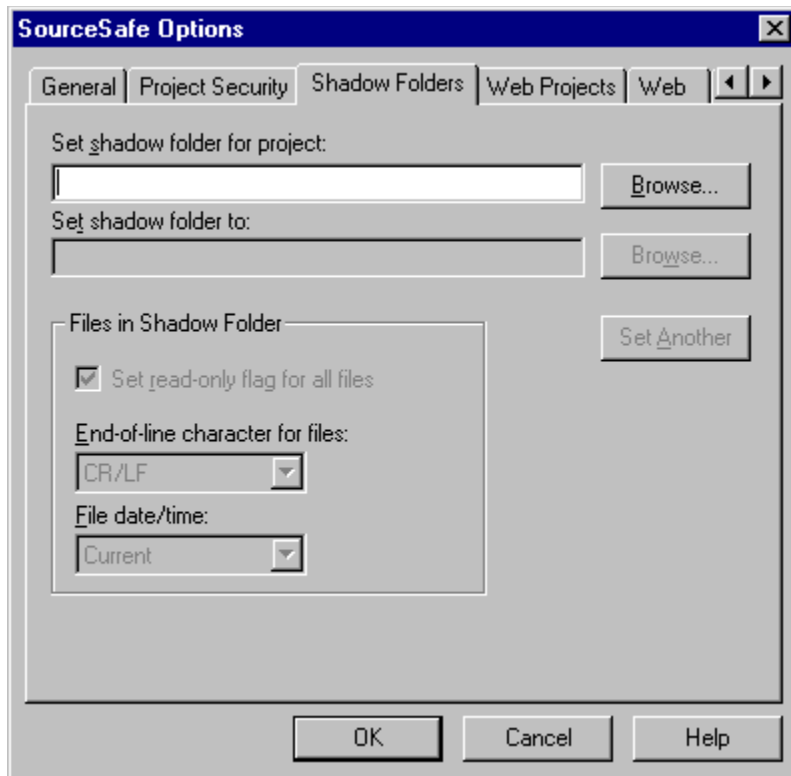
OK

Confirms selections.

Cancel

Cancels selections.

See Also

Using Multiple Check Out of Files

Shadow Folders Options Tab

See Also

Sets **shadow folders**.

Remarks

Use the Set Shadow Folder for Project option to specify a **project**.   Once specified, all other options on the tab apply only to the specified project.

– **To display the Shadow Folders tab**
- On the Tools menu, click Options, and then click the Shadow Folders tab.

Dialog Box Options

Set Shadow Folder For Project

Specifies the project for which you want to set up the shadow folder.   Click Browse to browse through the **Visual SourceSafe database** for the project you want.   Once you have entered a project name, any existing shadow information for that project is displayed in the remaining fields of the tab. Remember that shadow settings for a particular project may be set explicitly for that project or inherited from parent projects.

Set Shadow Folder To

Specifies the shadow folder.   This folder can be anywhere on the network, but in most cases is on the server with the Visual SourceSafe database.   Click Browse to search the file system for a suitable location.

Set Another

Applies the shadow folder specified in the other fields of this dialog box, and clears the dialog box so

you can set another shadow folder.   This button is not enabled unless you have specified a valid project and folder in the next two fields.

Set Read-Only Flags For All Files

Controls whether files in the shadow folder are **<u>read-only</u>**.

End-Of-Line Character For Files

Sets the end-of-line character used in the shadow folder.   When Visual SourceSafe copies a file into the shadow folder, it uses the end-of-line character used by its own operating system.   For example, Visual SourceSafe on Windows uses a carriage-return/line-feed pair.   In a multi-platform environment, the files in the shadow folder are a mixture of end-of-line characters, based on which operating system each file was updated from most recently.

This setting overrides that mechanism, forcing all versions of Visual SourceSafe to use the specified end-of-line character.   Set this field to LF for line feed (the UNIX default), CR for carriage return, or CR/LF for carriage-return/line-feed pairs (the PC default).   Note that Macintosh compilers can generally accept carriage-return/line-feed pairs, so CR/LF is a good setting for PC−Macintosh installations.

File Date/Time

Sets the date/time stamp given to files in the shadow folder.

By default, whenever Visual SourceSafe places a file in the shadow folder, its sets the date/time of the file to the current date/time.

However, if you set this variable to Modification or Check In, Visual SourceSafe sets the shadow folder file to the date/time that the file was last modified or checked in, respectively.

OK

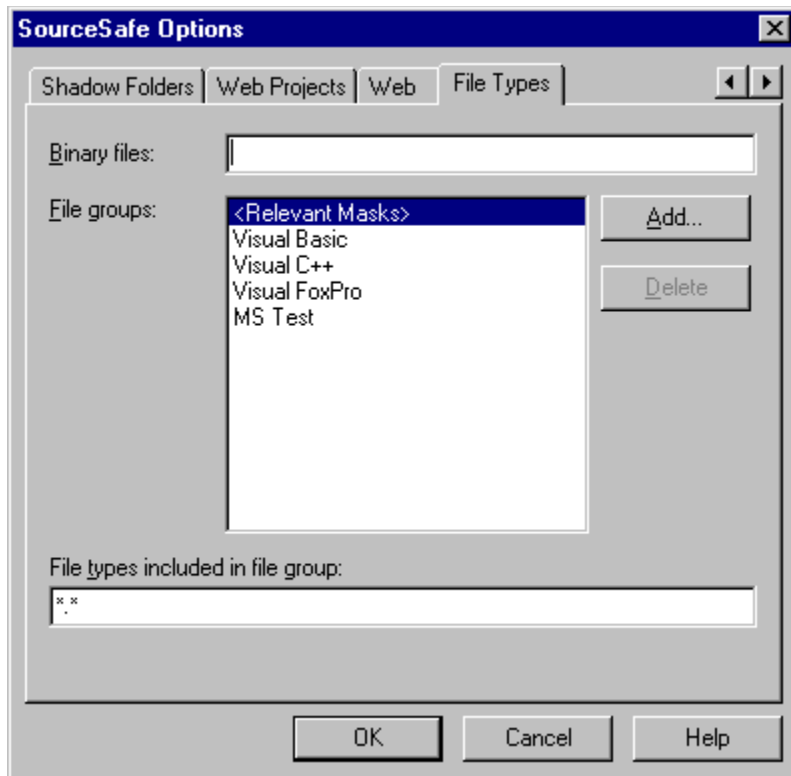Confirms selections, and saves the shadow settings to the SRCSAFE.INI file.

Cancel

Cancels selections and returns to Visual SourceSafe Administrator.

See Also

Using Shadow Folders

File Types Options Tab

Sets options relating to the types of files users store in Visual SourceSafe.

Remarks

Many of the Visual SourceSafe dialog boxes contain file lists that enable users to select multiple files for a specific operation.   Use this tab to filter these file lists so that they are more meaningful to the type of work your users do.   For example, if your users are C++ developers, they may want to store .C, .H, .OBJ, and .EXE files in Visual SourceSafe.   If they are Visual Basic developers, on the other hand, they may prefer .MAK and .FRM files.

-         **To display the File Types tab**
- On the Tools menu, click Options, and then click the File Types tab.

Dialog Box Options

Binary Files

Overrides Visual SourceSafe's automatic detection of text and binary files, and indicates that certain file types are always binary.   For example, you could enter *.OBJ, *.XLS, *.FRX in this box to specify that all object files, Microsoft Excel worksheets, and Visual Basic .FRX files should be treated as binary files.

File Groups

Groups file types for your users' type of work, together with the File Types Included in File Group option.   These groups are displayed under List Files of Type in many Visual SourceSafe dialog boxes, so you can specify groups that filter the file list efficiently for your users.

Add

Adds a new group of file types to the list.   In the Add File Group dialog box, type a name for the new file group you want to add, like "VC" or "VB."

Delete

Deletes the currently selected group.

File Types Included In File Group

Shows the list of file types included in the currently selected file group under File Groups.   To modify a specific group, select the group under File Groups, and then add or delete extensions in this box.

OK

Confirms selections.

Cancel

Cancels selections.

See Also

Setting the File Type: Text and Binary Files

About Visual SourceSafe (Help Menu)

Displays a dialog box with information about your copy of Microsoft Visual SourceSafe Administrator, including the version number, the platform type, and the copyright, legal, and licensing notices.

Click OK to close the dialog box and return to Microsoft Visual SourceSafe Administrator.

Browse Folder Dialog Box

Dialog Box Options

Name

Name and path of folder.

Folder

Your current folder.

Folders

Specifies a folder.

Drives

Selects the drive that contains the folder.

OK

Confirms the folder selection.

Cancel

Cancels the selection, closes the dialog box, and returns to Visual SourceSafe Administrator.

Create Folder

Creates a new folder as specified.   Type a new folder name in the Name box, then click Create Folder to create the folder.

Add File Group Dialog Box

Adds a new file group to the File Types tab in the Options dialog box (Tools menu).

− **To display the Add File Group dialog box**
- On the Tools menu, click Options, and then click File Types.   Click the Add button.

Dialog Box Options

Name

Type the name of the file group as you would like to identify it in the file group list on the File Types tab.
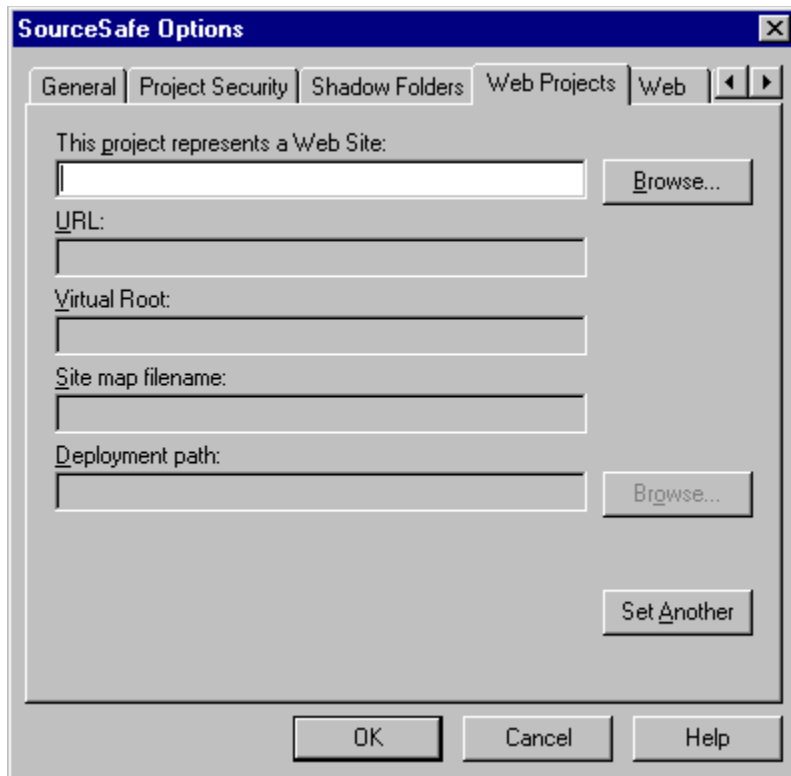
OK

Adds the file group.

Cancel

Cancels the Add File Group command.

See Also

File Types Options Tab

Web Projects Options Tab
See Also

Sets information applicable to world-wide web projects; that is, projects which contain files meant for HTML browsers.   Don't confuse the Web Projects tab with the Web tab, which sets options for all Web projects at once rather than for one individual web project.

Remarks

Use the top listbox to specify an existing Visual SourceSafe project as a web project.   Once specified, all other options on the tab (except proxy information, which is global) apply only to the specified project.

To remove the web project designation from a project, simply clear the URL field and click OK.

- **To display the Web Projects tab**
- On the Tools menu, click Options, and then click the Web Projects tab.

Dialog Box Options

This tab contains five text fields.   The first field and either or both of the URL and Virtual Root fields must be filled in before you can designate a project as a web project.   The remaining two fields are optional.

This project represents a web project
Type the name of the project into this field, or use the browse button to locate the project.   All the other settings apply to this project.

URL
Type the Uniform Resource Locator address for the web project into this field.

Virtual Root

Some Web server software supports this; if you have specified a virtual root in your server software, you should specify it here, without an initial slash.   Entering a value in this field sets a **Virtual_Root** variable in SRCSAFE.INI.   Note that lookups check through the SRCSAFE.INI file for any project with the specified virtual root.

Site map filename

Fill this field with the name of the file you want Visual SourceSafe to use when it creates a site map. (If you don't specify a value, SITEMAP.HTM is used by default.)

After you fill in the fields in the tab, have the clients involved in the Web project exit from Visual SourceSafe and restart.   Visual SourceSafe Explorer displays the designated project with a special icon to indicate that it is a Web project.

Note   You must designate a URL and/or a virtual root for the project.   The other fields are optional at the time the project is being designated as a Web project.

Deployment path

A deployment location can be on a hard drive on your computer or elsewhere on a local-area network. If a deployment location is local, you specify it with a drive letter or in UNC format.   Click the Browse button to search for servers on your network.   Entering a value in this field sets a **Deploy_Path** initialization variable in SRCSAFE.INI.

Alternatively, the location can be remote, in which case it is reached via File Transfer Protocol (FTP). Use URL format to specify an FTP path.   The path must include the protocol specification ("ftp://") and the location.   To these you can add a username or a username and a password, and (if necessary) a folder path.   Separate username and password with a colon (:).   You can specify more than one path in this field; separate the paths with commas.

For example:

```
ftp://www.trassi.com/
```

```
ftp://SamBal@www.trassi.com/ssafe
```

```
ftp://SamBal:baDJak@www.trassi.com/ssafe
```

```
ftp://www.trassi.com/ssafe,\\my\server\ssafe
```

Visual SourceSafe stores the entire ftp path in the **Deploy_Path** variable.

Note   You can Deploy any project that has been designated as a Web project, regardless of the types of files it contains.   A single command then sends the entire project to the remote location or locations you've specified.

You can enter several deployment paths in this box, separated by commas.   For instance, you could set it up so that whenever a user deploys this web site, it will go to your ftp location, and also to a folder on your local server.

When defining a path for an FTP-site,   the folder information works from the FTP server's root. When using typical FTP commands on PC or UNIX, you normally logon to the FTP server and begin at a home path that's typically not the server root.   You may think of this as sort of a "home" root.

For example, if you wanted to drop a file into your web site at North Carolina State you would perform the following steps:

1. ftp ftp.eos.ncsu.edu
2. Enter your username and password.
3. Change folder to temp (a subfolder).
4. use the ftp "put" command to write the file.

To specify that location on in the Deploy dialogbox, you would have to type:

```
ftp://username:password@ftp.eos.ncsu.edu/usr/name/temp
```

Note the necessity of specifying /usr/name/ – because the deploy to FTP works from the server root and not the current folder.   Also note that many systems use a tilde (~) to specify the user's home folder.   So you may have to enter the following (where the UNIX server supports this common syntax):

```
ftp://username:password@ftp.eos.ncsu.edu/~/temp
```

Also, you should be careful where you enter the /.   For example "cd ~/temp" is a perfectly valid FTP command, but "cd /~/temp" is not.

Set Another

Applies the web site information specified in the other fields of this dialog box, and clears the dialog box so you can set another web site project.   This button is not enabled unless you have specified a valid project and URL in the top field.

OK

Confirms selections, and saves the web site settings to the SRCSAFE.INI file.
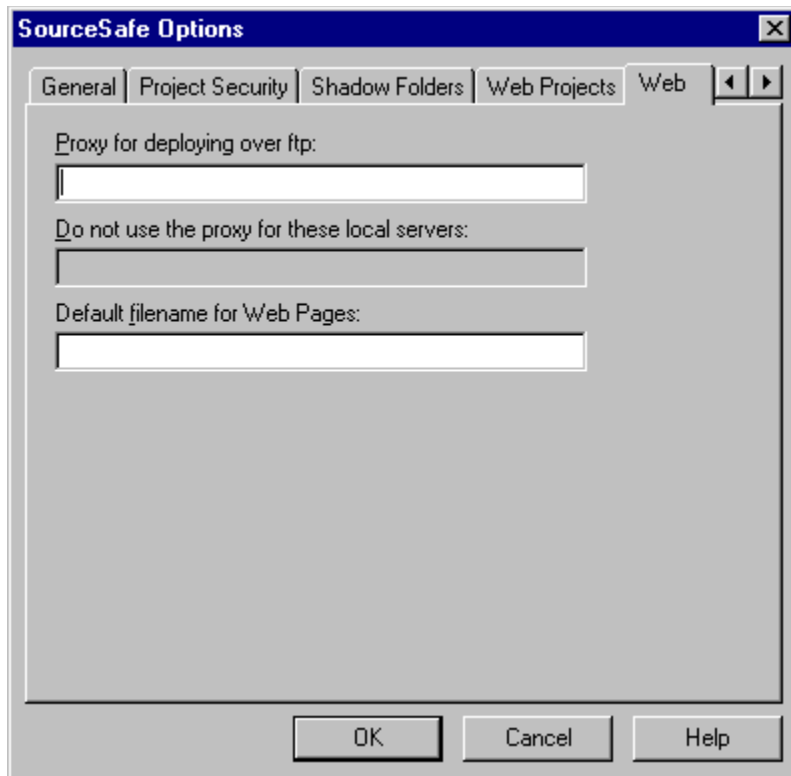
Cancel

Cancels selections and returns to Visual SourceSafe Administrator.

To Remove Web Project Designation

Clear the URL and virtual root fields on this tab; then click OK.

Web Options Tab

There are three general settings that apply to all your web projects in your Visual SourceSafe database at once.   Two apply to Web projects that are deployed via FTP through a firewall, and one applies to the Check Hyperlinks command.   You set these with the Web tab.   Don't confuse the Web tab with the Web Projects tab, which sets options for a specific web project rather than for all web projects at once.

Proxy for deploying over ftp

If you need to deploy remotely through a firewall, use this field to specify a proxy.   Sets the **Deploy_Proxy** initialization variable in SRCSAFE.INI.

Do not use the proxy for these local servers

Use this field to specify local servers for which the proxy is not appropriate because they are inside the firewall.   Sets the **Deploy_Host_Local** initialization variable in SRCSAFE.INI.

Default filename for Web pages

Use this field to specify the filename that Visual SourceSafe appends to a URL that ends in a folder name, during a Check Hyperlinks run.   Defaults to "DEFAULT.HTM"; you only need to specify this if you want something different.   Sets the **Web_Default_Filename** initialization variable in SRCSAFE.INI.

OK

Confirms selections, and saves the web settings to the SRCSAFE.INI file.

Cancel

Cancels selections and returns to Visual SourceSafe Administrator.

