

Troubleshooting “in use” Errors

All “in use” errors are an indication that something is wrong on the file system, or back end, where Version Manager is trying to work. Version Manager returns this error when it cannot write to the file system for one reason or another.

The first step is to consider the error. Does it specify that the FILE is in use, the PROJECT is in use, the PROJECT ROOT is in use or a USER is in use? What entity is returned in the error? Is it the archive file, the workfile, the project name, the user name or some other entity?

The second step is to consider the action that returned the error. Was it a file-level operation such as a check out, check in, adding/changing/deleting version labels or adding/changing/deleting promotion groups? Was it a project-related action such as adding a file, creating a sub-project or deleting an existing member of that project?

What follows next is the majority of the troubleshooting...

Tip: “in use” errors are 99 % of the time the result of one of three following conditions:

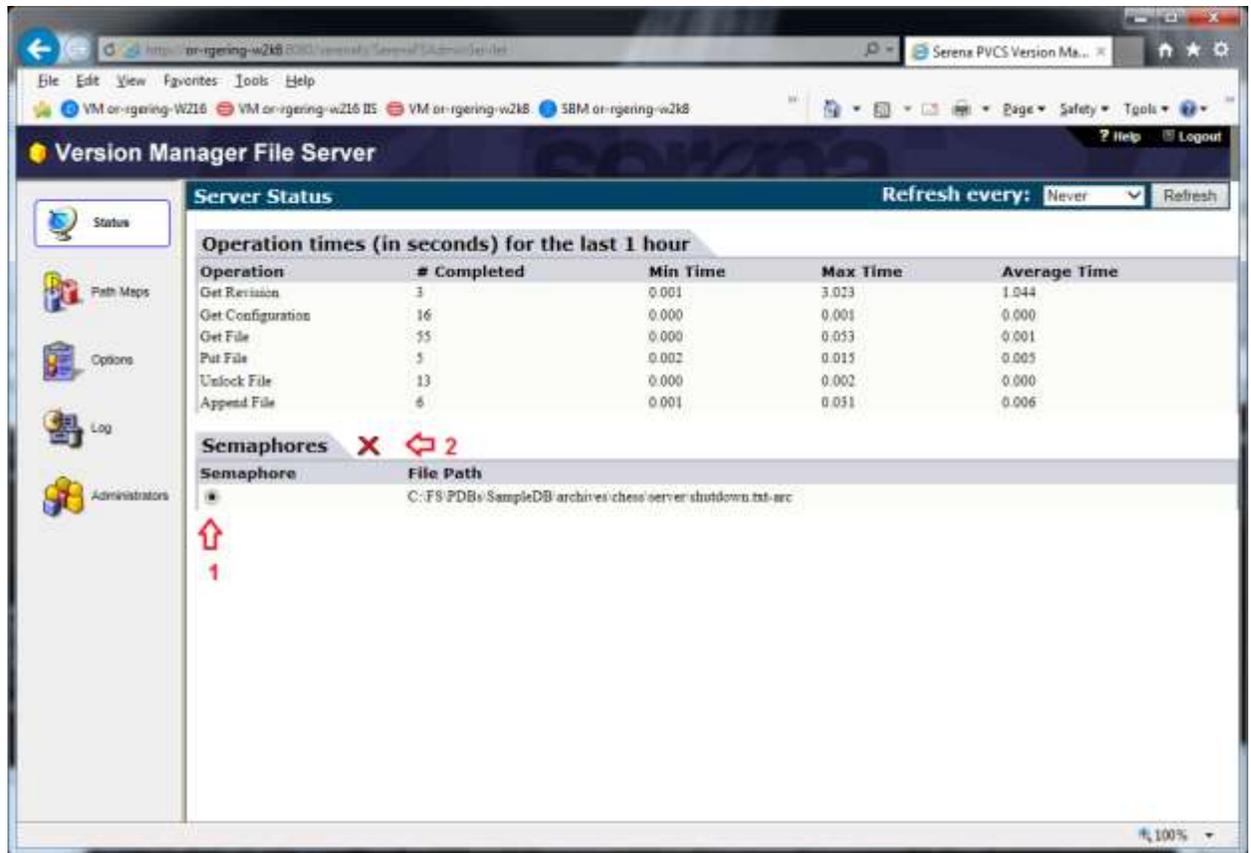
- A. There is a semaphore file (*.sem) or a lock file (*.lck) that exists after another operation when it should not. This can be caused by an abnormal termination of a process, like a network or hardware failure, or the user does not have sufficient permissions on the file system to delete the file.
- B. There is a network lock or a file handle open that is preventing access to the file that Version Manager is trying to modify or delete. Some process has the file open and is denying access to any other process.
- C. There are insufficient network permissions to create, modify or delete files as needed by Version Manager to complete the process.