Web Site Option Tab See Also

Web Options Tab

Web Projects Options Tab

Deploy Command

Create Site Map Command

Check Hyperlinks Command

Deploy_Path Initialization Variable

URL Initialization Variable

SiteMap Initialization Variable

External_Link_File Initialization Variable

Open SourceSafe Database Dialog

Allows you to open different <u>Visual SourceSafe databases</u> and store their locations.

Dialog Box Options

Available databases

Specifies the databases you have already visited, giving friendly name and path information for each. If the database you want is not shown in the list, you can click the Browse button to look for it.   Once you find it, and click OK in the Browse dialog with a specific SRCSAFE.INI file selected, you have the opportunity to give the database a friendly name, easily remembered so you can re-connect later.

Username

Specifies the username with which you want to log into the new database.   If you change this to a new username with a different password, you will be prompted to enter the new password.   You cannot change this option in the Administrator program.

Open this database next time I run Visual SourceSafe

Check this box to designate this database as your startup database.

Remove

Removes a selected database from the list.   You cannot remove the current database from the list.

OK

Confirms your selections.

Cancel

Cancels your selections.

Open SourceSafe Database See Also

Open SourceSafe Database Command

Browse for Visual SourceSafe Database Dialog

See Also

Allows you to name a <u>Visual SourceSafe database</u>.

Dialog Box Options

Database path

Specifies the path to the selected database.   You cannot change this value here.

Database name

Specifies the friendly name you want to give this database.   This name is displayed in the Open SourceSafe Database dialog box.

OK

Confirms your selections.

Cancel

Cancels your selections.

Browse for Visual SourceSafe Database See Also

Open SourceSafe Database Command (File Menu)

Allows you to switch between Visual SourceSafe databases via the Visual SourceSafe Explorer.

Access Rights

You must have read permission to use this command.

Remarks

Use this command if you regularly need to access different **Visual SourceSafe databases**.   You will, of course, need access permissions on the server storing the database and in the database you are trying to access.   When you use this command, the current Visual SourceSafe database is closed and the newly specified one is opened.

You can give each database an easy-to-remember name, and each time you connect to a new database it is added to the list in the Open SourceSafe Database dialog box.

Open SourceSafe Database See Also

Open SourceSafe Database Dialog Box

Lock SourceSafe Database Command (Tools Menu)

See Also

Allows you to lock all users out of the Visual SourceSafe database while you run a utility, upgrade the executables, or backup the database.

Access Rights

You must have Administrator permission to use this command.

Remarks

Use this command if you need to archive, backup, or run a Visual SourceSafe utility on the database. Note that the act of checking the checkbox in the Lock SourceSafe Database dialog box is what actually locks users out.   Once this box is checked, no one will be allowed to log into the database. The users listed who are already logged in will be unaffected − you will need to notify them manually to log out.

Lock SourceSafe Database Command See Also

Lock Visual SourceSafe Database Dialog

Lock SourceSafe Database Dialog Box

See Also

Allows you to lock all users out of the Visual SourceSafe database while you run a utility, upgrade the executables, or backup the database.

Dialog Box Options

Lock all users out of Visual SourceSafe

Check this box to lock all users out of the **Visual SourceSafe database**.   Remember to uncheck the box when you are finished with your administrative duties!

Note that the act of checking the checkbox in the Lock SourceSafe Database dialog box is what actually locks users out.   Once this box is checked, no one will be allowed to log into the database.   The users listed who are already logged in will be unaffected − you will need to notify them manually to log out.

X users logged in

This section of the dialog box shows the users who are currently logged in.
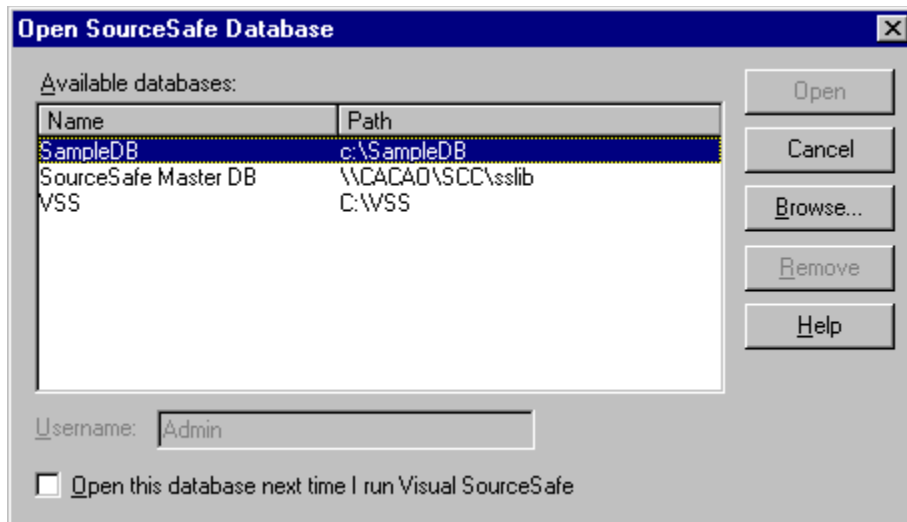
Close

Closes this dialog box.

Lock SourceSafe Database Command See Also

Lock Visual SourceSafe Database Command

Open Database Dialog Box

This dialog lets you browse for a SRCSAFE.IN file associated with a **Visual SourceSafe database**.

Dialog Box Options

File name

Enter the file name to search for.   When looking for a Visual SourceSafe database, use SRCSAFE.INI.

List files of type

Filter the file list box by displaying files only of the selected type.   SourceSafe Database is the default.

Folders

Select the folder to look in.

Drives

Select the drive letter to look on.

OK

Selects the database file listed in the File Name box.

Cancel

Cancels the browse operation.

Network

Lets you map a drive letter to a different network server, if necessary.

Open Dialog Box

Dialog Box Options

File Name

The name of the file you are looking for.   You can specify wildcards here, such as *.EXE, to look for more than one file matching the wildcard.

List files of type

Specify the type of files to include in the File name list box.

Folders

Specifies a folder.

Drives

Selects the drive that contains the folder.

OK

Confirms the file selection.

Cancel

Cancels the selection, closes the dialog box, and returns to <u>Visual SourceSafe Administrator</u>.

Network

Allows you to connect a network drive to search.

Set Shadow For Dialog Box

Lets you browse for a project to set a shadow folder for in your Visual SourceSafe database.

Dialog Box Options

Project

Select the project for which you want to set up a shadow folder.

OK

Accepts the project you have selected.

Cancel

Cancels the shadow project selection.

Topics

Project <project> has been destroyed, and cannot be rebuilt.

See Also

Visual SourceSafe may not be able to reproduce the project version you requested.   This error has the following cause and solution:

- You permanently destroyed a **<u>subproject</u>** or moved the subproject to another location.   For example, there may have been a subproject that you deleted or purged.   Later, you attempted to go back to an earlier version of the project.   Visual SourceSafe attempts to reconstruct the project exactly as it was in the past, but it cannot, because the subproject cannot be recovered.

   Click Yes, and Visual SourceSafe does its best to restore the old project version, but restoration is not guaranteed.

   Click No, and the attempt to restore an old project version is halted.

   Click Yes All, and Visual SourceSafe does not notify you that other selected files may also be unrecoverable, but continues with the best possible reconstruction.

See Also

Purge Command

Delete Command

File <file> has been destroyed, and cannot be rebuilt.

See Also

Visual SourceSafe may not be able to reproduce the file you requested.   This error has the following cause and solution:

- You permanently destroyed a file or moved the file to another location.   For example, there may have been a file in a project that you deleted or purged.   Later, you attempted to go back to an earlier version of the project.   Visual SourceSafe attempts to reconstruct the project exactly as it was in the past, but it cannot, because the file cannot be recovered.

  Click Yes, and Visual SourceSafe does its best to restore the old file, but restoration is not guaranteed.

  Click No, and the attempt to restore an old file is halted.

  Click Yes All, and Visual SourceSafe does not notify you that other selected files may also be unrecoverable, but continues with the best possible reconstruction.

An item with the name <name> already exists.

<u>See Also</u>

Each item in a Visual SourceSafe project must have a unique name.   This error has the following cause and solution:

- You have attempted to name a file or an immediate **<u>child project</u>** with the same name as the project.   (An item with that name can exist in other projects in the database, including subprojects.)

  To resolve this error, give the new item a different, unique filename.   Or, you can delete or rename the existing item.

Note   File and project names in Visual SourceSafe are not case-sensitive.   For example, if you have a file named TEST.C, you cannot create a subproject or file named "Test.c" or "test.c" in the same project.

See Also

Add Files Command

Delete Command

Rename Command

Share Command

File or project not found.

Visual SourceSafe is looking for a file or project that does not exist.   This error has the following cause and solution:

- You mistyped the file or project name, or specified a file or project that does not exist.
- The Visual SourceSafe database is corrupted. .

   Try retyping the name or select the name from the list in **Visual SourceSafe Explorer**.   If you suspect database corruption, run the ANALYZE.EXE utility.

See Also

Visual SourceSafe Explorer

Version not found.

The version specified cannot be found.   This error has the following causes and solutions:

- You specified a version that does not exist.

  In Visual SourceSafe, click the Show History command on the Tools menu, and then choose a version from within the History of File dialog box.

- If you are using Visual SourceSafe from the command line, you may not be using the -V parameter properly.

  Search online Help for more information on using the parameter correctly (click the Contents button).

See Also

Show History Command

Tracking Old Versions

-V Command-Line Option

This command only works on files.

This command can operate only on files.   This error has the following cause and solution:

- You are attempting to perform a file operation (such as using the View command) on a project. Select the proper command, or select a file that works with the command you've chosen.

This command only works on projects.

This command can operate only on projects.   This error has the following cause and solution:

- You are attempting to perform a project operation (such as the Move Project command) on a file. Select the proper command, or select a project that works with the command you've chosen.

Parent not found.

This error has the following cause and solution:

- Database corruption is indicated.

  Ask your **Visual SourceSafe administrator** to run ANALYZE.EXE to find and correct the problem.   If that fails, call Microsoft Technical Support for help with recovering the database.

See Also

Technical Support

<File> is not branched in <project>.

<u>See Also</u>

This error has the following cause and solution:

- The file you are trying to merge is not branched.

  When a file is branched, and you want to merge the different version of the file into one combined file, you can use the Merge Branches command.   In this case, the file is not branched, so Merge Branches has no effect.

Cannot move a project under itself.

See Also

A project cannot be a project of itself.   This error has the following cause and solution:

- You attempted to create an invalid loop in your project hierarchy.   For example, moving the project $/WORD under the project $/WORD, or, more subtly, moving $/WORD under $/WORD/APPS.   This creates an endless recursive loop, which causes this error.

  Restructure your project hierarchy so that there is no invalid loop.

See Also

Move Project Command

Projects

File not checked out by you.

Some Visual SourceSafe operations can only be performed on a file that is checked out.   This error has the following causes and solutions:

- You attempted to check in or undo a check out on a file that you did not have checked out.

  Check out the file before checking it in.   If you have made changes to a file, see the Advanced options in the Check Out dialog box, and then under Replace Writable, select Skip to check out the file to you without replacing the changes in your **working folder**.

- You attempted to perform an operation that can only be performed on a file that is checked out.

  Check out the file.

See Also

Check In Command

Check Out Command

File <file> does not retain old versions of itself.

See Also

This error has the following cause and solution:

- When you check in a file, Visual SourceSafe retains both the current version of the file, and the information required to rebuild past versions.   You have turned off the historical version storage for this file.

  If you explicitly turn off this change information in the Properties dialog box or when using the Add Files command, Visual SourceSafe can give you the most recent version, but it cannot rebuild old versions.

See Also

Check In Command

Tracking Old Versions

Invalid date string.

Visual SourceSafe cannot use the date syntax entered.   This error has the following cause and solution:

- The date you typed or entered is invalid.

  Proper date syntax is Month/Day/Year, with each number separated by a slash mark: for example, 2/29/64 or 11/5/1992.

Invalid time or date string.

Visual SourceSafe cannot use the date or time syntax entered.   This error has the following causes and solutions:

- The date you typed or entered is invalid.

  Proper date syntax is Month/Day/Year, with each number separated by a slash mark: for example, 2/29/64 or 11/5/1992.

- The time you typed or entered is invalid.

  Proper time syntax is Hour:Minute, followed by a one-letter A.M. or P.M. indicator: for example, 9:30A or 5:00p.   (The indicator is not case sensitive.)   Or, you can use a 24 hour syntax such as 13:45.

Invalid syntax on line <line> of file <file>.

See Also

This error has the following cause and solution:

- The syntax of one or more lines in the initialization file is invalid.
  Search Help for information on the proper syntax of the initialization variable.

Note   You can use any text editor to change or delete the invalid line.

See Also

The Initialization File Variables

Initialization variable <variable> set to invalid number.

<u>See Also</u>

The specified initialization variable has been set incorrectly.   This error has the following cause and solution:

- An initialization variable that should be set with an Arabic number is not.   For example, it may be set to "two" when it should be set to "2."

  Search Help for information on the proper syntax of the initialization files.

Note   You can use any text editor to change or delete the invalid line.

See Also

The Initialization File Variables

Initialization variable <variable> must be between <number> and <number>.

See Also

An initialization number is out of the valid range.   This error has the following cause and solution:

- An initialization variable has been set to an invalid number.
  Search Help for information on valid settings.

Note   You can use any text editor to change or delete the invalid line.

See Also

The Initialization File Variables

Initialization variable <variable> must be set to "Yes" or "No".

See Also

This error has the following cause and solution:

- An initialization variable that should be set to Yes or No is not.
  Change the variable so that it is set to Yes (or True) or No (or False).

Note   You can use any text editor to change or delete the invalid line.

See Also

The Initialization File Variables

Initialization variable <variable> set to invalid folder path.

<u>See Also</u>

An initialization variable that should be set to a valid folder path is set to an invalid path.   This error has the following cause and solution:

- The path syntax may be incorrect or the path may not exist.

   Verify that the path exists or that the syntax is correct and change it as necessary.

Note   You can use any text editor to change or delete the invalid path.

See Also

The Initialization File Variables

Initialization variable <variable> set to invalid value.

<u>See Also</u>

This error has the following cause and solution:

- An initialization variable is set to an invalid value.
  Search Help for information on the proper syntax.

Note   You can use any text editor to change or delete the invalid path.

See Also

The Initialization File Variables

Cannot find initialization variable <variable>.

<u>See Also</u>

Visual SourceSafe cannot find an initialization variable it needs to perform an operation.   This error has the following cause and solution:

- The specified initialization variable is missing and is required to complete the operation.

  Search Help or the *Microsoft Visual SourceSafe User's Guide* for information on the variable, determine a proper value for it, and add it to your initialization file.

See Also

Configuring Visual SourceSafe

Cannot find SS.INI file.

<u>See Also</u>

Visual SourceSafe cannot find your SS.INI file.   This error has the following causes and solutions:

- The SS.INI file is not where Visual SourceSafe expects to find it.   (See your Visual SourceSafe administrator if you have moved this file.)
- You have inadvertently deleted your SS.INI file.

  You can copy an SS.INI file from another user into your Visual SourceSafe folder, then modify it to customize Visual SourceSafe.   Your Visual SourceSafe administrator must then set Visual SourceSafe to look in the proper place.

See Also

The Initialization File Variables

Permission denied.

This error has the following cause and solution:

- You are attempting to take an action that you do not have the proper privileges to execute.

  Talk to your Visual SourceSafe administrator.   If you are the **Visual SourceSafe administrator**, run Visual SourceSafe Administrator to check on or change your username, password, and access rights.

User <user> not found.

This error has the following cause and solution:

- Your **<u>username</u>** was not found.

  Talk to your **<u>Visual SourceSafe administrator</u>**.   If you are the Visual SourceSafe administrator, run Visual SourceSafe Administrator to add your username.

Invalid password.

This error has the following cause and solution:

- Your **username**/**password** combination is invalid.

  Re-enter the password.   Because it does not appear on the screen, it is very easy to type it incorrectly.   Note that the Visual SourceSafe password is case-insensitive.   If you believe you have typed the password correctly, Visual SourceSafe doesn't recognize the password for some reason, so ask your **Visual SourceSafe administrator** to change your username and password. If you have forgotten your password, the administrator can use Administrator to assign you a new one.

Bad username syntax: <syntax>.

This error has the following cause and solution:

- You have entered a **username** with invalid Visual SourceSafe user syntax.
  Ask your **SourceSafe administrator** for your correct Visual SourceSafe username.

Invalid SourceSafe syntax: <syntax>.

The syntax you entered is invalid in Visual SourceSafe.   This error has the following cause and solution:

- You have entered a Visual SourceSafe file or **project** path that does not use valid syntax.

  In Visual SourceSafe syntax, every path starts with a dollar sign followed by a series of project names, which are separated by slashes and optionally followed by a filename.   For example:

- $/ is the **root project**.
- $/CODE is a subproject of $/.
- $/CODE/WIN/TEST.C is a file in the $/CODE/WIN project.

  Many parts of this syntax are optional.   You can omit the dollar sign in most circumstances.   You can also shorten the path by basing it on your current project; for example, if you are in $/CODE, you can type WIN as a shorthand for $/CODE/WIN, TEST.C for $/CODE/TEST.C, and ".." for the root.

Filenames are any valid operating system filenames; project names can include spaces and can be up to 256 characters long.   Project names are not case-sensitive (for example, to Visual SourceSafe, Code and CODE are the same project).

See Also

SourceSafe Naming Syntax

This command operates on only one item.

For many Visual SourceSafe dialog box lists, you can choose more than one item at once, and then act on all the items together.   Some actions, however, cannot operate on more than one item at a time.   This error has the following cause and solution:

- You selected more than one item and then applied a command that operates on only one item at a time.   For example, you may have tried to use the View command, which works with one file at a time.

  Press the space bar on an item to make it the only item selected in the list.

You cannot use wildcards with this command.

Many Visual SourceSafe commands accept **wildcard characters**, such as * and ?.   But some commands, such as the View command, can operate on only one item at a time.   This error has the following cause and solution:

- The command you have chosen does not accept wildcard characters.
  Use an explicit filename, with no wildcard characters.

No items listed to operate on.

This error has the following cause and solution:

- You are attempting to perform an action without first selecting an item to operate on.
  Click an item to select it, and then try the action again.

Network not found.

This error has the following cause and solution:

- No network is installed, or Visual SourceSafe cannot find a valid network.

  Perhaps there is no network installed, or one or more network connections have failed. Check your network hardware, and make sure that all the necessary network drivers are running.   Check with your Visual SourceSafe administrator for additional assistance.

\<File\> is an invalid or corrupted logfile.

See Also

This error has the following cause and solution:

- Visual SourceSafe cannot find or read your logfile.

  Run ANALYZE.EXE with the -F option.   If necessary, check with Technical Support for more assistance.

See Also

Technical Support

Corrupted history in file <file> after revision <number>.

This error has the following cause and solution:

- The Visual SourceSafe database has been corrupted.

  DDCONV.EXE saves as much of the database as possible, but some information will be irretrievably lost (unless you have backed up the database and can recover the information from there.)

Illegal file or project name(s): <file>".

See Also

This error has the following cause and solution:

- You have entered an illegal file or project name.

  Check to make sure that the name was entered correctly.   If you are unsure which characters are allowed in a SourceSafe name, check the Visual SourceSafe Naming Syntax and Conventions topic.

See Also

Visual SourceSafe Naming Syntax and Conventions

Illegal version syntax: <number>.

This error has the following cause and solution:

- The version number you entered uses illegal syntax.
  Check to make sure that the information was entered correctly.

An item named <item> was already deleted from this project.

See Also

In Visual SourceSafe, you can delete only one instance of a file or project with a given name in a given project.   This error has the following cause and solution:

- You deleted a file, created a new file with the same name, and then attempted to delete that file as well.

  You can choose Yes to purge the old file, or No to cancel the delete operation.   If you purge the original file, you will lose all historical information in it, but you can then continue with the current delete operation.   Or, you can choose No, and then rename the new file before deleting it, if you want to keep the file history of the old file.

See Also

Purge Command

Delete Command

Rename Command

Cannot delete, rename, or move the root project.

This error has the following cause and solution:

- The Delete, Rename, and Move Project commands cannot be applied to the **root project** ($/),
  which is analogous to the root folder of your hard disk (C:\).

  Since the root project cannot be changed in any way, consider restructuring your project.

\<file\> is already checked out by \<user\>.   Continue?

This warning has the following cause and solution:

- You attempted an operation on a Visual SourceSafe file that is checked out by another user.

  Visual SourceSafe gives you the option to continue, which means any differences between the changes you make to the file and the changes made by the other user will need to be merged in the future.   Click Yes to continue your operation and create a multiple-checkout situation; answer No to cancel the check out action.

See Also

Check Out Command

Destroy Command

Delete Command

Rename does not move an item to another project.

See Also

The Rename and Move Project commands cannot be implemented at the same time.   This error has the following cause and solution:

- The Visual SourceSafe Rename command and Move command are separate, distinct operations; you cannot do both in one step.

  First perform one action and then do the other.

See Also

Move Project Command

Projects

Rename Command

File <file> is currently checked out by <user>.

See Also

This error has the following cause and solutions:

- When someone other than you has a file checked out, there are some things you cannot do to the file.   For instance, you cannot delete or rename a file that is checked out.

  Click Yes to check out the file anyway.   (Yes All checks out all selected files even if they are already checked out by another user.)   If you choose yes, there may be conflicts between your changes and those of another user, which will have to be resolved and merged later.

  Click No to cancel the Check Out command.   (No All cancels the command for all selected files.)

If you want to work on a checked-out file, it's best to check the User column of **SourceSafe Explorer** for the name of the user who has it checked out and ask that user to check it in as soon as possible. If the user is unavailable, the **Visual SourceSafe administrator** can undo the check out for you.

See Also

Undo Check Out Command

Cannot rename to <file>; that name already exists.

The filename specified already exists in the project.   This error has the following cause and solution:

* When you rename a file that is shared by many different projects, the name of the file changes in all the projects that are sharing it.   If any one of these projects already has a file with the same name, you create an ambiguous and invalid situation, two files in the same project with the same name, and the Rename operation fails.

  To correct this, choose another name for the file.

See Also

Rename Command

Shared Files

Project <project> does not exist.

This error has the following cause and solution:

- You are attempting to move a project to a **parent project** that does not exist.

  Check the path of the parent project and try again.   If you are unsure of Visual SourceSafe project syntax, try selecting the parent from the **project list** in **SourceSafe Explorer**, and then note the project path displayed above the **file list**.

Note   To move a file or subproject to a new parent, you must have already created the new parent project.   Use the Create Project command to create a new project.

Path <path> too long.

The path specified is too long.   This error has the following cause and solution:

- You typed a Visual SourceSafe path that is too long.

  Visual SourceSafe project names are limited to 64 characters.   Full project path names can be no longer than 255 characters.

See Also

SourceSafe Naming Syntax

This version of <name> already has a label: overwrite?

Any version of a Visual SourceSafe file or project can have a **label**, which is used to identify that **version**.   This error has the following cause and solution:

- One version of a file or project cannot have more than one label.   Therefore, when you attempt to label a version that already has a label, Visual SourceSafe displays this message.

  Click Yes to apply the new label, and the old label is replaced.   Click No to cancel the action, and the old label is left in place.

This label <label> is already used.   Remove the old label?

A Visual SourceSafe **label** is used to identify a particular **version** of a file or project.   This error has the following cause and solution:

- You are attempting to assign a label to a file or project, and another version of that file or project has the same label.

  Click Yes, the new label is applied, and the old label is deleted. Click No, and the old label is not changed.   Click Cancel to cancel the labeling operation.

File <file> is already shared with this project.

See Also

This error has the following cause and solution:

- You are attempting to share a file with the **current project**, and that file already exists in the current project.

  If you want to share a different version of the file, specify a version in the History of File dialog box and click Pin.   If you want to share this file with another project, select the project to share the file into in Visual SourceSafe Explorer and on the SourceSafe menu, click Share. Then, select the file to share from the Share dialog box.

See Also

Show History Command

Shared Files

Tracking Old Versions

-V Command Line-Option

Cannot check out an old version of a file.

<u>See Also</u>

The Check Out command always uses the most recent version of a file.   This error has the following cause and solution:

- You are trying to check out an old version of a file.

    If you want to use and modify a previous version, take the following steps:

    1. Check out the most recent version of the file.
    2. Delete it from your **<u>working folder</u>** (not the Visual SourceSafe database).
    3. Get the old version that you want to modify.
    4. Make your changes to the old version.   (This may require removing its read-only flag manually.)
    5. Check the modified file into Visual SourceSafe.

Note   You can also use the Rollback button in the History of File dialog box (on the Tools menu, click Show History).   If you roll back to a certain version, all versions subsequent to that version are deleted, and you can check out the file and start modifying it.   The disadvantage of this method, however, is that you lose all the subsequent versions, and cannot retrieve them.

See Also

Check In Command

Check Out Command

Get Latest Version Command

Rollback Command

File <file> is checked out to project <project>, and you are in <project>.

When a file is shared among multiple projects, a file in one project is linked to all other projects that share it.   Therefore, when you check out the file from one project, it is also checked out from all the projects that share it.   This warning has the following cause and solution:

- You checked out a file from one project, and you are attempting to check the file in or undo the checkout from another project that shares the same file.

   This is a valid action−Visual SourceSafe is simply alerting you in case this is not what you intend. To proceed, click Yes; to cancel the operation, click No.

Note   You can check in a file or undo a checkout from any project that shares a file.

File <file> was checked out to folder <path>.

When you check out a file, Visual SourceSafe keeps track of the **working folder** you checked it out to.   In general, you make your modifications in that folder, and then check in the file or undo the checkout from the same folder when you are done.   This warning has the following cause and solution:

- You are attempting to check in or undo a checkout of the file from a folder different from the one you checked it out to.

  Your working folder for a project is displayed just above the file pane in **SourceSafe Explorer**, and can be changed with the Set Working Folder command from the File menu (CTRL+D); the **check out folder** of a file is displayed in the **file list**.   (The check out folder may be different from your working folder assigned to the project, if someone else has the file checked out, for example.) Although you can check a file in from a folder other than the one you checked it out to, make sure that this is what you intend.

File <file> has not been changed.

This error has the following cause and solution:

- You are attempting to check in a checked out file that you have not changed.

  Click Yes, and Visual SourceSafe checks it in anyway.   (Clicking Yes All checks in all selected files.)   A check in creates a new version of the file, with an optional comment.

  Click No, and Visual SourceSafe undoes the check out of the file.   (Clicking No All undoes the checkout for all selected files.)   No new version of the file is created in this case.

Note   To avoid this message on future occasions, you can turn off this notification on the General tab in the SourceSafe Options dialog box on the Tools menu.

A writable copy of <file> already exists.

When you get a Visual SourceSafe file, the file is tagged with the **read-only** flag — you cannot modify the file.   When you check out the file for modification, the read-only flag is deleted.   When you later check it in, the file is flagged read-only again.   This error has the following cause and solution:

- You attempted to use the Get Latest Version command on a file that already exists on your hard disk in writable form.   If the **local copy** is writable, you may have it checked out, or are otherwise modifying it; Visual SourceSafe does not replace the file, and displays this error message.

  Click Yes or Yes All to replace the file or files with the current version is the Visual SourceSafe database.

  Click No or No All to not replace the file or files.

  Click Cancel to cancel the Get Latest Version command.

File <file> is not shared with any other projects.

The Branch command is used to break a **<u>share link</u>**.   This error has the following cause and solution:

- The file you've chosen to apply the Branch command to has no link to other projects.
  Choose another command, or choose a shared file to apply the Branch command.

You are using an old version of <file>, and cannot change it. Rollback to make that version the most recent if you need to change it.

See Also

This error has the following cause and solution:

- You tried to modify an old version of a file.

  If you want a project to always use a specific version, and not receive checked in changes made in other projects, you can share a specific version of the file using the Pin command in the History of File dialog box.

  If you want to start modifying the file from an old version, you can roll back the file.   The Rollback command only affects the current project; it performs an implicit separate operation.   Both branching and roll backs are available from the History of File dialog box.

See Also

Rollback Command

Tracking Old Versions

Cannot roll back to the most recent version of <file>.

The Rollback command returns a file to a previous version of itself.   This error has the following cause and solution:

- You have attempted to roll back a file to its most recent version or have not specified a version to roll back to.

  You cannot roll back a file to the current version in Visual SourceSafe. If you meant to specify an earlier version, retry the command with the earlier version specified.

See Also

Rollback Command

A deleted link to <file> already exists.

A deleted file remains part of a project, but it doesn't appear in the **file list**.   This error has the following cause and solution:

- You are attempting to share a file from another project with the **current project**.   However, the file already exists in the current project.   It doesn't appear in the file list because it has been deleted.

  To make the file appear in the file list, don't share it from another project; rather, recover it in this one using the Recover button on the Deleted Items tab in the Properties dialog box (File menu).

See Also

Recover Command

Delete Command

Shared Files

Folder <path> not found. Create?

The specified **working folder** does not exist.   This error has the following cause and solution:

- The folder may have been deleted, or you may have entered its specification incorrectly.   Your network connection to a networked location may also have gone down.

  Reset the working folder for this project (using the Set Working Folder command on the File menu), or click Yes to create the folder.   This tells Visual SourceSafe that this is the local working folder.   When you open the project again, Visual SourceSafe automatically goes to this working folder.

  If your working folder seems fine, check to see if this project is shadowed to another location, and that your connection to that location is still OK.

<item> specifies a version number, which is illegal for this command.

This command operates only on the most recent version of a file.   This error has the following cause and solution:

- You attempted to execute the command on an older version of the file.

  Choose another command, or apply the command you've chosen to the most recent version of the file.

This command operates only on projects, not on files (like <file>).

This command operates only on **projects**.   This error has the following cause and solution:

- You attempted to execute the command on a file.
  In the **SourceSafe Explorer**, select a project in the **project list**, then try the command again.

This command operates only on files, not on projects (like <project>).

This command operates only on files.   This error has the following cause and solution:

- You attempted to execute the command on a **project**.
  In the **SourceSafe Explorer**, select a file in the **file list**, then try the command again.

\<file\> is not a text file.

Although most Visual SourceSafe commands work on either text or binary files, this command does not.   This error has the following cause and solution:

- You attempted to execute a command, such as View File on the Edit menu or Find In Files on the Tools menu, that operates only on text files.

  Binary file types are set either automatically by Visual SourceSafe or manually using the File Types tab in the SourceSafe Options dialog box (Tools menu).

Invalid SourceSafe path <path>.

<u>See Also</u>

This error has the following cause and solution:

- You have entered an invalid Visual SourceSafe project or filename.

  Check to make sure that you typed the path correctly, or use **<u>Visual SourceSafe Explorer</u>** to choose the item from the **<u>project list</u>** or **<u>file list</u>**.

  If you chose the item from Visual SourceSafe Explorer, it is possible that another user has deleted or renamed the item since you last used Visual SourceSafe.   Collapse the **<u>root project</u>** and re-expand it to refresh the **<u>project pane</u>**.

See Also

Delete Command

Rename Command

<item> is not a deleted file or project.

The Recover and Purge commands operate only on items that have been deleted.   This error has the following cause and solution:

- You attempted to use the Recover or Purge command on an item that has not been deleted.

  If you are trying to recover a file or project, there is no need, as the file or project already exists in Visual SourceSafe.   To view the file, use the View File command.

See Also

Delete Command

Destroy Command

Purge Command

Recover Command

\<item\> has been deleted.

See Also

This error has the following cause and solution:

- You are trying to act on a file or project that has been deleted in the current project.

  Before you can do anything to the file, you must recover it using the Recover command.

  If you want to delete the file from the Visual SourceSafe database permanently, on the File menu, click Properties, and then on the Deleted Items tab, click Purge.

See Also

Purge Command

Recover Command

Delete Command

\<File\> is checked out by \<user\>; deleting it will cancel the check out.   Continue?

This warning has the following cause and solution:

- You are attempting to delete a checked out file.

  This warning provides the opportunity to correct from deleting the wrong file. If the filename is correct, click Yes to proceed. Click No to cancel the delete operation.

Destroy cannot be undone; information will be lost permanently!

See Also

This warning has the following cause and solution:

• When you use the Destroy command on a file or project, the action is not reversible.

 Click Yes to continue with the Destroy operation. Click No to cancel the operation.

Note If you do not want to receive this message every time you destroy a file or project, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

See Also

Destroy Command

Purge Command

Delete Command

Delete all specified items?

This warning has the following cause and solution:

- You are about to delete a file or project.

  Click Yes, and Visual SourceSafe deletes the file or project.

  Click No, and Visual SourceSafe does not.

  Use the Recover command to undo Delete.

Note   If you do not want to receive this message every time you delete a file or project, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

See Also

Recover Command

Purge cannot be undone; information will be lost permanently!

See Also

This warning has the following cause and solution:

- When you purge a file or project, the action is not reversible.

  Click Yes to continue with the Purge operation. Click No to cancel the operation.

  If you do not want to lose the file or project permanently, use the Delete command instead.

Note   If you do not want to receive this message every time you delete a file or project, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

See Also

Destroy Command

Purge Command

Delete Command

Rollback cannot be undone; some versions will be lost irretrievably!

This warning has the following cause and solution:

- You are about to roll back a file to an earlier version.   The Rollback command is not reversible.
  To make the old version the most recent version without losing all the versions in between:
  1. Check out the file.
  2. Delete the **local copy.**
  3. Get the version you want to roll back to.
  4. Check that version into Visual SourceSafe.

Note   If you do not want to receive this message every time you roll back a file, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

See Also

Rollback Command

<file> has changed.   Undo check out and lose changes?

See Also

This warning has the following cause and solution:

- You are about to undo the check out of a file.   If you do so, you discard all the changes made since you checked out the file.

  If you want to put your changes into Visual SourceSafe instead of losing them, use the Check In command instead.

Note   If you do not want to receive this message every time you undo a checkout on a file, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

See Also

Check In Command

Undo Checkout Command

This will end your SourceSafe session.

This message has the following cause and solution:

- You are about to exit Visual SourceSafe.

  Click OK to exit Visual SourceSafe, or click Cancel to return to Visual SourceSafe.

Note   If you do not want to receive this message every time you exit, you can turn it off on the Warnings tab in the SourceSafe Options dialog box (Tools menu).

There are no deleted items in this project.

This error has the following cause and solution:

- There are no items in this project that have been deleted.
  Because no files have been deleted from this project, there are no files that you can recover or purge.

The files are identical.

This message has the following cause and solution:

- The files you have selected, the Visual SourceSafe **master copy** and the **local copy**, are identical. You may have selected a file in the **file list** and tried to run the Show Differences command only on that file, in which case the file is being compared to itself and is by definition identical.

  In this case, before applying the Show Differences command again, specify two distinct files.

No search pattern was given.

This error has the following cause and solution:

- You have not entered a string or pattern that Visual SourceSafe can use to search through files. Enter a string in the Find In Files dialog box and then click OK, or click Cancel to exit the dialog box and return to Visual SourceSafe Explorer.

No matches were found.

This error has the following cause and solution:

- Visual SourceSafe has searched all the selected files, and has found no occurrences of the string you are trying to find.

  Check to make sure that you've entered the string correctly.

Always share the latest version of <file>?

Generally, every project sharing a file always has the latest version of the file.   When you check in the file in any project, all the projects are automatically updated.   However, you can use the Pin button in the History of File dialog box to share an explicit version of a file.   This message has the following cause and solution:

- By clicking the Pin button in the History of File dialog box, you've chosen to share the most recent version of the file.

   If you click No, the project shares the pinned version explicitly, and does not receive checked-in changes from other projects.

   If you click Yes, the project returns to a sharing state in which it is always using the latest version of the file.

See Also

Pin Command

Shared Files

Tracking Old Versions

Unpin Command

You must specify a new name.

This error has the following cause and solution:

- You are attempting to rename a file or project without specifying a new name for it.

  Either specify a new name and then click OK, or press the ESC key to close the dialog box without renaming the file or project.

Binary files differ.

This message has the following cause and solution:

- The binary files you've selected differ.
  The Show Differences command does not display line-by-line differences for binary files. Therefore, Visual SourceSafe can only tell you whether or not the Visual SourceSafe **master copy** is different from your **local copy**.

The passwords don't match.

This error has the following cause and solution:

- The **password** you typed in the Password box does not match the one typed in the Verify box.

  If the passwords you type are identical, Visual SourceSafe assumes you typed them correctly, and changes your password.   If you type two different passwords, Visual SourceSafe assumes that you mistyped the password; you must re-enter it.

Get the entire project containing this file version?

This message has the following cause and solution:

* You have selected a file in the History of File dialog box list and then clicked Get.

  Click Yes, and Visual SourceSafe gets the entire project as it existed immediately after the selected file was checked in.

  Click No, and Visual SourceSafe gets only the file version you selected, but doesn't get any other files in the project.

See Also

Get Latest Version Command

Running more than one instance of the SourceSafe Explorer or the SourceSafe Administrator is not supported under 16-bit Windows.

This error has the following cause and solution:

- You can run only one version of SourceSafe at a time on a 16-bit platform.

  Do not try to run a second instance of Visual SourceSafe.   Running multiple instances of Visual SourceSafe is supported under Microsoft Windows NT and Microsoft Windows 95.

File <file> not found.

Visual SourceSafe could not find the file it was looking for.   This error has the following cause and solution:

- You may have mistyped a filename in **Visual SourceSafe Explorer** or in an initialization file variable.
- The file may not exist.