FILE “in use” Errors

If the action being performed is a Check Out / Check In, or any action that would cause a change in the archive information (labeling or assigning promotion groups), the error is most likely going to be a **FILE** “in use” error caused by a stale semaphore file.

A semaphore or lock file is just an indicator to keep other users from modifying the file it was created for while you’re working with it, so its contents are not important. A semaphore file currently is an empty file (0 byte) when the VM File Server is not used. The VM File Server records information in this file on the user ID and host who created the semaphore, to help identify the origin of stale files.

1. To resolve “in use” errors caused by a stale semaphore, that file must be removed. For project databases accessed through the PVCS VM File Server, go to the Version Manager File Server Admin web page at <http://localhost:8080/serenafs/Admin> and remove the semaphore from the Status tab:



On Windows you can also use the Start menu shortcut All Program | Serena | PVCS Version Manager | Version Manager File Server Admin.

⚠ DO NOT manually delete semaphores from the file system (i.e. outside the Version Manager File Server Admin page) if the File Server is running, unless specifically directed by Support. Doing so will result in a discrepancy between the in-memory view of semaphores and those existing on disk, results in clients reporting the same error even though the file is gone.

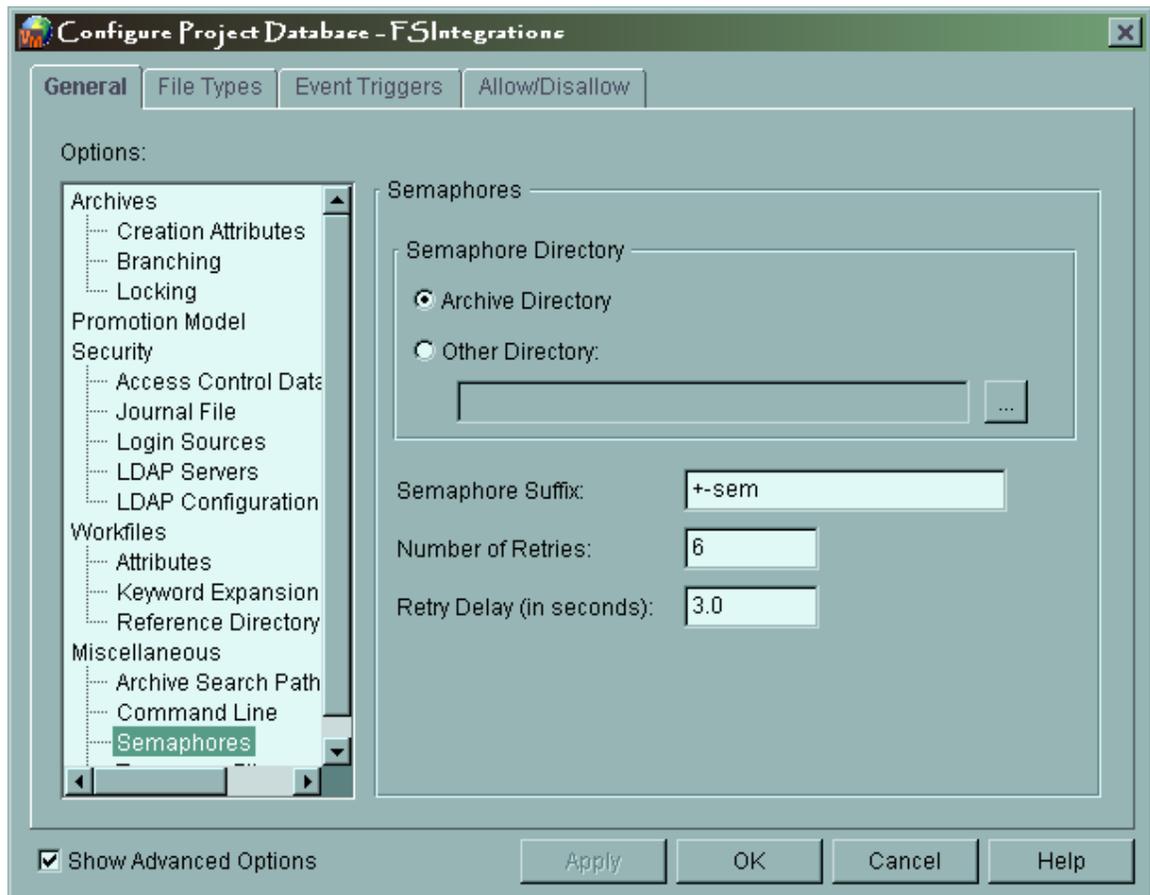
Only if the Version Manager File Server is not being used, or if that server is not running, can semaphore files safely be deleted from the file system directly.