  Make sure that you've correctly entered the name of the file.

Too many file handles open.

This error has the following cause and solution:

- There are too many files open on your system.

  You may want to check whether any of the files can be closed, or you may want to increase the Files count in your operating system's initialization file.

Access to file <file> denied.

At certain times, Visual SourceSafe denies a user access to a file.   This error has the following causes and solutions:

- You were denied access (read or write privileges) to a file that Visual SourceSafe was attempting to open.

  It is possible that another user has the file momentarily locked; try again.

- You may not have sufficient network privileges to take the action you are attempting.

  See your **Visual SourceSafe administrator** for more information on your Visual SourceSafe **access rights**.   Also see the Administration chapter of the *Microsoft Visual SourceSafe User's Guide* for a description of the network access rights required for various users in different directories.

Invalid drive: <drive>.

This error has the following cause and solution:

- The drive you are attempting to access is invalid.   You may have mistyped the drive specification, either in **Visual SourceSafe Explorer** or in an initialization file.

  Check to make sure that you've entered the correct the drive specification.

File <file> already exists.

This error has the following cause and solution:

- You tried to create a file or folder with a name that already exists.

  You can give Visual SourceSafe files the same name as many times as you want in different directories, but only once in a given folder.   Select a different name or rename the conflicting file or project.

Disk full.

This error has the following cause and solution:

- You tried to save data to a full disk.
  Free some disk space by deleting or moving unneeded files.

Invalid filename: <file>.

This error has the following cause and solution:

- You typed a filename with invalid syntax.

  Check your typing, or select a different name with valid characters. You may also get this error if you have multiple files selected and try to use a command meant for a single file, such as View.

Error reading from file.

Visual SourceSafe cannot read from a file it has opened.   This error has the following cause and solution:

- You may have lost your network connection.
  Exit Visual SourceSafe, and then check all network connections before restarting.

Error writing to file.

Visual SourceSafe cannot write data into a file that it has opened. This error has the following cause and solution:

- You may have a full disk.
  Exit Visual SourceSafe, and then free disk space by deleting unnecessary files.

Out of memory.

You do not have sufficient computer memory to perform this operation.   This error has the following cause and solution:

- You have too many applications, documents, or source files open.
  Close any unnecessary applications, documents, or source files that are open.
- You have many device drivers loaded.
  Eliminate unnecessary device drivers.

You have <file> checked out; deleting it will cancel the check out.

This warning has the following cause and solution:

- You are attempting to delete a file checked out to you.

  This warning provides the opportunity to avoid deleting the wrong file.   If the filename is correct, click Yes to proceed.   Click No to cancel the delete operation.

Folder not found.

This error has the following cause and solution:

- You specified a operating system folder that could not be found.

  Check to make sure that the folder specification is correct. Check capitalization on case-sensitive operating systems, and re-try.

Cannot find Help file <file>.

This error has the following cause and solution:

- The Help file isn't available.

  Try reinstalling Visual SourceSafe to get a copy of the Help file.   If the file is still missing, call Technical Support.

See Also

<u>Technical Support</u>

Too many file handles open.

This error has the following cause and solution:

- You have too many files open at once..
  Close some open files.

File <file> is locked.

This error has the following cause and solution:

- The file you are trying to access is already being used.
  Wait and try your operation again later.

An automatic merge has occurred and there are conflicts.   Edit <file> to resolve them.

See Also

Changes you made to the file since you checked it out conflict with other changes made (and checked in) to the source file by another user.   This error has the following cause and solution:

- You have to resolve the conflicts between your changes to a file and another user's changes before you can check in the file.

  Open the file in your **working folder** and search for six consecutive equal signs (======).   Visual SourceSafe uses these to mark the location of a conflict.

You may prefer to use the Visual Merge feature to make conflict resolution easier.   To keep this situation from arising again, your **Visual SourceSafe administrator** can disable multiple check outs in your project.

See Also

<u>Resolving Conflicts</u>

You do not have access rights to <command name>.

This error has the following cause and solution:

- Your Visual SourceSafe access rights are insufficient for the command you are attempting to execute.   For example, if you are a read-only user, you can use the Get Latest Version command to view files, but you cannot use the Check Out command to modify them.

  If you believe you have received this message in error, contact your **Visual SourceSafe administrator**.

Version <NUMBER> is identical to version <NUMBER> of <file>.

This error has the following cause and solution:

- The two versions you are comparing are identical.
  Check to make sure that the two versions are the ones you want to compare.

No additional Help is available on this item.

See Also

- No more information is available on this dialog box.   If the cause of the problem is not clear from the message, please call Technical Support for more assistance.

See Also

Technical Support

User <user> already exists.

This error has the following cause and solution:

- You attempted to add a user with a **username** that already exists.   You may want to choose another name; for instance, if there is already a Joyce, add JoyceF.

  Note that names are not case-sensitive, so if you already have a Joyce, adding JOYCE generates this error.

Cannot delete/rename/edit the Admin user.

This error has the following cause and solution:

- You attempted to delete, rename or change the Admin user.   The **Visual SourceSafe administrator** must always exist in your Visual SourceSafe Administrator user list, although you should not use Admin as an actual user.

  You can, however, change the Admin user's password, used to **logon** to Visual SourceSafe Administrator.

Cannot check out <file>. It is binary and is already checked out.

This error has the following causes and solutions:

- The file you specified is binary and already checked out.   Visual SourceSafe does not allow multiple check outs of binary files, because their changes cannot be merged.

  Wait to check out the file until after the other user has checked it back in.

Are you sure you want to delete?

This message has the following cause and solution:

- This is the standard delete warning displayed by Visual SourceSafe.
  Click Yes, and Visual SourceSafe deletes the item(s).
  Click No, and Visual SourceSafe cancels the delete command.

Filter string exceeds the maximum length of <number>.

This message has the following cause and solution:

- The combined length of the filter values on the File Types tab in the SourceSafe Options dialog box (Tools menu) are too long.

  SourceSafe uses the File Types Included in File Group field to create a filter string for the Share and Add Files dialog boxes.   This error occurs when the combined length of the file groups are too long.   The total buffer length is 511 characters.   To correct this error, shorten your file type lists.

Invalid value <value> on File Types tab.

This message has the following cause and solution:

- You have attempted to create a file type using invalid characters.

  File Type names cannot contain the following characters:[ ] ( ) , ;.   The value for a file type is a search pattern, and may not contain invalid characters.   Also, you cannot leave the "File type included in this group" edit box blank − some file types must be entered.

No application is associated with this file type.

This message has the following cause and solution:

- You have attempted to view a file for which no file association exists.

  This error is usually caused by trying to view a binary file. Check that you have selected the correct file, and if you have, associate an editing program with this file extension by using either the Windows95 Explorer or the File Manager.

Have the conflicts in <file> been properly resolved?

This message has the following cause and solution:

- You have attempted to check in a file changed from the current version stored in SourceSafe .

  This warning message often is displayed when you have checked out the same file to more than one user or computer. It is displayed so you can be sure that any merge conflicts have been resolved. In some cases, this warning may be displayed even if no merge conflicts occurred, and that is potentially confusing. But, if this is the case, just click Yes and continue with your work.

  If there were conflicts, and you did not open the source file to resolve them, click No and do so now. When finished, re-try your Check In operation.

The passwords do not match.

This error has the following cause and solution:

- The **password** you entered for the user did not match.

  When changing a password for a user, you must type the new password twice to guard against typing errors. In this case, the passwords you typed did not match. Try typing them again.

See Also

Technical Support

Out of memory - the rights changes could not be saved.

You do not have sufficient computer memory to perform this operation.   The changes you made to the user's rights have not been saved.

This error has the following cause and solution:

- You have too many applications, documents, or source files open.
  Close any unnecessary applications, documents, or source files that are open.
- You have many device drivers loaded.
  Eliminate unnecessary device drivers.

Path <path> does not exist.

This message has the following cause and solution:

- You have attempted to open a file for which the specified path does not exist.

  Check the path and try re-typing it again. Make sure you are on the computer to which the path applies.

You cannot change the Admin name or access privileges.

This message has the following cause and solution:

- You have attempted to change the Admin user's name or access privileges.
  The Admin user's name and access rights cannot be changed. Select another user on which to use this command.

Cannot edit the Admin user.

This message has the following cause and solution:

- You attempted to edit the Admin user's access rights.

  The Admin user's name and access rights cannot be changed. Select another user on which to use this command.

The editor '<editor>' was not found.

This message has the following cause and solution:

- You have attempted to view or edit a file for which the specified editor could not be found. (Editor associations to file extensions are set on the General Options tab.)

  Click Yes if you would like to specify a different editor. Click No to cancel the editing operation.

Editor path is a required field when prompting by editor is selected.

This message has the following cause and solution:

- When you check the Use Editor to Prompt for Comments checkbox, you must specify the editor to use.
  Set the path in the Editor field of the Command-Line Options tab of the Tools Options dialog box.

The folder '<folder>' already exists.

Each item in a Visual SourceSafe project must have a unique name.   This error has the following cause and solution:

- You have attempted to name a file or an immediate **child project** with the same name as the project.   (An item with that name can exist in other projects in the database, including subprojects.)

- You are using both a 16-bit version and a 32-bit version of Visual SourceSafe for the same project and filenames are not being mapped accurately.

  To resolve this error, give the new item a different, unique filename.   Or, you can delete or rename the existing item.

Note   File and project names in Visual SourceSafe are not case-sensitive.   For example, if you have a file named TEST.C, you cannot create a subproject or file named "Test.c" or "test.c" in the same project.

There are no items in the list.

This error has the following cause and solution:

- You specified a history operation where no items met the specified criteria.

Examine the check boxes in the history options dialog box, and make sure they are set correctly.

Label comment is invalid when no label specified.

This error has the following cause and solution:

- You specifed a comment for a label   but no label was supplied.

If you want to label a file or project, and wish to include a comment, you must specify the label as well as the comment.

This command cannot complete without a specified working folder.

This error has the following cause and solution:

- You choose a command that requires a working folder, without having set a working folder.

Choose OK to set a working folder for the current project now, or Cancel to cancel the command. You can also set a working folder at any time by selecting the project and then choosing the Set Working Folder command from the File menu.

No application is associated with this file type.

This error has the following cause and solution:

- You choose to view a file for which a file association doesn't exist.   Without this association, SourceSafe doesn't know which editor to use to display the file.

Use File Manager to add an association from this file to the appropriate editor, then try your command again.

If you want to override the File Manager association for a specific file type, you can enter specific associations in your SS.INI file, for example:

```
.C = C:\VC\VC.EXE
.FRM = C:\WINDOWS\NOTEPAD.EXE
```

Set <folder> as your personal working folder for the project <project>?

This message has the following cause and solution:

- You need to establish a working folder for all of your projects. This message is giving you the option to create the working folder immediately, rather than having to create it later using the Set Working Folder command on the File menu.

Click Yes to set the working folder for this project. Click No to skip this step for now.

Rollback is not allowed before a pinned version.

This message has the following cause and solution:

- You tried to roll back a file to a version before the one it is pinned to.

A pinned file cannot be rolled back to a version previous to the version it is pinned to.   To rollback the file, unpin it first, then do the rollback.

<file> is an Explorer link file.

This message has the following cause and solution:

- You dragged a .LNK file into Visual SourceSafe.

When you drag a .LNK file into Visual SourceSafe, SourceSafe needs to determine if you want the .LNK file itself added to the database, or the file the .LNK file points to. If you want to add the file referred to by the .LNK file, click Yes. Otherwise, click No.

Could not find the file referred to by shortcut <file>.

This message has the following cause and solution:

- The file refered to by this .LNK file could not be found.

Check that the file has not been moved or deleted. Click OK to accept and retry.

File <file> was checked out to <folder>, not to <folder>.

This message has the following cause and solution:

• The file you are checking in was checked out in a different folder.

When dragging a file to check in, you must check it in from the folder where it was checked out. The file you dragged will not be checked in.   If you want to check the file in from this folder, use the Visual SourceSafe Check In command from the SourceSafe menu.

<type> is not a valid mask.

This message has the following cause and solution:

- You specified an invalid mask in the Tools Options File Types tab.

You must specify a * or ? character in your file mask, and you cannot have any illegal characters such as (,) [ ] in the mask.

Retype the file mask using legal characters.

All conflicts have been resolved, would you like to save the file?

This error has the following cause and solution:

- You have successfully resolved all merge conflicts.

Choose Yes to save the file, or choose No to discard your changes.

Some conflicts have not been resolved, would you like to save the file anyway?

This error has the following cause and solution:

- Not all the conflicts in the merged file have been resolved.

Choose Yes to save the unresolved file, or no to discard your changes and exit.

Cannot difference the specified item <item>.

This error has the following cause and solution:

At least one of the items you selected to compare with another item cannot be differenced by Visual SourceSafe.

No deployment path has been setup for <project>.

This error has the following cause and solution:

- You tried to Deploy a project without specifying a deployment path for the project in the Administrator program.

Ask your Visual SourceSafe administrator to set up a deployment path for your web project.

The address <address> is not a valid FTP address.

This error has the following cause and solution:

- You specified an invalid FTP address.

Re-type the address, checking for misspellings and correct addressing.

Unable to open a connection to the FTP host <host>.

This error has the following cause and solution:

- Visual SourceSafe was unable to open a connection to the specified FTP host.

Check your network connections, and check to be sure you entered the correct syntax for your FTP host.   This error could also be caused if your FTP host is not available at this time.

Unable to deploy <project> to the FTP host <host>.

This error has the following cause and solution:

- Visual SourceSafe was unable to open a connection to the specified FTP host and deploy your project.

Check your network connections, and check to be sure you entered the correct syntax for your FTP host.   This error could also be caused if your FTP host is not available at this time.

No differences were found for project <project>.

This error has the following cause and solution:

* You compared two items with no differences.

This is an informational message to tell you that no differences were found between the compared items.

WARNING: If you do not use a password, others will be able to log into this SourceSafe account. Proceed with no password?

This error has the following cause and solution:

- You attempted to log on without specifying a password.

This is an informational warning to make sure you want to proceed without setting a password on your account.   Choose Yes to log on, or No to cancel.

Unable to launch the web browser for page <file>.

This error has the following cause and solution:

- Visual SourceSafe was unable to start your web browser.

Check that your web browser is functioning normally.   You may need to re-install the browser to verify your registry settings.

Remove <database> from your Database list?

This error has the following cause and solution:

- You are deleting a Visual SourceSafe database entry from your Open SourceSafe Database list.

Choose Yes to delete the specified entry, or No to leave the database entry on the list.

\<database> is already in your database list. Would you like to rename it?

This error has the following cause and solution:

- You are renaming a Visual SourceSafe database entry in your Open SourceSafe Database list.

Choose Yes to rename the specified entry, or No to leave the database entry as it is on the list.

The Database <database> does not exist, would you like to search for it?

This error has the following cause and solution:

- You are trying to find a database that does not exist.

Choose Yes to browse for the correct Visual SourceSafe database, or No to cancel the operation.

You must enter a filename with no path for the Site map.

This error has the following cause and solution:

- You tried to specify a file path for the site map file.

The site map file specification does not accept a path.   Enter just the filename, such as SITEMAP.HTM.

The Database <database> does not exist. Please select another.

This error has the following cause and solution:

- You are trying to find a database that does not exist.

Choose the Browse button to look for the correct Visual SourceSafe database, or No to cancel the operation.

SourceSafe cannot find a default database. Would you like to select one?

This error has the following cause and solution:

- Visual SourceSafe is trying to find a default database that does not exist.

Choose Yes to browse for the correct Visual SourceSafe database, or No to cancel the operation.

Could not login to <ftp site>.

This error has the following cause and solution:

- You cannot log in to the specified ftp site.

You may be using an invalid address, invalid username, invalid password, or perhaps the site is just not responding.   Make sure you have the right ftp address. You can enter the ftp address using a web browser and see if the address is valid.

Could not Deploy <project> to <host>.

This error has the following cause and solution:

- Visual SourceSafe was unable to open a connection to the specified FTP host and deploy your project.

Check your network connections, and check to be sure you entered the correct syntax for your FTP host.   This error could also be caused if your FTP host is not available at this time.

This message can also mean that you are out of disk space.   It could also mean that you don't have permission to write to the location although you may have read permission.

The URL for project %s was not set properly.

This error has the following cause and solution:

- You set a bad URL path for a web project, or the URL uses a syntax that Visual SourceSafe does not understand (for instance, it might use www.microsoft.com/... instead of http://www.microsoft.com/...)

Use the Administrator program to set a proper URL.

An error occurred while checking hyperlinks for <file>.

This error has the following cause and solution:

- The working folder path, Visual SourceSafe specification, or URL was too long..

Check that you aren't exceeding path limits for folder, specification, or URL.   Check that htmlfilt.dll is still present on your system and registered properly (re-running the client installation should fix this problem).

A link in \<file\> was ignored because it was too long.

This error has the following cause and solution:

* The URL is too long for the internal buffer.

Use a shorter URL.

No HTML files were available in this project to check hyperlinks.

This error has the following cause and solution:

- This informational message tells you that no files were checked for web hyperlinks because no HTML files were found.

If you expected HTML files in this project, check that they were added properly and that you entered the correct project name..

The base URL for this web project, '<project>', has bad URL syntax.

This error has the following cause and solution:

- The URL that you entered for the base URL for a Visual SourceSafe web project has bad URL syntax.

Valid syntax for the URL is shown in these examples:

```
http://www.microsoft.com/default.htm
http://www.microsoft.com
```

You can change the URL in the Web Projects tab of the Visual SourceSafe Administrator Tools options dialog box, or edit the URL in the SRCSAFE.INI file.

Unable to load the HTML parser.

This error has the following cause and solution:

- You have a corrupted Visual SourceSafe installation. Either the DLL is gone or the registry isn't set to the right DLL.

Re-install Visual SourceSafe.

This command requires internet components which are not installed on your machine.

This error has the following cause and solution:

- You tried to use the   Check Web Links command on a RISC platform, but there wasn't a WININET.DLL to distribute when we shipped Visual SourceSafe.   You can get this required DLL when a RISC version of the browser exists, or
- You are missing the correct version of WININET.DLL on your Intel or Alpha computer.   Check to see if the file has been moved or renamed.

To make sure you have the latest internet components, go to http://www.microsoft.com/ie/ and download Microsoft Internet Explorer 3.0 or higher.   Then restart Visual SourceSafe to enable this command.

Could not display this URL.

This error has the following cause and solution:

- You tried to check an external web link, but there was no web browser we could find in the register to handle a http:: path; or, the link syntax was incorrect for ShellExecute (this could happen if you tried to check a virtual root by double clicking on it instead of setting the virtual root).

Install Internet Explorer 3.0 (or later).

Warning: the Rollback command will branch this file, breaking its link to other projects that share it.

This warning has the following cause and solution:

- This warning occurs when you roll back a file that has been branched or is pinned in another project.

Choose Yes to continue, or No to cancel.

Get on Checked Out File Dialog

This message has the following cause and solution:

- You chose to get a file when it was already checked out.

When you get a file, a new copy is written to your working folder.   If you already have the file checked out, you may have made changes that will be overwritten.   In this situation, Visual SourceSafe gives you some options:

- Choose replace to overwrite the working folder file.
- Choose Merge to merge changes from the working folder file into the Visual SourceSafe database file.
- Choose Leave to leave the working folder file alone (the Get command will be ignored for that file).
- Choose Cancel to cancel the get operation.
- Choose Apply to all items to have your choices in this dialog apply to all other files you currently have checked out.

The Admin user currently has no SourceSafe password.

This message has the following cause and solution:

- You have not assigned a password for the administrator user of the Administrator program.

Assign a password to the Admin user by choosing the Users Change Password command.   If you don't do this, any user can log in as Administrator and make changes to your Visual SourceSafe database.

You are about to remove the Admin user's SourceSafe password, which will allow any user to run the Administrator's program.

This message has the following cause and solution:

- You are deleting the Admin user's password.

Leaving the Admin user with no password is a security risk, because then any user can log into the Administrator program and make changes to your Visual SourceSafe database. Are you sure you want to remove the Admin user's SourceSafe password?   Choose Yes to continue, or No to cancel the operation.

The URL <URL> does not have proper HTTP URL syntax.

This message has the following cause and solution:

- The URL that you entered has bad URL syntax.

Valid syntax for the URL is shown in these examples:

```
http://www.microsoft.com/default.htm
http://www.microsoft.com
```

The deploy path <path> is not a proper file system or FTP deploy path.

This message has the following cause and solution:

- One of the paths entered for deploying the project is either not a proper operating system file system path or is not a proper FTP URL.

ftp paths can look like one of the following.

```
ftp://user:user@www.microsoft.com/ssafe
ftp://user@www.microsoft.com/ssafe
ftp://www.microsoft.com/ssafe
```

They always start with "ftp://".   They always end with the ftp address (server name and, optionally, directory path).   In between, you may have nothing (as in the bottom example above).   Or, you may have a username (as in the middle), or a username and password (as on the top).

Please enter a number between 0 and 32767.

This message has the following cause and solution:

- You have not entered a number of context lines between 0 and 32767.

Context lines are the lines immediately preceding and following a difference.   Showing these lines in a difference display helps you understand the context of the change or difference.   If you have checked the Show Context box in the Advanced options of the Show Difference command, you need to specify the number of preceding and following lines to show.

Journal_File Initialization Variable

<u>See Also</u>

Sets a filename where all Visual SourceSafe commands that change stored data are logged.   Used in the SRCSAFE.INI file only.

Syntax

Journal_File = *filename*

Example

Sets SSLOG.TXT as a journal of all Visual SourceSafe commands:

```
Journal_File = O:\DOCS\SSLOG.TXT
```

Remarks

This variable is generally set in SRCSAFE.INI, rather than in SS.INI, so that it applies to all users.   If this variable is set, the file becomes a journal of Visual SourceSafe actions.   The journal is a text file that is updated automatically by Visual SourceSafe to show the various actions that users take.   Only actions that change data stored in the database are logged.

Set this variable in the SRCSAFE.INI file by using the Log All Actions In Journal File option on the General tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

General Options Tab

Keyword_Masks Initialization Variable

See Also

Indicates which files to use, or not to use, **keyword expansion** on.   Used in the SRCSAFE.INI file.

Syntax

Keyword_Masks = *masks*

Example

Indicates that all .C and .H files except for SS.C should be keyword expanded:

```
Keyword_Masks = *.C, *.H, !SS.C
```

Remarks

By default, Visual SourceSafe does not scan files for keyword expansion when it performs a Check In or Add command.   Set this variable to indicate which files Visual SourceSafe should keyword expand, noting that the Add and Check In commands may be slower for these files.

You can also set this variable in the SRCSAFE.INI file by using the Expand Keywords In Files Of Type option on the General tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

General Options Tab

Keyword Expansion

Shadow Initialization Variable

See Also

Sets a **shadow folder** for a **project**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow = *shadow folder*

Example

Sets M:\MASTERS\WORD as a shadow folder for the project `$/WORD`:

```
[$/WORD]
Shadow = M:\MASTERS\WORD
```

Remarks

This variable references a shadow folder you've created for your project.   This folder shadows the project, reflecting every file change in its own contents, and remains up-to-date.   Before setting this variable, you must create the shadow folder somewhere and populate it with the files you want maintained there (using the Get Latest Versions command to that folder is the easiest way). From that point on, the files you've put in this shadow folder are kept up-to-date.   Also, any new files added or deleted from the project in the **Visual SourceSafe database** are updated in the shadow folder.

The shadow folder should not be the folder you work from and modify code in.   It is a separate folder that is maintained by Visual SourceSafe.   You can, however, go there and build your project or look around the code, as the folder reflects the exact current state of the project.

You can also set this variable in the SRCSAFE.INI file by using the Set Shadow Folder For Project option on the Shadow Folders tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.   If you use the Shadow Folders tab, you do not need to use the Get latest Version command to initialize the folder.

See Also

Shadow Folders Options Tab

Using Shadow Folders

Shadow_DOSName Initialization Variable

Controls whether Visual SourceSafe places long filenames in the **shadow folder**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow_DOSName = {Yes|No}

Example

Allows long filenames in the shadow folder:

```
Shadow_DOSName = No
```

Remarks

Versions of Visual SourceSafe that support long filenames (Windows NT and Macintosh) can place these long filenames into a shadow folder.   In a cross-platform environment, however, this may not be wanted, as it may create filenames that are unreadable to MS-DOS.   This variable can therefore be used to force all versions of Visual SourceSafe to truncate names to MS-DOS standard names in the shadow folder.

See Also

<u>Using Shadow Folders</u>

Shadow_EOL Initialization Variable

See Also

Sets the end-of-line character used in the **shadow folder**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow_EOL = {n|r|rn}

Example

Terminates all lines in the shadow folder with carriage-return/line-feed pairs:

```
Shadow_EOL = rn
```

Remarks

When Visual SourceSafe copies a file into the shadow folder, it uses the end-of-line character used by its own operating system.   For instance, Visual SourceSafe for MS-DOS uses a carriage-return/line-feed pair.   In a multiplatform environment, the files in the shadow folder are therefore a mixture of end-of-line characters, based on which operating system each file was updated from most recently.

This variable overrides that mechanism, forcing all versions of Visual SourceSafe to use the specified end-of-line character.   Set the variable to **n** for line feeds (the UNIX default), **r** for carriage returns, or **rn** for carriage-return/line-feed pairs (the PC default).

You can also set this variable in the SRCSAFE.INI file by using the End of Line Characters For Files option on the Shadow Folders tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

Shadow Folders Tab

Using Shadow Folders

Shadow_Extension Initialization Variable

<u>See Also</u>

Changes the extension of files placed in the Visual SourceSafe **<u>shadow folder</u>**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow_Extension = *extension*

Example

Sets the shadow extension to ??V:

```
Shadow_Extension = ??V
```

Remarks

By default, all files are given the same name in the shadow folder that they have in Visual SourceSafe.   In some cases, however−especially for the sake of Make utilities

−you may want files in the shadow folder to have extensions different from their real counterparts'.   You can therefore use this variable to set the extension that is used for files in the shadow folder.
For example, if you set this variable to .SDW, all files in the shadow folder are given the extension .SDW.
This does not enable you to distinguish different file types, however; the question mark is therefore provided as a special character, which tells Visual SourceSafe to use the actual character.   Hence, in the above example (??V), the extension .PAS turns into .PAV; the extension .C turns into .C_V.

See Also

<u>Using Shadow Folders</u>

Shadow_ReadOnly Initialization Variable

See Also

Controls whether files in the **shadow folder** are **read-only**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow_ReadOnly = {Yes|No}

Example

Specifies that all Visual SourceSafe files in the shadow folder are read-only (the default):

```
Shadow_Readonly = Yes
```

Remarks

This variable turns on the read-only flag for all files that Visual SourceSafe places in the shadow folder.

You can also set this variable in the SRCSAFE.INI file by using the Set Read Only Flag For All Files option on the Shadow Folders tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

Shadow Folders Options Tab

Using Shadow Folders

Multiple_Checkouts Initialization Variable

See Also

Controls whether many people can **check out** one file simultaneously.   Used in the SRCSAFE.INI file only.

Syntax

Multiple_Checkouts = {Yes|No}

Example

Indicates that only one person can have a file checked out at one time (the default):

```
Multiple_Checkouts = No
```

Remarks

Under ordinary circumstances, a check out in Visual SourceSafe is exclusive−that is, only one person can have a file checked out at one time.   However, if you set this variable to Yes in your SRCSAFE.INI file, Visual SourceSafe allows many users to check out the same file.   When the first user checks in the file, Visual SourceSafe updates the file; thereafter, Visual SourceSafe **merges** all subsequent changes.

You can also set this variable in the SRCSAFE.INI file by using the Allow Multiple Checkouts option on the General tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.
   Note   Multiple checkouts are not allowed on binary files.

See Also

General Options Tab

Shadow_SetTime Initialization Variable

See Also

Controls the date/time given to files in the **shadow folder**.   Used in the SRCSAFE.INI file only.

Syntax

Shadow_SetTime = {Current|Mod|Update}

Example

Stamps shadow files with the date and time they are placed into the shadow folder (the default):

```
Shadow_SetTime = Current
```

Remarks

This variable affects the date/time stamp given to files in the shadow folder, in the same way SetTime affects files in your **working folder**.

You can also set this variable in the SRCSAFE.INI file by using the File Date/Time option on the Shadow Folders tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

Shadow Folders Tab

Using Shadow Folders

Data_Path Initialization Variable

Sets the folder path to the data folder used by Visual SourceSafe to keep **version** information.   Used in SRCSAFE.INI only.

Syntax

Data_Path = *folder path*

Example

Sets the data folder to the DATA folder in the VSS folder on the \\MIS\Developers share:

```
Data_Path = \\MIS\Developers\SS4\DATA
```

Remarks

The Visual SourceSafe administrator should use the default or set this variable only once (before creating a new database).

By default, Visual SourceSafe creates a relative data path variable as part of its setup.   This variable is set to Data_Path = DATA.   This default causes Visual SourceSafe to look for the DATA subfolder on the path where the SRCSAFE.INI file is located.

Use_Network_Name Initialization Variable

See Also

Sets whether Visual SourceSafe reads users' logon names from the operating system.   Used in the SRCSAFE.INI file only.

Syntax

Use_Network_Name = {Yes|No}

Example

Uses network logon names as Visual SourceSafe logon names:

```
Use_Network_Name = Yes
```

Remarks

Set this to No only if you want all the Visual SourceSafe users to use logon names different from their network logon names.   If you set it to No, Visual SourceSafe displays a logon dialog box unless you are supplying a **username** in some other way (such as through an environment variable).

When set to Yes, Visual SourceSafe uses the network logon name for Windows 95 and Windows NT as the Visual SourceSafe username.   If the network name matches a Visual SourceSafe username, the logon to Visual SourceSafe is automatic.

You can also set this variable in the SRCSAFE.INI file by using the Use Network Name For Automatic User Log In option on the General tab in the SourceSafe Options dialog box in Visual SourceSafe Administrator.

See Also

General Options Tab

Users_Txt Initialization Variable

Sets the location of the USERS.TXT file used to store the location of users' SS.INI files. Available in the SRCSAFE.INI file only.

Syntax

Users_Txt = *folder path*

Example

Tells Visual SourceSafe to use the USERS.TXT file located in the SRCSAFE.INI folder:

```
Users_Txt = USERS.TXT
```

Remarks

Change this only if you have to move the USERS.TXT file.   You can specify a different filename or a complete path to the users text file; for example, \\SRCSRVR\SS5\USERS.TXT.

Database_Name Initialization Variable

Sets the name of the current **Visual SourceSafe database** as displayed on the **Visual SourceSafe Explorer** title bar.   Available in the SRCSAFE.INI file only.

Syntax

Database_Name = *character string*

Example

Sets the database name to "Information Services":

```
Database_Name = Information Services
```

Remarks

The database name should be something that clearly indicates what a set of projects is for.   Product names, project code names, and so forth, are appropriate.   If undefined, the Visual SourceSafe Explorer title bar contains the last point of the path to an installation.   For example, the title bar for an installation at \\server\share\SS5.0 will contain SS5.0.   Note that if the folder name is VSS, the title bar uses the share name instead.

Delete_Local Initialization Variable

<u>See Also</u>

Indicates during Add, Check In, and Undo Check Out commands whether or not you are maintaining a **<u>local copy</u>** of a complete **<u>project</u>**.

Syntax

Delete_Local = {Yes|No}

Example

Tells Visual SourceSafe not to delete local copies when you are done with them:

```
Delete_Local = No
```

Remarks

By default, when you are working on a project, Visual SourceSafe keeps a copy of the entire project in your **<u>working folder</u>**.   When you set Delete_Local to Yes, Visual SourceSafe only keeps copies of the files that you are currently working on.   Therefore, when you add, check in, or undo the check out of a file, instead of flagging it read-only, Visual SourceSafe deletes your local copy of the file.

When this variable is set, you cannot compile out of your working folder, because crucial files may not be in that folder.   For this reason, when you set Delete_Local to Yes, you also frequently set a **<u>shadow folder</u>** for that project.

Even if Delete_Local is set, you can still use the Get Latest Versions command to retrieve a file into your working folder.   In that case, the file remains in your working folder until you delete it, or check out and then check in.

See Also

<u>Using Shadow Folders</u>

Update_No_Change Initialization Variable

See Also

Controls what happens when you **check in** a file that you have not changed.

Syntax

Update_No_Change = {Update|Uncheckout|Ask}

Example

Undoes check out of files that have not changed:

```
Update_No_Change = Uncheckout
```

Remarks

When you check in a file that has not changed since you checked it out, Visual SourceSafe has three options: It updates the file, undoes the check out of the file or asks you how to handle each file. Because no changes have been made, the only difference between the first two cases is that check in creates an entry in the file **history** with a comment but no change record; undo check out does not.

- Update−Updates files even if they have not changed.
- Uncheckout−Undoes the check out of unchanged files.
- Ask−Asks whether Visual SourceSafe should update an unchanged file.

You can also set this variable in your SS.INI file by using the Check In Unchanged Files option on the General tab in the SourceSafe Options dialog box.

See Also

General Options Tab

Temp_Path Initialization Variable

Sets the folder where Visual SourceSafe stores temporary files.

Syntax

Temp_Path = *path to Visual SourceSafe temporary folder*

Example

Indicates that the temp path is a folder called `temp` immediately under the folder that contains SRCSAFE.INI:

```
Temp_Path = Temp
```

Remarks

You can also set this variable in your SS.INI file by using the Folder for Temporary Files option on the General tab in the SourceSafe Options dialog box.

See Also

General Options Tab

Lock_Mode Initialization Variable

See Also

Sets the type of database **locking** used by SourceSafe.   Available in the SRCSAFE.INI file only.

Syntax

Lock_Mode = {Native|Lockfile}

Example

Indicates that native locking is used by Visual SourceSafe to avoid database record collisions:

```
Lock_Mode = Native
```

Remarks

Native locking uses the operating system's inherent file and record locking capabilities (via the **fcntl** system call).   This system is very fast, but may not be supported if you are running Visual SourceSafe across multiple UNIX machines.

Lockfiles are a Visual SourceSafe-specific mechanism, involving the creation of special files in the Visual SourceSafe DATA\LOCKS folder, to achieve the same database record locking as the Native setting.   This method is slower than native locking, but works in networked and cross-platform situations.

It is important to choose the right locking mechanism for your system.   If you are running Visual SourceSafe exclusively on Microsoft Windows NT machines, use native locking (the default).   If you are running Visual SourceSafe in an environment where byte locks to the database are not supported, use Lockfile.   This prevents collisions that might cause unpredictable results in the Visual SourceSafe database.   If you are not sure if your network supports native locking, use the TESTLOCK.EXE program.

Note   If Visual SourceSafe does not exit normally, in some circumstances the UM.LCK file may be left in the DATA\LOCKS directory.   Delete this file manually to avoid TESTLOCK errors or weird behavior with the Lock_Mode variable.

See Also

TESTLOCK.EXE Utility

Deploy_Path Initialization Variable

<u>See Also</u>

Specifies the location files are copied to when a user chooses the Deploy command on Visual SourceSafe's Web menu.

Syntax

Deploy_Path = *folder*

Example

Instructs SourceSafe to deploy files to the \\internet\server1\ssafe location when the Deploy command is selected:

```
Deploy_Path = \\internet\server1\ssafe
```

Remarks

Used only for projects marked as web site projects by the URL initialization variable. (You can also mark a project as a web site project using the Administrator's Tools Options Web Sites tab.) This variable can also be set on the Administrator Tools Options Web Sites tab.   You can set multiple deployment paths, separated by a comma.

See Also

Web Site Tab

General Project Properties Tab

Deploy Command Line Command

Deploy Command

URL Initialization Variable

Marks a project as a web site project.   When a web site project is selected, Visual SourceSafe Explorer's Web menu is activated.

Syntax

URL = *url*

Example

Specifies that the project under which this variable is listed corresponds to a web site Uniform Resource Locator (URL):

```
URL = http://www.microsoft.com/ssafe
```

Remarks

This variable can also be set on the Administrator Tools Options Web Sites tab.

See Also

Web Site Tab

Visual SourceSafe Web Features

External_Link_File Initialization Variable

Specifies the location of the file used to list external web links deemed valid by a Visual SourceSafe user.

Syntax

External_Link_File = *file path*

Example

Specifies that the LINKS.TXT and GOODLINKS.TXT files hold the list of validated web links associated with a Visual SourceSafe web site project:

```
External_link_File = \\VSSserver\links.txt, c:\webproj\goodlinks.txt
```

Remarks

The links file is a text file list of URLs which are considered valid, even though they have not been validated by Visual SourceSafe.   You may specify a comma-delimited list of files to keep several copies of the link list.   This variable can also be set on the Administrator Tools Options Web Sites tab.

See Also

Web Site Tab

Visual SourceSafe Web Features

Check Web Links Command

SiteMap Initialization Variable

Specifies the location of the HTML file generated when a user selects the Create Site Map command from Visual SourceSafe's Web menu..

Syntax

SiteMap = *file path*

Example

Specifies that the SITEMAP.HTML file is the auto-generated site map for the project under which this variable appears:

```
SiteMap = c:\webproj\sitemap.html
```

Remarks

This variable can also be set on the Administrator Tools Options Web Sites tab.