Start by locating the semaphore file. By default, the file will have a **-sem** appended to the end of the archive suffix if the version of VM is 6.5.00 or later, and a **\$** in the suffix if the project was created with a release of Version Manager 6.0 or older.

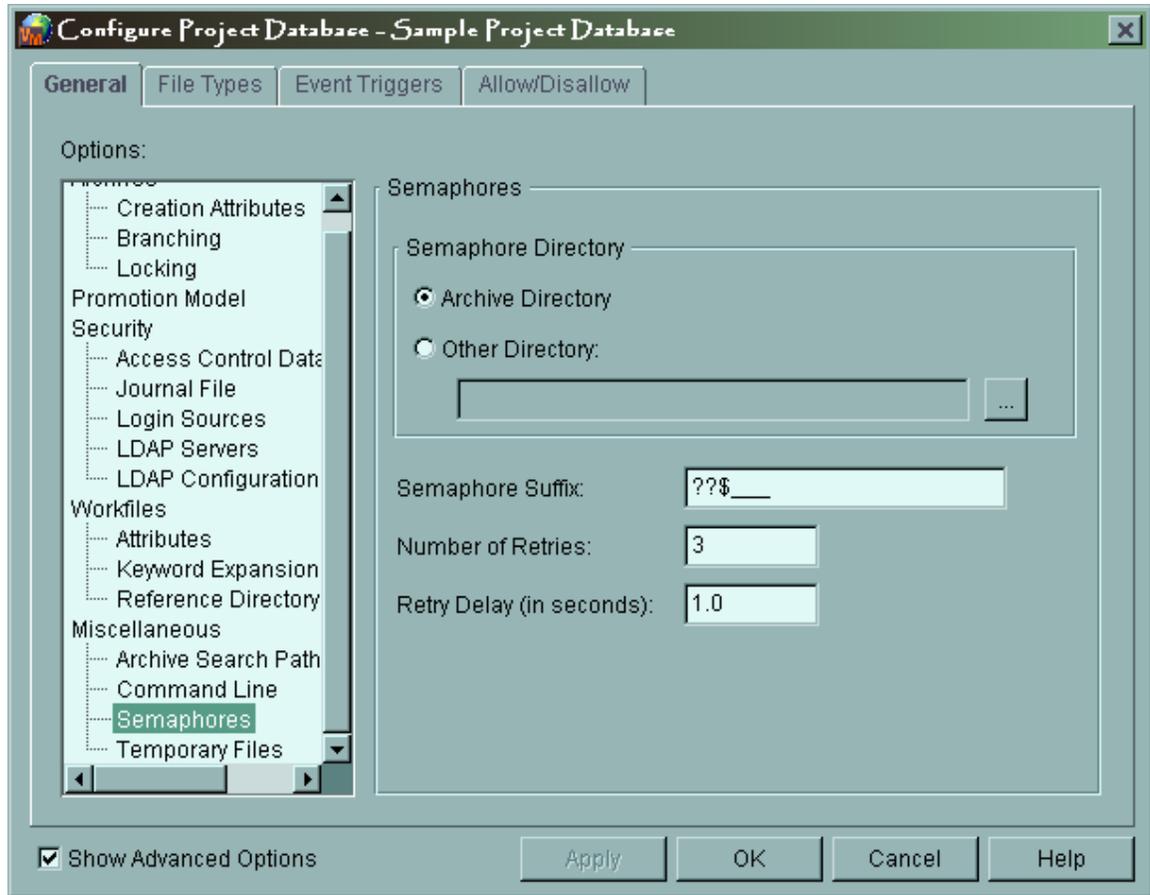
To determine what the semaphore suffix is for your project database or project, right click where the configuration file level is and select "Configure Project". A project database will have a configuration file by default. A project or subproject may have one and can be visually identified by the manila folder icon having a black tab where one would put a file folder label on a real manila folder (📁).