See Also

Web Site Tab

Visual SourceSafe Web Features

Create Site Map Command

Deploy_Proxy Initialization Variable

Specifies the name of your proxy server.

Syntax

```
Deploy_Proxy = myproxy
```

Example

Specifies that the corporate intranet requires Visual SourceSafe to use the myproxy proxy server to access external ftp sites:

```
Deploy_Proxy = myproxy
```

Remarks

You can set only one proxy server here.

Deploy_Host_Local Initialization Variable

Lists the local hosts that do not require a proxy server for ftp deployment.

Syntax

```
Deploy_Host_Local = ftp.microsoft.com
```

Example

Specifies that when the web site project is deployed, no porxy server is required for the ftp.microsoft.com site:

```
Deploy_Host_Local = ftp.microsoft.com
```

Remarks

If a Deploy_Proxy initialization variable is specified, it is used for all ftp deployment, except for the server(s) specified in Deploy_Host_Local

Web_Default_Filename Initialization Variable

Specifies the default filename for your web project.   The default file is the one used by web browsers if someone specified a URL to your site without a filename specification.   Common default filenames are DEFAULT.HTM and INDEX.HTM.

Syntax

```
Web_Default_Filename = c:\website\default.htm
```

Example

Specifies that the INDEX.HTML file is the default file when a user enters the URL to your web site without a filename specification:

```
Web_Default_Filename = c:\website\index.html
```

Virtual_Root Initialization Variable

Sets the virtual root for a particular web project.

Syntax

```
Virtual_Root = /SSAFE
```

Example

Specifies that when web hyperlinks are checked, links including a virtual root designation (supported by Microsoft's Internet Information Server) of /VBASIC are checked:

```
Virtual_Root = /VBASIC
```

Remarks

If your web project doesn't include hyperlinks to URL's with a virtual root, you do not need to set this variable.

Check_Hyperlink_Masks Initialization Variable

Controls the file types hyperlinks are checked in.

Syntax

Check_Hyperlink_Masks = *.HTM, *.HTML, *.ASP

Example

Specifies that when web hyperlinks are checked, links are checked in files with the *.HTM, *.HTML, and *.ASP file extension:

Check_Hyperlink_Masks = *.HTM, *.HTML, *.ASP

Glossary

A

access rights

Advanced dialog box

automatic merge

B

branched file

branching

C

checked-in file

working folder

user list

The list of users who can use the Visual SourceSafe database.   This list is maintained by the Visual SourceSafe administrator and displayed in Visual SourceSafe Administrator's main window.

access rights

The levels of permission to use the Visual SourceSafe database that users are granted by the Visual SourceSafe administrator when project security is enabled.   The levels of access rights are Read**,** Check Out, Add, and Destroy.   For more information, see the Administration chapter of the *Microsoft Visual SourceSafe User's Guide*.

Advanced dialog box

A dialog box that contains optional or additional settings.   Many SourceSafe dialog boxes have an Advanced button that displays these additional settings.

automatic merge

When multiple users have the same file checked out, their changes to the file are merged by Visual SourceSafe during check in.

branched file

A file whose share link has been broken using the Branch command.

branching

The process of sharing a file with another project and then separating it into two (or more) branches. Branching creates two paths of the file's history, both with a common version as the origin, but different subsequent versions.

checked-in file

A file stored in the Visual SourceSafe database and unavailable for modification.

checked-out file

A file that has been reserved for work by a user.   Users check out files so changes can be made to them.   In the default configuration, Visual SourceSafe allows only one user at a time to check out a file.   Checking out a file copies its latest version into the user's working folder.

check out folder

The folder to which a file is checked out in Visual SourceSafe.   It is important to distinguish this from the working folder.   If you check out a file, the file is checked out to your working folder.   From the perspective of another user, the file is in the check out folder.   The check out folder is displayed in the Check Out Folder column of the file pane of Visual SourceSafe Explorer; the working folder is displayed under the toolbar.

Clipboard

A temporary storage location used by Microsoft Windows to transfer text, graphics, and code.

cloaking

Hiding a project from being affected by certain commands, namely Get, Check Out, Check In, Undo Check Out, and Project Show Difference.

column format

A formatting option used with keyword expansion within a file.   This option (using two colons instead of one in the keyword instructions) tells Visual SourceSafe to align columns of keyword information for a more attractive and readable display.

conflict

Two or more differing changes to the same line of code in a multiple check out situation. Visual SourceSafe recognizes conflicts during a merge operation, either explicitly when a user clicks the Merge Branches command, or implicitly when checking in a file that multiple users have checked out.

conflict marker

A symbol used to designate conflicting changes to a file.   These symbols are:

- <<<<<<   SourceSafe version
- ======   Conflict separator
- >>>>>>   Local version

Visual SourceSafe places these markers in the file after a conflicting check-in or merge operation, so that you can find and resolve the conflict more easily.

cross-platform development

Visual SourceSafe supports transparent file-compatibility across multiple processors and operating systems.

current project

The project selected in the project pane of the Visual SourceSafe Explorer window.

current version

The version of a file most recently stored in the Visual SourceSafe database.   The current version has the highest version number of a file in Visual SourceSafe.

delta

Changes. In Visual SourceSafe, a delta is the difference between version X of a file and version X−1 of the same file. Visual SourceSafe uses reverse delta technology to store changes.

Delete command

Removes files and projects from a Visual SourceSafe project, and marks them as deleted; the items still exist, however, and can be recovered using the Recover command.

Destroy command

Permanently removes deleted files and projects from the Visual SourceSafe database.   Once destroyed, the items cannot be recovered.

development environment

A set of software development tools, presented as a unified environment in which the software developer can efficiently work.   Microsoft Visual Basic and Microsoft Visual C++ are examples of such environments, as they combine a coding language with tools and controls.   Visual SourceSafe can be integrated into these environments, providing access to its source code control features.

drag and drop

A combination of features that allows the user to drag an item and drop it onto another item using the mouse.   An item can be a source (the item the user drags) or a target (the item on which the user drops a source).

To drag an object, click the object, then while holding down the mouse button, move it to the new location.   Release the mouse button to drop the object.

file header

Text at the beginning of a file that contains information, such as the revision number of the file, the author, the date and time of last modification, and so forth.   So it doesn't interfere with compiling, this information usually appears as comments in programming language files.

file pane

The right side of the Visual SourceSafe Explorer window.   This pane contains the file list, a list of all the files in the current project.

file list

The list of files in the current project, found in the file pane of the Visual SourceSafe Explorer window.

history

A record of changes to a file since it was initially added to Visual SourceSafe, which Visual SourceSafe keeps by using reverse delta technology.   With the file history, you can return to any point in the file's history and recover the file as it existed at that point.   The History of Project dialog box shows the record of significant events in the current project, such as labeling, and deletion or addition of files and subprojects.

inheritance

The inherited effect of variables in Visual SourceSafe initialization files that subprojects receive from their parent projects.   Variables in these files can be grouped under headings to specify behavior.   If a variable is set before any group heading in the initialization file, the variable affects all projects; the effect is inherited by the subproject variables.

keyword

A word with a special meaning to Visual SourceSafe.   You can use keywords in your text files to automatically create a file header with version-specific information.   Use Visual SourceSafe keywords to place information from Visual SourceSafe directly into a text file.

keyword enabled file

A file that contains Visual SourceSafe keyword information.

keyword expansion

The process Visual SourceSafe uses to replace a Visual SourceSafe keyword with meaningful header information when you use the Check In and Add Files commands.   For example, the string `"$Revision:    $"` tells Visual SourceSafe to expand the **Revision** keyword automatically with the current file version number.

label

A user-defined name you can attach to a specific version number of a file or project.

local copy

A copy of a file stored in your working folder on your local computer.   The local copy may differ from the Visual SourceSafe master copy if the local copy has been changed since the last check out, or if the master copy was changed by another user while you were working on the local copy.

locking

A system of ensuring that two processes do not try to affect the same record in a database at the same time.   To coordinate record access, Visual SourceSafe applies native locking, which uses native operating system functions.   Visual SourceSafe also can be set to use lockfiles, which create temporary files in the LOCKS folder.

log on

The process of entering and verifying a user's name and password to access the Visual SourceSafe database.

master copy

The most recently checked-in version of a file stored in the Visual SourceSafe database, as opposed to the local copy of a file in your working folder.

merging

The process of combining differences in two or more changed copies of a file into a new version of the file.   A merge involves at least two different files (which may be different versions of the same file or changes made to the same version of the file) and creates a new file made up of the results of the merge.

multiple check out

Simultaneous check outs by two or more users.   Multiple check out is not possible unless it is enabled by the Visual SourceSafe administrator.

parent project

A project that contains one or more subprojects.   A project can be both a parent project and a subproject at once, if it is in the middle of the project hierarchy.

password

A text string used as security to verify the identity of a user. A user password is often required to use the Visual SourceSafe database.

project

A group of related files, typically all the files required to develop a software component.   Files can be grouped within a project to create subprojects.   Projects can be defined in any way meaningful to the user(s)−as one project per version, or one project per language, for example.   In general use, projects tend to be organized in the same way file directories are.

project pane

The left side of the Visual SourceSafe Explorer window.   The project pane contains the project list, a list of all available projects in the Visual SourceSafe database.

project list

A list, in the left pane of the Visual SourceSafe Explorer window, of all the projects available in the Visual SourceSafe database.

Purge command

Permanently removes previously deleted files and projects from the Visual SourceSafe database. Once purged, the items cannot be recovered.

read-only file

A file marked as read-only in its file attributes.   Such a file can be viewed in an appropriate text editor, but cannot be modified.   Visual SourceSafe marks the file as read-only when you use the Check In and Get Latest Version commands.

recursive operation

An operation that is applied both to a parent project and to all the subprojects of that parent. For example, you can use the Check Out command recursively to check out all the files in the project list simultaneously, and avoid selecting each file individually.

Results pane

A portion of the Visual SourceSafe Explorer window where results from Visual SourceSafe operations are shown. For example, when you check in a file, this pane shows the file name being checked in.

reverse delta

The change-storage technology Visual SourceSafe uses, in which incremental changes to a baseline file are stored, rather than each successive version of the file in its entirety.   In Visual SourceSafe, the current version of a file is used as the baseline, and the changes from the previous versions are saved.   This technology results in reduced disk storage requirements and faster access times, because only the current version is always stored in the database in its entirety.

rights propagation

The default assignment of user-access rights in subprojects based on rights assigned in the parent project.   This default assignment can be changed.

root project

The highest-level project, with the name $/ in the project list.   All projects in a Visual SourceSafe database are subprojects of the root project.

security

Visual SourceSafe has two levels of security: default security and project security.   Default security provides two types of access rights: Read-Write and Read-Only.   When project security is enabled, four access rights are available per user, per project: Read, Check Out, Add, and Destroy.   Each succeeding right includes all rights preceding it.   The Destroy access right provides unlimited access and is equivalent to Read-Write rights under default security.

shadow folder

A central, optional folder that contains current versions of all the files in a project.   The shadow folder does not contain the master copy of a file or the local copy of a file.   Instead, it provides a central location from which to view the overall structure of the project and serves as a convenient place to build or compile the project.

shadow variable

An initialization file variable used to specify the location of a shadow folder.

shared file

A file simultaneously used by, and part of, more than one project.

share link

The link between a file and the one or more projects it may be shared with.   This link serves to update the shared file with any checked-in changes, regardless of which project the file was checked out from.

source code control

The management of a file's change history and the file's relation to a larger grouping of related files known as a project.   Source code control is a vital part of the efficient development of software applications.   Visual SourceSafe is a project-oriented type of source code control.

subproject

A project within a parent project.

Visual SourceSafe administrator

The person responsible for the Visual SourceSafe database.   The administrator uses Visual SourceSafe Administrator to control the location of the database, the user list, and access rights of each user, and performs setup and backup duties on the database.   The administrator's user name is always Admin.

Visual SourceSafe database

The central database where all master copies, history, project structures, and user information is stored.   A project is always contained within one database; multiple projects can be stored in one database, and multiple Visual SourceSafe databases can exist to store multiple projects.

Visual SourceSafe Explorer

The main user interface of Visual SourceSafe, by default comprising two panes－the left project pane and the right file pane, as well as the toolbar, status bar, and menus, etc.   Visual SourceSafe Explorer is displayed when you click the Visual SourceSafe icon.

status bar

A bar at the bottom of the Visual SourceSafe Explorer window used for displaying project and file status information and current command information.

username

A unique identifying string for a given user.   Used for logging on.

version control

The control of users' changes to a file, as well as the maintenance of version history on the file.

version number

A number that indicates the number of revisions a file has undergone since it was added to Visual SourceSafe.   This number is displayed in the History dialog box to help identify specific versions of a file or project.   Version numbers are always whole numbers.

version tracking

The record keeping process of tracking a file's history from the initial version to the current version. Changes to a file are tracked as part of this process.

wildcard characters

The asterisk (*) and question mark (?) are wildcard characters.   You can use these characters   to match patterns.   You can also use wildcard characters and matching characters to further refine a search.

Symbol	Example	Usage
*	wh* finds what, white, and why; *at finds cat, bat, and what	Like the MS-DOS asterisk (*) wildcard character, this asterisk matches any number of characters.
?	b?ll finds ball, bell, and bill	Like the MS-DOS (?) wildcard character, this symbol matches any single character.

A backslash preceding an asterisk or question mark indicates a literal asterisk or question mark: * or \?   (This is necessary if you want to search for actual asterisks, question marks, or backslashes.)   A double backslash (\\) indicates a backslash.   Any other use of a backslash is ignored.

working folder

A specified folder on a user's local computer used to store files when they are checked out of the Visual SourceSafe database.   A user makes changes to files in the working folder, and then checks the modified files back into the Visual SourceSafe database for version tracking.

web site project

A project marked as a web site project in the Visual SourceSafe Administrator.   Such a designation allows special web site commands, such as Deploy, to be used in this project.

visual merge

A merge operation where <u>conflicts</u> are resolved visually, in an easy-to-use graphical interface.

Accessibility for People with Disabilities

Microsoft is committed to making its products and services easier for everyone to use.   This help file provides information about the following features, products and services, which make Microsoft Windows, Microsoft Windows NT and Microsoft Visual SourceSafe more accessible for people with disabilities:

Microsoft services for people who are deaf or hard-of-hearing.

Access Packs for either Microsoft Windows or Microsoft Windows NT, a software utility that makes using Windows and Windows NT easier for people with motion or hearing disabilities.

Keyboard layouts for single-handed users.

Microsoft software documentation on audio cassette, floppy disk and compact disc (CD).

Third-party utilities to enhance accessibility.

Hints for customizing Microsoft Windows or Microsoft Windows NT.

More information about products and services for people with disabilities.

Note   The information in this section applies only to users who purchased Windows or Windows NT in the United States.   If you purchased Windows or Windows NT outside the United States, your Windows package contains a subsidiary information card listing Microsoft support services telephone numbers and addresses.   You can contact your subsidiary to find out whether the type of products and services described in this help file are available in your area.

Microsoft Services for People Who Are Deaf or Hard-of-Hearing

Through a text telephone (TT/TDD) service, Microsoft provides people who are deaf or hard-of-hearing with complete access to Microsoft product and customer services.

You can contact Microsoft Sales Information Center on a text telephone by dialing (800) 892-5234 between 6:30 A.M. and 5:30 P.M. Pacific time. For technical assistance you can contact Microsoft Support Network on a text telephone at (206) 635-4948 between 6:00 A.M. and 6:00 P.M. Pacific time, Monday through Friday, excluding holidays. In Canada, dial (905) 568-9641 between 8:00 A.M. and 8:00 P.M. Eastern time, Monday through Friday, excluding holidays. Microsoft support services are subject to Microsoft prices, terms, and conditions in place at the time the service is used.

Access Packs for Microsoft Windows and Microsoft Windows NT

Microsoft distributes Access Packs for Microsoft Windows and Microsoft Windows NT, which provide people with motion or hearing disabilities better access to computers running Windows or Windows NT. (If you are running Microsoft Windows 95, these same Access Pack features are already built in. See online Help for more information.) Microsoft Windows and Microsoft Windows NT contain features that:

- Allow single-finger typing of SHIFT, CTRL, and ALT key combinations.
- Ignore accidental keystrokes.
- Adjust the rate at which a character is repeated when you hold down a key, or turn off character repeating entirely.
- Prevent extra characters if you unintentionally press a key more than once.
- Enable you to control the mouse cursor by using the keyboard.
- Enable you to control the computer keyboard and mouse by using an alternate input device.
- Provide a visual cue when the computer beeps or makes sounds.

Access Pack for Microsoft Windows is included on the Microsoft Windows Driver Library in the file ACCP.EXE.   Access Pack for Microsoft Windows NT is included in the Microsoft Application Note WNO789. If you have a modem, you can download ACCP.EXE or WNO789.EXE, which are self-extracting archive files from the following network services:

- CompuServe®
- GEnie™
- Microsoft OnLine
- Microsoft Download Service (MSDL), which you can reach by calling (206) 936-6735 any time except between 1:00 A.M. and 2:30 A.M. Pacific time.   Use the following communications settings:

For this setting	Specify
Baud rate	1200, 2400, 9600 or 14400
Parity	None
Data bits	8
Stop bits	1

- Various user-group bulletin boards (such as the bulletin-board services on the Association of PC User Groups network)
- In /SOFTLIB/MSLFILES on the Internet servers FTP.MICROSOFT.COM and WWW.MICROSOFT.COM

People within the United States who do not have a modem can order the Access packs by calling Microsoft Sales Information Center at (800) 426-9400 (voice) or (800) 892-5234 (text telephone).   In Canada, you can call (905) 568-3503 (voice) or (905) 568-9641 (text tehone).

#K$+ Keyboard Layouts for Single-Handed Users

Microsoft distributes Dvorak keyboard layouts that make the most frequently typed characters on a keyboard more accessible to people who have difficulty using the standard "QWERTY" layout. There are three Dvorak layouts: one for two-handed users, one for people who type with their left hand only, and one for people who type with their right hand only. The left-handed or right-handed keyboard layouts can also be used by people who type with a single finger or a wand. You do not need to purchase any special equipment in order to use these features.

Microsoft Windows and Microsoft Windows NT already support the two-handed Dvorak layout, which can be useful for coping with or avoiding types of repetitive-motion injuries associated with typing. To get this layout use the Windows Control Panel; consult your on-line documentation for detailed instructions. The two layouts for people who type with one hand are distributed as Microsoft Application Note GA0650. It is also contained in file GA0650.EXE on most network services and on the Microsoft Download Service.   For instructions on oining this application note see <u>Access Packs for Microsoft Windows and Microsoft Windows NT</u>.

K$+ Microsoft Documentation in Alternative Formats

People who have difficulty reading or handling printed documentation may obtain many Microsoft publications from Recording for the Blind, Inc.   Recording for the Blind distributes these documents to registered eligible members of their distribution service either on audio cassettes or on floppy disks. The Recording for the Blind collection contains more than 80,000 titles, including Microsoft product documentation and books from Microsoft Press.   You can contact Recording for the Blind at the following address or phone numbers for information on eligibility and availability of Microsoft product documentation and books from Microsoft Press:

Recording for the Blind, Inc.
20 Roszel Road
Princeton, NJ   08540

Phone: (609) 452-0606
Fax: (609) 987-8116

Manf the *Product Name* version *x* documents are also available on a CD that comes with the package.

K$+ Third-Party Utilities to Enhance Accessibility

A wide variety of third-party hardware and software products are available to make personal computers easier to use for people with disabilities. Among the different types of products available for the MS-DOS, Microsoft Windows, and Microsoft Windows NT operating systems are:

- Programs that enlarge or alter the color of information on the screen for people with visual impairments.
- Programs that describe information on the screen in braille or synthesized speech for people who are blind or have difficulty reading.
- Hardware and software utilities that modify the behavior of the mouse and keyboard.
- Programs that enable users to "type" using a mouse or their voice.
- Word or phrase prediction software that allows one to type more quickly and with fewer keystrokes.
- Alternate input devices, such as single switch or puff-and-sip devices, for those who cannot use a mouse or a keyboard.

For more information on obtaining third-party utilities, see <u>Getting More Information</u>. For more information about customizWindows and Windows NT for people with low vision, see <u>Customizing Windows or Windows NT.</u>

K$+ Customizing Windows or Windows NT

There are many ways you can adjust the appearance and behavior of Windows or Windows NT to suit varying eyesight and motor skills without requiring any additional software or hardware.   These include ways to adjust the appearance as well as the behavior of the mouse and keyboard.   The specific methods available depend on which operating system you are using.   Application notes are available describing the specific methods available for each operating system.

See the appropriate application note for information related to customizing your operating system for people with disabilities. For information on obtaining application notes, see <u>Access Pack for Microsoft Windows and Microsoft Windows NT</u>.

Operating system	Application note number
Microsoft Windows 3.0	WW0786.TXT
Microsoft Windows1	WW0787.TXT
Microsoft Windows for Workgroups 3.1	WG0788.TXT
Microsoft Windows NT 3.1 and 3.5	WN0789.EXE
Microsoft Windows95	WN1062

K$+ Getting More Information

For more information on Microsoft products and services for people with disabilities, contact:

Microsoft Sales Information Center
One Microsoft Way
Redmond, WA 98052-6393

Voice telephone: (800) 426-9400
Text telephone: (800) 892-5234
Fax: (206) 635-6100

The Trace R&D Center at the University of Wisconsin-Madison produces a book and a compact disc that describe products that help people with disabilities use computers. The book, titled *Trace ResourceBook,* provides descriptions and photographs of about 2,000 products. The compact disc, titled *Co-Net CD,* provides a database of more than 18,000 products and other information for people with disabilities. It is issued twice a year. To obtain these directories, contact:

Trace R&D Center
S-151 Waisman Center
1500 Highland Avenue
Madison, WI 53705–2280

Voice telephone: (608) 263–2309
Text telephone: (608) 263–5408
Fax: (608) 262-8848

For general information and recommendations on how computers can help specific people, you should consult a trained evaluator who can best match your needs with the available solutions. An assistive technology program in your area will provide referrals to programs and services that are available to you. To locate the assistive technology program nearest you, you can contact:

National Information System
Center for Developmental Disabilities
Benson Building
University of South Carolina
Columbia, SC 29208

Voice/text telephone: (803) 777-4435
Fax: (803) 777-6058

Backing Up Your Database

<u>**See Also**</u>

Periodically**,** you may want to back up your **<u>Visual SourceSafe database</u>**, or parts of it. You can use the SSARC (SourceSafe archive) utility to do this. The archive utility lets you save disk space on your Visual SourceSafe database server, make the Show History command work more quickly, and transport files and projects between Visual SourceSafe databases, keeping history information intact.

Examples

Archives and deletes FILE1.C and FILE2.H from the root project.

```
SSARC -d archive.ssa $/FILE1.C $/FILE2.H
```

Archives all versions of the Code project up to the version labeled "Final Beta" (inclusive) to archive file ARCHIVE.SSA.   You will be asked whether to delete archived versions from the Visual SourceSafe database.

```
SSARC "-vlFinal Beta" ARCHIVE.SSA $/Code
```

Archives the entire database to archive file ARCHIVE.SSA, without any required user input (answers No to all Yes/No questions automatically). Archived files are not deleted from the Visual SourceSafe database, so in this case the example acts as a backup command.

```
SSARC -d- -i-N ARCHIVE.SSA $/
```

Syntax

SSARC [**-c**] [**-d**] [**-v#**|*D*|*L*] [**-s**] [**-o**] [**-i**] [**-x**] [**-y**] [*archive filename*] [*files/projects to archive*]

The following table describes command-line options available with this utility. In general, options work as they would in Visual SourceSafe. For example, you can use -o with all the normal -o options, you can use -i with all the normal -i options, etc.   Exceptions to this general rule are noted below.   Also, note that the order of the hyphenated parameters does not matter.   The first non-hyphenated parameter is the archive file to create; all others are files or projects to archive.

Option	Description
-d	Deletes the archived items from the database after saving them in the archive file. -d**-** tells the archive utility *not* to delete the items from the database. If the -d parameter is not used, the default is to ask whether items should be deleted from the database (the question comes once per archive operation, not once per item).
-v[#|*D*|*L*]	-v[#|*D*|*L*] -v specifies a version, in standard SourceSafe format (number, date, or label). This is the most important SSARC parameter. If you don't use it at all, SSARC will archive all versions of all files or projects listed in the *files/projects to archive* list. If you do specify a version with -v, SSARC archives all versions before and including the specified version. SSARC does not delete the current version, even if you -vd9/9/99. Also, if you specify -vl, the label you specify is not deleted from the database (so you can later perform a Get Latest Version operation on that label from the Show History dialog box).

As an example, suppose you have a project with a history as shown:

Version   **Action**

1	Created
2	Label: Yo!
3	Checked In
4	Label: Hi!
5	Checked In

Using SSARC "-vLHi!" archives versions 1 to 4, deletes versions 1 to 3 (if specified), but leaves the label (version 4).

Note: In general, use -v# only when you want to work on an individual file. This avoids ambiguity that may lead to results different from those you expect. For example, if $/abc has 5 versions of help.h, and 200 versions of help.cpp, specifying "ssarc -v100 $/abc" would eliminate all previous versions of help.h and half the versions of help.cpp. Also, because SourceSafe doesn't show version numbers in project history,   version 100 of the project could have been created last week, month, or year.   A more precise approach is to use -vL or -vD.

-s   This parameter is optional, and use should be fairly rare. `-s` allows you to specify the path to SRCSAFE.INI and/or the data path.   For example, the syntax looks like `-sC:\VSS,OldDB` where the item before the comma is the full path to SRCSAFE.INI, and the item after the comma is the string in parentheses after the data path.   So, in this example, you presumably have the following in your SRCSAFE.INI file:

```
    Data_Path(OldDB)= .\OldData
```

If there is no comma in the specification, you are specifying a SRCSAFE.INI path but no data string.   If the first character after the `-s` is a comma, you are specifying a data string but no SRCSAFE.INI path. Without -s specified, SSARC looks for SRCSAFE.INI in the default location (in your root Visual SourceSafe folder). Most of the time, you should copy the archive utility to wherever SourceSafe is, and run it from there.

-o   Redirects output. You can use the normal SourceSafe -o options with this utility option.

-c   Allows you to enter a comment for the archive operation. If the items are deleted from the database, the comment is inserted into the SourceSafe history as part of the archived record. It is also inserted in the archive file itself, if there is one. This comment can be 511 characters long, but a minimal comment is recommended here.   This is because the comment is applied to every file being archived, and a long comment may actually make your database size increase if you are archiving a large number of files.

SSARC does not support the normal -c@FILE.TXT option, nor does it support using a custom editor for your comments.

-i   No input. You can use the normal SourceSafe -i options with this utility option.   Note that it is much safer to use -i-

	N. Choosing -i-Y will automatically overwrite files, unpin files, etc... − which might not be what you intended.
-x	Archive only *deleted* items in the files and projects specified in the Files/projects to Archive parameter*.* (Deleted items are still stored in the SourceSafe database unless the Destroy Permanently option was selected when performing the Delete operation.) Note that this parameter is not required on a subsequent restore.
-y	Allows you to specify your administrator username and password. You must be an administrator to use this utility, so SSARC will prompt you for your password information if needed and not supplied. The format for this option is `-yAdmin,Bunny`
archive filename	Specifies the archive file to create. If you don't provide an extension, the default extension is `.SSA`. If you specify the string "`NULL`", SSARC will not create an archive file at all, but will simply delete (permanently) the specified items from the database. If the specified archive file already exists, you cannot append to it − a message is displayed asking if you want to overwrite the existing file, or cancel the archive operation.
Files/projects to archive	Specifies the files and/or projects to archive. This specification can be one project, one file, or a whole list of projects and files. Wildcards are also acceptable. All of this is interpreted in exactly the same way SourceSafe interprets command-line wildcards, including the fact that $/A/*.c means all files with the .c extension in the project $/A but not in it's subprojects. There is one exception: whenever you specify a project, that project is always acted on recursively. There is no `-r` parameter.

Remarks

Before running the Archive utility (SSARC.EXE) you should always run the Analyze utility on your entire database.   Analyze will find and correct any database corruption that could prevent SSARC from finishing its task correctly.   Check the <u>Archiving Issues</u> topic for more technical details.

Check in all files to be archived before running the utility, because files left checked out will be automatically checked in, and you may lose some local file changes.

Do not use SSARC as a substitution for making regular back ups of your database.   The fastest way to recover from a catastrophic error, such as a hard disk crash, is to restore the data folder containing your original files.   Backups should be performed on a regular basis.   In addition, it is a good idea to backup your database before doing any archive operation.   This will protect you from potential corruption that could occur if archive is unable to finish due to a database corruption error or other system failure such as power outage or hard disk crash.

The archive file created by SSARC.EXE contains all of the information you specify to be archived. This file can get quite large.   As with any file, it is possible to introduce integrity errors due to faulty disk media, or minor hardware problems.   SSARC does its best to verify the consistency of the file before deleting any data from your database.   If you haven't run a disk checking utility, such as ScanDisk, in a while, it would be worthwhile to do so.   In addition, whenever backing up an archive file to another medium, such as a tape drive or writable CD Rom burner, always have the software check the validity of the file after the transfer is done.   See the documentation that came with your backup software for more details.

Important   We do not recommend using Visual SourceSafe during an archive or restore.   You can

really mess up your archive by changing the files while SSARC is processing, which means an incomplete or incorrect set of information would be restored.   Instead, use the <u>Lock SourceSafe Database</u> command on the Tools menu to lock users out of the database.

Using SSARC, you can:

- Archive files, projects, or the entire database into a special compressed file, and then restore it later with history, shared links, and all version information intact.   When you archive something from a Visual SourceSafe database using this utility, you can choose to have the object automatically deleted from the database.   The archive file can be on a separate drive.   Of course, you can also copy the final .SSA file to tape or recordable CD.
- Archive specific versions of a file.   For example, you could archive all versions of your code prior to version 3.1, effectively freeing up the space those versions use in the database. You can then restore them to the database later if needed.
- Archive a file, project, or project tree from one database, and restore it into a different database.
- Move files and projects between databases in wide-area Visual SourceSafe installations.

Some things SSARC cannot do:

- You cannot leave holes in the archive history. For example, you can't archive all versions between 11 and 19 of a file or project with 25 versions.
- You cannot move selected versions of a file or project between Visual SourceSafe databases. You can archive an entire file or project history, and then restore the entire history to another database, but not just parts of it.
- You cannot save or restore *new* versions of a file or project.   Therefore, you cannot (for instance) send all of your latest changes to a colleague, and have that person add them to their database. To archive the latest changes, first check them into SourceSafe in the normal way, and then use "SSARC -d-" to archive all versions.
- You cannot save and restore rights, checkout information, and initialization file settings.   So, for example, if a user has a file checked out when you perform the archive operation, no user will have the file checked out upon restoring it.   Also, initialization file settings, general or specific to certain projects and files, are not saved.

Backing Up Your Database See Also

Archiving and Restoring Technical Issues

SSRESTOR

-C Command-Line Option

-I Command-Line Option

-O Command-Line Option

-V Command-Line Option

Archive and Restore Error Messages

Restoring Files and Projects from an Archive File

See Also

After you have run the SSARC utility to archive your database files and projects, you use the SSRESTOR utility to recover them if and when needed. You can restore information from an archive file back to the same database it was extracted from or into a totally different database.   This gives you the capability of moving information between different **Visual SourceSafe databases** in a wide-area Visual SourceSafe installation.

Examples

Restores the project $/My Project from the archive file ARCHIVE.SSA.

```
SSRESTOR ARCHIVE.SSA "$/My Project"
```

Tests the archive file ARCHIVE.SSA for corruption.

```
SSRESTOR -t ARCHIVE.SSA
```

Lists the contents of archive file ARCHIVE.SSA.

```
SSRESTOR -l ARCHIVE.SSA
```

Restores everything in project $/A from ARCHIVE.SSA into project $/Old Versions.

```
SSRESTOR "-p$/Old Versions" ARCHIVE.SSA $/A
```

Syntax

SSRESTOR [**-p***Project*] [**-l***a*] [**-t**] [**-o**] [**-s**] [**-i**] [**-y**] [**-x**] [**-c**] [*archive filename to restore*] [*files/projects to restore*]

The following table describes command-line options available with this utility. In general, options work as they would in Visual SourceSafe. For example, you can use -o with all the normal -o options, you can use -i with all the normal -i options, etc.   Also, note that the order of the hyphenated parameters does not matter.   The first non-hyphenated parameter is the archive file to restore from; all others are files or projects to restore.

Option	Description
-p	Specifies a project within the database to restore into. (This parameter will only rarely be needed.) For instance, if you archive file $/A/HELP.C and then restore it, it will be restored as $/A/HELP.C. Only when you want to restore the same file into a different project do you have to use the -p parameter.
-l[*a*]	Lists the contents of an archive fileThe -l option only shows you the items you explicitly put into the file; -la shows you all the child projects and files as well. For example, suppose you archived $/A in an archive file. If you specify -l, SSRESTOR will simply tell you that $/A is there. If you specify -la, SSRESTOR will show you all the files and subprojects under $/A. -la is used to see what an archive file contains before proceeding with the actual restore operation.
-t	Tests an archive file for corruption. Using -t doesn't actually restore an archive, but only returns the results of a corruption test.
-o	Redirects output. You can use the normal SourceSafe -o

	options with this utility option.
-s	Specifies the location of the SRCSAFE.INI file associated with the Visual SourceSafe database you are restoring into. For example, the syntax looks like `-sC:\VSS,OldDB` where the item before the comma is the full path to SRCSAFE.INI, and the item after the comma is the string in parentheses after the data path. So, in this example, you presumably have the following in your SRCSAFE.INI file: `Data_Path(OldDB)= .\OldData` If there is no comma in the specification, you are specifying a SRCSAFE.INI path but no data string. If the first character after the `-s` is a comma, you are specifying a data string but no SRCSAFE.INI path. Without -s specified, SSRESTOR looks for SRCSAFE.INI in the default location (in your root Visual SourceSafe folder).
-i	Avoids the need for user input. You can use the normal SourceSafe -i options with this utility option.
-y	Allows you to specify your administrator username and password. You must be an administrator to use this utility, so SSRESTOR will prompt you for your password information if needed and not supplied. The format for this option is `-yAdmin,Bunny`
-c	Specifies the comment applied to the history entry for the restored item(s).
-x	Identifies which item to restore when the item has been deleted from the project. -x means restore the deleted item. For example, assume you have both a deleted and non-deleted $/a/b. If you archive and delete $/a, you actually get three projects ($/a, $/a/b, and $/a/b) deleted. What, therefore, does ssrestor $/a/b really mean? If you don't specify -x, it means use the non-deleted $/a/b. If you do specify -x, it means restore the deleted $/a/b. Even though they have the same name, these are two different projects.
archive filename	Specifies the name of the archive file to restore from.
Files/projects to restore	Specifies the files and/or projects within the archive file to restore. This specification can be one project, one file, or a whole list of projects and files. Wildcards are also acceptable. This option is interpreted in the same way SourceSafe interprets other command-line options, including the fact that $/A/*.c means all files with the .c extension in the project $/A but not in it's subprojects. However, there is one exception: whenever you specify a project, that project is always acted on recursively. There is no `-r` parameter.

Remarks

After a restoration, you need to re-create any initialization file variables and project or user rights that are needed for efficient operation in your team. This information is not saved as part of the archive.

If you have archived a shared file, restoring it into the same database will re-establish the share link. However, restoring it into another database will not result in the file being shared between databases.

Note that you cannot restore the root project to the root.   This is because its not possible to restore something that already exists.   So, if you had archived the root into ARCHIVE.SSA and wanted to restore it to an empty database, you would have to specify each project archived off: you'd have to, for example, SSRESTOR ARCHIVE.SSA $/A $/B $/C, etc.

Restoring Files and Projects from an Archive File See Also

Archiving and Restoring Technical Issues

SSARC

-C Command-Line Option

-I Command-Line Option

-O Command-Line Option

-V Command-Line Option

Archive and Restore Error Messages

-C Command-Line Option

<u>See Also</u>

Use the -C option for any command that requires a comment.   If you do not specify the -C option, Visual SourceSafe asks for a comment for each item you specify on the command line.   If you do specify the option, all items receive the same comment.

The following table lists ways to use the -C option.

Option Example	Description
-C	Ask for one comment, and then use it for all items.
-C-	Do not use a comment.
-C*text*	Use the string *text* as the comment for all items.
-C@FILE.TXT	Use the text file FILE.TXT as the comment for all items.
-C?	Return to Visual SourceSafe's default; ask for a comment for each individual file.

To set any of the option variations listed in the table as a default for the Check Out command, use the Checkout_Comment initialization variable in SS.INI.   If you have set a default in SS.INI, you can replace it for a particular command with the option -C?, and Visual SourceSafe asks for a comment for each individual file (Visual SourceSafe's default).

You can set the Comment_Editor variable in SS.INI to invoke an executable editor of your choice as an alternative to the prompt-driven method of comment entry Visual SourceSafe uses.

Note   Comments can be up to 63 characters long for the Check Out command, and 4K characters long for other comments.