In the dialog that comes up, the first tab (General) will have a number of categories as shown below. Under "Miscellaneous" is the option for "Semaphores". The Semaphore Suffix will help identify what to look for in semaphore files.

In the example below, the semaphore suffix appends a **-sem** to the file name of the archive suffix. The semaphore for a .doc-arc file would be .doc-arc-sem and for a .java-arc file it would be .java-arc-sem.



The following example uses a VM 6.0 style semaphore suffix, which replaces the third character of the file suffix with a \$. So a semaphore for .doc-arc archive would be .do\$-arc, for .txt-arc it would be .tx\$-arc and for .java-arc would be .ja\$a-arc. If the project also uses VM 6.0 style archives suffixes, those files would end in .do\$, .tx\$ and .ja\$a



2. There is a network lock on the file, a file handle is open or a process still is trying to access a file. Try renaming the archive file through the file system using Windows explorer or some other means. An access violation or a message that the file is in use will be returned by the OS if there is a network lock or if a process still has a hold on the file. To resolve this, restart the system where the file resides or use a tool designed to identify and clear such locks (usually available to IS or IT teams).
3. The user does not have appropriate permissions to create, modify or delete files related to the operation.

For the desktop client, or GUI, without the Version Manager File Server, the end user's permissions to the network is where the focus is on this step of troubleshooting. Identify the directories involved with the process. For example, a check out could include:

- the archive directory (for Releases 8.0.x and later using the File Server and split archives, the archive directory refers to the directory where the archive with the metadata resides as well as the revision library or revision path)
- the temp directory (defined in the configuration file or by the system's TEMP and TMP environment variables or the default C:\TEMP on Win32 platforms or /tmp on UNIX platforms) or directories specified in the configuration file "Temporary Files" (translates to

ArchiveWork and WorkDir directives in the configuration file). For Releases 8.0.x and later, it is prudent to not use the ArchiveWork and WorkDir directives.

- the workfile directory
- the semaphore directory (if a directory other than the archive directory is used)
- the journal file directory (if a directory other than the archive directory is used). An “in use” error can also occur if the journal.vcs file is read-only.

If the action being performed is a Get, the cause is most likely an inability to write the destination workfile. Make sure you have the ability to create that file in the target directory. If the file already exists, make sure it isn't open in another application (e.g. don't Get a .doc/.docx file if that file is open in Word.) This could also explain a failure to perform a Check Out operation, though a stale semaphore (#1) is more likely.

A Check In would include all of the above directories AND the reference directory, if enabled.

Adding workfiles includes a check out and check in process, so it too would use all of the directories above. There are also edits made to the metadata files (pvcsproj.ser and pvcsid.ser files) for that project (See information on “project ... in use” errors).

Adding, changing or removing version labels or promotion groups do not use the reference directory or the workfile directory.

To verify a user's permissions, create a simple text document in the directory, make changes to the file and delete it.

When using the VM Web Client or using one of the other clients and accessing the data through the Version Manager File Server (feature available with VM 8.0 and later), the end user's permissions are not used, but the owner of the VM Web Application Server process will be the user making changes to the files on the file system.

If the process is running as a service on a Windows machine, the user that is listed in the “Log in as” properties of the server will be the user permissions to check. If running as an application, where it is manually started and left to run on the server, the user that is logged into the Windows machine will be the user profile to check for permissions. On UNIX or Linux systems, the user that launched the pvcsstart script will be the user whose permissions need to be checked.

The PVCS Version Manager Installation Guide/Merant Version Manager Installation Guide includes a section called “Protecting Program Files and Project Data” that has a matrix of suggested file system permissions.

An “in use” error can also occur when the metadata for a project database is being changed. Instead of using the semaphores used in the previous example, Version Manager creates .lck files.

When modifying a configuration file, access control database or serialization (.ser) file, Version Manager creates a corresponding file with the same name as the file being modified but with a file extension of .lck

Project “in use” errors

These errors will occur when you are adding or deleting objects for a project or subproject.

Project files are named pvcsproj.ser and are placed all throughout the project database for the project structure side. There will be one at the root of the project database and also one for each *.prj directory and that translates to one for each project or subproject in the project database. Because they are not named similarly to the project, it may be difficult to find the corresponding *.prj directory in order to identify the problem.