See Also

Checkout_Comment Initialization Variable

Comment_Editor Initialization Variable

-O Command-Line Option

The -O option controls the output from commands that might display large amounts of information.   If you do not specify a -O option, the information scrolls by on the screen too quickly to be read.

The following table lists ways to use the -O option.

Option Example	Description
-O	Format output in pages, waiting for a keystroke each page.
-O-	Do not provide output other than errors.
-O@FILE.TXT	Send all output to the file FILE.TXT. This is the command-line method for exporting reports, such as the History report, to a filename or printer. If the file named already exists, output is appended to that file.
"-O&FILE.TXT"	Output standard errors to the file FILE.TXT, in addition to the command output. Note the quotation marks required on NT platforms, and optional on others. If the file named already exists, output is appended to that file.
-O&-	Do not provide output.

-V Command-Line Option

The -V option specifies earlier versions of files or projects.   For example, you may want to view a file as it looked last week, or a project exactly as it was last month.

The following table lists ways to use the -V option.

Option Example	Description
-V14	Display version 14.
-Vd2-29-92	Display the version dated 2-29-92.
"-VL Final Beta"	Display the version with the **label** Final Beta.

The -V option displays all files and projects identified by the **version number** specified on the command line.   You can also specify a version number for one particular item, by following that item with a semicolon and then the version number.

The following table shows how to specify a particular item.

Option Example	Description
HELP.C;14	Display version 14 of HELP.C.
SHOW.PRG;d2-29-92	Display SHOW.PRG as it appeared on 2-29-92.
"$/CW;L Beta 1"	Display the version of the project $/CW with the label Beta 1.

Occasionally, you will want to reference a time with the date in a Visual SourceSafe command.   To do this, reference the time along with the date using the -vd switch.

For example, to list the history of the $/TEST project from 9 a.m. to 3 p.m. on 3/3/95, issue the command:

```
ss history $/test -vd3/03/95;3:00p~3/03/95;9:00a
```

In the above command "a" or "p", refers to a.m. or p.m. respectively.   And the ~ is used to indicate you want the history between the dates and times specified. Note that the later date and time must be specified first.

-R Command-Line Option

The -R option is used with Visual SourceSafe commands that operate on projects; it makes the commands recursive to subprojects.   Normally, commands act only on the project and the files in it: If you get a project, you get all the files that are in that project.   Subprojects, and the files and subprojects inside them, however, are ignored.   When you use the -R option to make a command recursive, the command acts on the entire project hierarchy.   For example, if you type Get *projectname* -R, directories are created for that project's subprojects, and the subprojects inside of them, and so on.

-G Command-Line Option

The Get Latest Version command copies files from a Visual SourceSafe project into a folder.   Visual SourceSafe gets files in response to the Get Latest Version command and also in response to many other commands such as Check Out, Undo Check Out, and Share.

For any of these commands, the -G option can be used to change the nature of the Get Latest Version operation.   Many of these options provide command-specific overrides that you can set in your SS.INI file.

The following table lists ways to use the -G option.

Option Example	Description
-G-	Don't perform a Get Latest Version.
-GL*path*	Copy a file to the specified folder, not to the current or **working folder**.
-GTC	Give the **local copy** the current date and time.
-GTM	Give the local copy the date and time that the file was last modified, not the current date and time.
-GTU	Give the local copy the date and time that the file was last updated, not the current date and time.
-GCC	Determine whether the local copy of the file is up-to-date by comparing the file's contents.
-GCK	Use a checksum to determine whether the local copy of the file is up-to-date.
-GCD	Use a date and time comparison to determine whether the local copy of the file is up-to-date.
-GR	Specify the carriage return as the end-of-line character in text files.
-GN	Specify the linefeed as the end-of-line character in text files.
-GRN	Specify the carriage return–linefeed pair as the end-of-line character in text files. This is the default on the PC.
-GF	Turn on the Force_Dir initialization variable for this command.
-GF-	Turn off the Force_Dir initialization variable for this command.
-GWA	Display a dialog asking the user to choose between replacing, skipping, or merging writable files on Get Latest Version and Check Out operations.
-GWR	Replace writable files on certain operations (Get Latest Version and Check Out).
-GWS	Skip writable files on certain operations (Get Latest Version and Check Out).
-GWM	Merge writable files on certain operations (Get Latest Version and Check Out).

-S Command-Line Option

The Smart_Mode initialization variable in SS.INI puts Visual SourceSafe in SmartMode, in which it follows many actions with logical next steps.   With SmartMode on:

- After you create a project, SmartMode sets it as your current project.
- When you delete or destroy a file, SourceSafe automatically deletes the corresponding local file.
- When you rename a file, SourceSafe automatically renames the corresponding local file.

This command-line option overrides the Smart_Mode variable for a particular command.

The following table lists ways to use the -S option.

Option Example	Description
-S	Turn on SmartMode for this command.
-S-	Turn off SmartMode for this command.

See Also

Smart_Mode Initialization Variable

Smart_Mode Initialization Variable

Enables and disables Visual SourceSafe's SmartMode feature.

Syntax

Smart_Mode = {Yes|No}

Example

Disables Visual SourceSafe's SmartMode feature:

```
Smart_Mode = No
```

Remarks

Smart_Mode tells Visual SourceSafe whether or not to use its SmartMode features.   The SmartMode features are:

- When you create or recover a **project**, Visual SourceSafe sets the new project as the **current project**.
- When you delete or destroy a file, Visual SourceSafe deletes the corresponding local copy.
- When you rename a file, Visual SourceSafe renames the corresponding local **copy**.

You can also set this variable in your SS.INI file by using the Remove Local Files After Delete option on the Local Files tab in the SourceSafe Options dialog box.

-W Command-Line Option

When you do not have a file checked out, your **local copy** is locked, or made **read-only** to remind you that you should not change it.   The Use_ReadOnly variable in SS.INI changes this default behavior.   The -W variable changes it for a specific command.

The following table lists ways to use the -W option.

Option Example	Description
-W	Make local files writable.
-W-	Make local files read-only.

-N Command-Line Option

On operating systems that support long filenames, the -N option changes between long and short filename mode one command at a time.

The following table lists ways to use the -N option.

Option Example	Description
-NL	Use long filename mode.
-NS	Use short filename mode.

-I Command-Line Option

There are a number of circumstances in which Visual SourceSafe commands ask for input from the user−warnings, and dialog boxes containing yes or no questions, for example.   This is not wanted when you write scripts or macros that execute the Visual SourceSafe command line from inside other programs.   In these cases, you can use the -I- option to ensure that Visual SourceSafe never asks for user input.

When using -I, it is almost always a good idea to use the -C option to avoid the comment prompt.   The following table lists ways to use the -I option.

Option Example	Description
-I-Y	Instruct Visual SourceSafe to answer Yes to all Yes or No questions.
-I-N	Instruct Visual SourceSafe to answer No to all Yes or No questions.

-Y Command-Line Option

The -Y option specifies a **username** in case you occasionally want to execute a command as a user other than yourself.

The following table lists ways to use the -Y option.

Option Example	Description
-YMaria	Specify a username.
-YMaria,TooSecret	Specify a username and **password**.

-? Command-Line Option

Use the -? option with any command to get Help on that command.   You can also use the -H option to obtain online Help.

Option Example	Description
ss Share -?	Displays Help on the Share command.

Comment_Editor Initialization Variable

Determines which editor is used to enter comments.   Applies only to the Visual SourceSafe command line program (SS.EXE)

Syntax

Comment_Editor = *editor executable file*

Example

Sets B.EXE as the editor for entering comments:

```
Comment_Editor = C:\EDITORS\B.EXE
```

Remarks

Visual SourceSafe's normal method for receiving comments is a command-line prompt.   This variable overrides that, telling Visual SourceSafe to bring up the specified editor whenever it has to prompt for a comment.   This enables you to enter comments in a familiar editor.

If the Editor initialization variable is set, you can set the Comment_Editor to override it on comments; setting it to blank restores prompt-driven comment entry while allowing an editor for viewing.   You can also set this variable in your SS.INI file under the Comment Prompt options on the Command Line Options tab in the SourceSafe Options dialog box.

Checkout_Comment Initialization Variable

Sets the -C command-line option for the Check Out command.   Applies only to the Visual SourceSafe command-line program (SS.EXE).

Syntax

Checkout_Comment = *valid -C setting*

Example

Causes check out to not prompt for comments by default:

```
Checkout_Comment = -
```

Remarks

Visual SourceSafe gives you the option of providing a comment for each file you check out.   This is the default behavior.   The check-out comment tells other users why you checked out the file, and is then erased when you check in the file with a permanent comment describing the change you made.

The Checkout_Comment variable controls Visual SourceSafe's prompt for a check-out comment. The following settings are all valid.

Does not prompt for a comment at check out:

```
Checkout_Comment = -
```

Prompts for one comment for all checked out files:

```
Checkout_Comment =
```

Reads the specified text file as a comment:

```
Checkout_Comment = @filename
```

Uses the specified comment:

```
Checkout_Comment = comment
```

You can also set this variable in your SS.INI file under Check Out Comments on the Command Line Options tab in the SourceSafe Options dialog box.

Note   The limit for the Check Out comment is 64 characters.

File <file> is pinned to version <number> in project <project>.

You are trying to archive and delete a file that is pinned to one of the archived versions in some project. You need to unpin it in the specified project before you can delete it from the SourceSafe database.

Enter Yes, and Visual SourceSafe unpins the file and then deletes the versions from the database.

Enter No, and only earlier versions of the file are deleted from the database. The specified version remains pinned.

Compression read failed.

You are trying to read a compressed archive file. SSARC.EXE uses compression to store the data in the .SSA file.   In this case, archive was unable to read the compressed data, most likely due to a corruption in the .SSA file.   Restore the .SSA file from a backup copy and try again.   You can also run SSRESTOR -t to check the file.

Compression header corrupt.

You are trying to read a compressed archive file. SSARC.EXE uses compression to store the data in the .SSA file.   In this case, archive was unable to read the compressed header data, most likely due to a corruption in the .SSA file.   Restore the .SSA file from a backup copy and try again.   You can also run SSRESTOR -t to check the file.

File or project %s not found.   Continue?

You specified a file or project on the command line that wasn't in the database/archive file.

Enter Yes, and Visual SourceSafe continues with the remaining specified files.
Enter No, and Visual SourceSafe cancels the operation so you can retype the file/project names.

Delete archived information from database?

You did not specify -d or -d- on the SSARC command line, and no -i-y or -i-n was specified
SourceSafe needs to confirm whether you want to delete what you archive.

Enter Yes, and Visual SourceSafe deletes the archived information from the SourceSafe database.

Enter No, and archived information is not deleted from the SourceSafe database.

Only ADMIN can run this utility.

You are trying to archive a file, or restore an archive, without specifying the administrator name and password.   Use the -y option for this.   The format for this option is `-yAdmin,Bunny`.

Incompatible archive file version <number>.

The .SSA file specified is an older format than the version of the archive utility you are using. Contact Microsoft Technical Support.

Incompatible compression version <number>.

The compression technology used to create the .SSA file requires the same version to decompress it on restore.

CRC mismatch for <file>.

The CRC (cyclical redundancy check) used to create the .SSA file does not match the current CRC.
Run SSRESTOR -t to test the .SSA file for corruption.   If necessary, restore the .SSA file from backup
and try again.

Signature mismatch in archive file.

The archive signature used to create the .SSA file does not match the current archive signature.   Run SSRESTOR -t to test the .SSA file for corruption. If necessary, restore the .SSA file from backup and try again.

Archive file is corrupt.

Your .SSA is corrupt. First, try running SSRESTOR -t. If this verifies the corruption, restore your .SSA from a backup and try again.

Nothing to do.

You specified archive or restore parameters that if run wouldn't actually do anything.   For example, you specified a path that doesn't exist, and then specified Yes to continue.   Or, you specified -d- with NULL as the .SSA file.

<item> does not exist and the archive file contains only version ranges.

You archived with -v, but are trying to restore to a file or project that doesn't exist.   The file must exist if the .ssa file has only version ranges in it.

Without an archive file you will not be able to restore any deleted data.   Continue anyway?

You specified NULL for the .ssa file name, along with -d, which means the data will be removed permanently, with no backup.   Choose Yes to destroy your data, No to cancel the operation.

Parent <parent> for file <file> (<project>) does not exist.

The parent for the specified file doesn't exist. In some cases, this could mean the file is corrupted. This parent will be skipped for this archive operation, and the archive will continue with the next file (if applicable).   Run ANALYZE.EXE at your earliest convenience to detect and fix any potential corruption.

Name information for offset <name> not found in archive file.

The name will be truncated to <name>.   A long name (from names.dat) was archived but for some reason that name couldn't be found on restore.   This may be due to file corruption (you should run ANALYZE.EXE to check this). The name was truncated to the new value.   After the archive operation, rename the file in SourceSafe to give it the correct name again.

Checkouts for <file> (<project>) deleted.

You archived and deleted a file that was checked out.   Archive removes the file's checkout status before deleting the file, with no chance for a check in. You can do a recursive status search for checked out files before archiving to avoid this problem.

\<project> still exists in the database and cannot be restored to a different name.

You used the -p switch on restore, but the project you are restoring already exists in the database. Specifying archive parameters that would create a new copy of the project is not allowed.   Either use the same project name, or delete the existing project before doing the restore.

<file> exists and archive file does not have version ranges. Cannot restore.

You tried to restore a file for which a version range does not exist in the given .ssa file.   Most likely, the file has been archived several times and you need to restore from a different .ssa file first.

The path <path> is already in use.   Cannot restore path from archive file.

You tried to restore a project to a name that is already in use.   For example, you archived and deleted $/A, then created a completely different project named $/A.   The project archived in the archive file is a different project than what exists now, and therefore a restore is not possible.

You can rename the existing project and re-run SSARC.EXE to recover this project information.

Destination database is not the same as the original.   Some links may not be restored.

You restored a file to a different database than the original.   In this situation, some shared or branched projects or files may not be restored correctly.

<file> has been automatically renamed to <file> in project <project> to match the current file name in the database.

A file was archived as file1.cpp, but then renamed to file2.cpp.   Archive changes the file name upon restoring the project to keep the project in synchronization.

<file> already exists in project <project>, and is not the same as the archived file.   No restore possible.

You tried to restore a file to a project, but that name is now in use by a different file.

<file> (<project>) is a branch parent for a file which is not being archived.   No versions deleted.

If you use the -v parameter on a range, and some of those versions are used by a branch child, the archive range can get moved forward so they are left on disk.   This means that some files may be left in the project even though you expected some versions to be deleted.

Deleting the root is not allowed.

You can use the -v parameter against the root folder, but can't delete the root.

\<file\> is not in the archive file.

You tried to restore a file that isn't in the .ssa file.   Use -la to find valid files.

Archiving and Restoring Technical Issues

SSARC performs several passes during its processing.   These passes are:

1. Gather a list of files to archive.
2. Save the data to the archive file.
3. Check the consistency of that file.
4. Optionally delete the files or version ranges as specified by the -d flag.   Due to the potential long running time of such a large amount of processing, all users should be logged out of the system if possible.   Any changes users make while archive is running could potentially cause corruption. For example, Pass 2 might save all 10 versions of source.cpp, but if a user checks in a new version before Pass 4, their data might be lost.

SSRESTOR performs two major passes:

1. Load the list of information from the archive file.
2. Restore each project or file specified, using recursion if necessary.

As with the archive step, all users should be logged out of the database if possible.   If you are performing a very small archive or restore operation, it may be sufficient to ask users to refrain from working with the affected projects.   You will always be safest, however, if you keep all users out of the database using the the <u>Lock SourceSafe Database</u> command on the Tools menu of the Visual SourceSafe Administrator.

You can restore files that came from one database to a different database.   Visual SourceSafe internally uses unique 8 character file names to track your files.   These names are not unique for each database.   To solve this problem, SSRESTOR gives the files you are restoring a new name in the target database.   In some cases files rely on each other, such as branched and shared projects. Since the name mapping only applies to one restore operation, it might not be possible to restore all information in this situation.   For example, if file source.cpp was shared between $/a and $/b, and you restore only $/a/source.cpp to a new database, then restoring $/b/source.cpp in another step will create a separate file.   The second operation has no knowledge of the first.   To solve this problem, either restore both projects at once, or manually re-establish the share between the two projects.

Other Issues

Archiving and deleting a shared file from one project does not always result in the file being deleted from the database (because it is still needed in the other projects it is shared with). Sometimes a shared file will be deleted − if you delete all projects that share it or if you specify the file specifically. For example, "SSARC -d SHARE.SSA $/A/B" does not delete any files in $/A/B that are shared with $/A/C.   "SSARC -d SHARE.SSA $A/B/SHARED.TXT" will delete all occurrences of SHARED.TXT even if it is shared to another project.

Similarly, branched files being archived and then restored to different databases may not have the behavior you expect. For example, if you had file FILE1.C in project $/A/B and you shared and branched at version 3 with $/A/C, FILE1.C in $/A/C uses the history of FILE1.C in $/A/B.   So, if you archive and delete FILE1.C in $/A/C, FILE1.C still exists in the database (meaning you may not have saved as much drive space as you expected) because it is still in $/A/B.   If you use the Show History command on FILE1.C in $/A/B, it will show history up to version 3 where you branched.   If you archive and delete FILE1.C in $/A/B, FILE1.C will remain in the database because it is still in $/A/C. When you check Show History on FILE1.C in $/A/C, you will see versions earlier than 3 marked as Archived. If you archive and then delete $/A, FILE1.C will be deleted from the database.

Archive and Restore Error Messages

While running SSARC.EXE or SSRESTOR.EXE, you may encounter one of the following error messages.   Select the message to jump to an error topic that explains the problem (messages are sorted alphabetically).

<file> (<project>) is a branch parent for a file which is not being archived.   No versions deleted.

<file> already exists in project <project>, and is not the same as the archived file.   No restore possible.

<file> exists and archive file does not have version ranges.   Cannot restore.

<file> has been automatically renamed to <file> in project <project> to match the current file name in the database.

<file> is not in the archive file.

<item> does not exist and the archive file contains only version ranges.

<project> still exists in the database and cannot be restored to a different name.

Archive file is corrupt.

Checkouts for <file> (<project>) deleted.

Compression header corrupt.

Compression read failed.

CRC mismatch for <file>.

Delete archived information from database?

Deleting the root is not allowed.

Destination database is not the same as the original.   Some links may not be restored.

File <file> is pinned to version <number> in project <project>.

File or project %s not found.   Continue?

Incompatible archive file version <number>.

Incompatible compression version <number>.

Name information for offset <name> not found in archive file.

Nothing to do.

Only ADMIN can run this utility.

Parent <parent> for file <file> (<project>) does not exist.

Signature mismatch in archive file.

The path <path> is already in use.   Cannot restore path from archive file.

Without an archive file you will not be able to restore any deleted data.   Continue anyway?