One method to find the corresponding *.prj directory is to use the PCLI ScanPDB command.

An example command run against the SampleDB project database provided with Version Manager:
`pcli scanpdb -v -idadmin -prC:\Program Files\Merant\vm\common\SampleDB >scanpdb.txt`

You will see in the output lines like the following:

```
005600 - Project:      Reading children of "/bridge/hlp" from
D:\Programs\Merant\8.0.0.3\vm\common\SampleDB\Ppjxd0m.prj\Pku6p9r.prj\pvcsproj.ser, Parent ID = PvcId: 125
```

Another method is to use the PCLI List command `pcli list -caDirectory "Entity"`

Example:

```
C:\>pcli list -prD:\programs\merant\8.0.2.3\vm\common\sampledb -caDirectory /chess
```

```
Merant Version Manager (PCLI) v8.0.2.0 (Build 651) for Windows NT/80x86
Copyright 1985-2004 Merant. All rights reserved.
```

```
chess
```

```
Directory=D:\Programs\Merant\8.0.2.3\vm\common\SampleDB\Pc4dxjh.prj
```

When the pvcsproj.ser file for a project is modified by Version Manager, a pvcsproj.lck file is created to keep others out while this process is completing and a pvcsproj.new file will be created with the changes.

(The existing pvcsproj.ser file is renamed to pvcsproj.old and the pvcsproj.new file is renamed to pvcsproj.ser. When these steps are completed, the pvcsproj.lck file will be deleted. The pvcsproj.old file is kept in case the pvcsproj.ser file is corrupted. The old file can be renamed to pvcsproj.ser and the project information, minus the changes from when that file was the current pvcsproj.ser file, will be recovered.)

If you find a stale .lck file, delete it like a semaphore using the instructions from pages 1 and 2.

Project root “in use” errors

A common scenario for this error is a new user accessing a project database for the very first time.

Whenever a new user accesses a project database they are assigned an entity ID. The ID number is taken from the pvcsid.ser file in the root of the project database and then the integer that is stored in that file is incremented by 1. While this file is being updated, the file pvcsid.lck is created to prevent updates

from other users.

(During the update process the file pvcsid.new is created with the new entity ID number, an existing pvcsid.old file will be deleted, pvcsid.ser is renamed to pvcsid.old, and pvcsid.new is renamed to pvcsid.ser. Following this, pvcsid.lck is deleted.)

Additionally, a new U*.dir folder is created the pvcsuser subdirectory of the project database to store settings for the user. This will include information about any private workspaces that the user creates for themselves and their workspace settings. Also, the allusers.ser file at the root of the pvcsusers subdirectory is modified to include information connecting the user and their entity ID to the corresponding U*.dir folder in that structure. As with pvcsid.ser, an allusers.lck file is created during the update.

An error that states that the project root could not be accessed because it is “in use” often indicates that a pvcsid.lck or allusers.lck file was left behind from a failed attempt to update pvcsid.ser or allusers.ser. Should you find a stale .lck file, delete it like a semaphore using the instructions from pages 1 and 2.

It could also mean that the user does not have sufficient permissions to the file system where the project database resides in order to make the necessary changes to the project database structure as explained above, although that only applies when the VM File Server is not being used. If the VM File Server is used, make sure the user ID this runs as has sufficient privileges to update the repository on disk. It may be necessary to solicit assistance from the IT department or network administrators at the customer’s company or site.

User or workspace “in use” errors

When a user is accessing a project database for the first time or an existing user is creating a new public or private workspace, they may encounter an error that the user or the workspace is “in use”. Again, this could mean that there is a .lck file in the pvcsfolder, that the end-user or VM File Server user does not have sufficient permissions in the pvcsuser folder under the project database, or that there is corruption in the metadata, specifically the allusers.ser file.

Should you find a stale .lck file, delete it like a semaphore using the instructions from pages 1 and 2.

Permissions issues should be relatively easy to resolve. Corruption of the allusers.ser file could possibly be resolved by renaming the allusers.old file to allusers.ser as the .old file has the last known good state before it was changed. Renaming the allusers.old to allusers.ser would not include the last changes, but at least would be able to restore access to the project database user and workspace information. If renaming the allusers.old file does not change the behavior, it may be necessary to rebuild the project database.

For further assistance, contact Micro Focus Support (<https://www.microfocus.com/support-and-services/contact-support/>)