CMVC frequently asked questions: overcoming problems when extracting files, releases and levels

Document Number TR 29.3093

Angel Rivera, Lee Perlov

VisualAge TeamConnection/CMVC Development
IBM Software Solutions
Research Triangle Park, North Carolina

ABSTRACT

This technical report provides answers to frequently asked questions made by CMVC users for overcoming and diagnosing problems when extracting files, releases and levels, when using the following CMVC commands:

- File -extract
- Release -extract
- Level -extract

ITIRC KEYWORDS

.

- CMVC
- File -extract
- Release -extract
- Level -extract
- NFS mount
- vmstat

ABOUT THE AUTHORS

ANGEL RIVERA

Mr. Rivera is an Advisory Software Engineer and team lead for the CMVC development group; he is also a member of the VisualAge TeamConnection development team. He joined IBM in 1989 and since then has worked in the development and support of library systems.

Mr. Rivera has an M.S. in Electrical Engineering from The University of Texas at Austin, and B.S. in Electronic Systems Engineering from the Instituto Tecnológico y de Estudios Superiores de Monterrey, México.

LEE R. PERLOV

Mr. Perlov is an Advisory Software Engineer in the VisualAge TeamConnection/CMVC development group. He started working for IBM in 1985 in Gaithersburg, Md, working in the Federal Systems Division on various projects for the United States intelligence community. He then moved to RTP to work on library development and support.

Mr. Perlov received a B.S.Acc degree in Accounting from the University of Florida in 1983. He also completed two years of graduate work in the Department of Computer Science at the University of Florida.

CONTENTS

FIGURES

INTRODUCTION

This technical report provides answers to frequently asked questions made by CMVC users for overcoming and diagnosing problems when extracting files, releases and levels, when using the following CMVC commands:

- File -extract
- Release -extract
- Level -extract

This technical report is organized as follows:

- Chapter “Overview of possible problems” on page 5 provides an overview of some possible problems which can be quickly diagnosed and answered.

- Chapter “Analysis and diagnostics of the problem” on page 9 provides some suggestions on how to perform the analysis of the extraction problem and how to better diagnose it.

- Chapter “Adjusting for intermittent failures” on page 15 describes the use of NFS, some common problems associated with this usage and some alternatives on how to overcome related problems.

  We have learned from painful experience that even well managed networks can be intermittently unreliable due to NFS. As a result, we have provided several means of adapting the behavior of CMVC.

- Chapter “What to do when everything else fails?” on page 23 describes what to do when the main troubleshooting techniques and workarounds do not seem to work.

- Chapter “Information about using NOMOUNT” on page 29 describes the use of the new 2.3.x function for performing a Release or Level extract to a local file system (in the CMVC family host), bypassing NFS.

- Chapter “Handling performance problems” on page 33 describes the use of the Unix tool "vmstat" to find out performance characteristics of your system.

- Chapter “Running the File -extract command” on page 37 describes important concepts and techniques for running the File -extract command.

- Chapter “Using CMVC with AFS” on page 43 describes important details on using AFS file systems to extract releases and levels from CMVC.

- Chapter “Release/Level/File -extract error messages” on page 51 describes the main error messages commonly seen during Release/Level/File -extract, and suggestions on what to do.

- Appendix A, “Description of the tools and how to get them” on page 59 provides a set of Korn and REXX samples that use the File -extract command to perform a massive brute-force extraction of single files, in case that there are problems with the release or level extract.

ABOUT THIS AND PREVIOUS TECHNICAL REPORTS

This document, as well as other recently released technical reports related to CMVC, contains some information that was previously available in other "unofficial" documents. We have endeavored to better organize the information and improve its readability so that you will find it more useful.

The focus of this technical report is CMVC 2.3.1.x which is the Year 2000 ready version of CMVC because it uses 4 digits to represent the year information. In contrast, CMVC 2.3.0.x and previous releases used only 2 digits to represent the year.

This technical report supersedes the technical report TR 29.2323 **CMVC frequently asked questions: handling file, release and level extract problems** and it replaces the CMVC readme file **/usr/lpp/cmvc/doc/README.fix.release.extract.problems**. Please use the new technical report, as we have made many improvements and will be removing the older documents from our shipped media and ftp sites.

For information on everyday usage of file, release and level extracts, see the companion technical report 29.2296 **CMVC frequently asked questions: files, releases, levels and tracks**.

ABOUT CMVC AND VISUALAGE TEAMCONNECTION

VisualAge TeamConnection is the successor product to CMVC. It has greatly expanded capabilities and is built upon an object model. While this and related technical reports discuss CMVC issues, we will periodically refer to new solutions provided in VisualAge TeamConnection.

Where a CMVC technical report covers a topic that might be of importance to VisualAge TeamConnection users and if the VisualAge TeamConnection documentation does not already cover the topic, a VisualAge TeamConnection technical report is or will be available.

ACKNOWLEDGEMENTS

Many of the questions and answers that are compiled in this technical report were obtained from the CMVC forum in the IBMPC conferencing disk and from the CMVC6000 forum in the IBMUNIX conferencing disk.

We want to thank the main participants in these electronic forums for their support, specially:

- Chris Cawthorne, IBM Hursley, United Kingdom.
- John Speer, CMVC Support in IBM Austin, Texas, USA.
- Gary Warner, CMVC Support in IBM Austin, Texas, USA.

We also want to thank the following CMVC customers who have provided additional feedback and suggestions:

- George Brandman, from Morgan Stanley, New York City.
- Tom Bing from Andersen Consulting and BellSouth, Atlanta, Georgia.
- Edwin Fernandez from LHS, Atlanta, Georgia.

Also, we want to thank Dodde Stark for editing this technical report.

OVERVIEW OF POSSIBLE PROBLEMS

This chapter provides an overview of some possible problems which can be quickly diagnosed and answered.

LIMITATION ON RESOURCES

Sometimes, a limitation of system resources may cause an extraction to fail. Check for the following:

- Disk constraints; for example, using up all space in /tmp or in the target file system.
- Memory constraints; for example, extracting the current release which contains many files, because the query that is used will exploit all available memory.
- Unreliable network; for example, packet timeout or the network router is dropping packets.

USAGE OF CMVC EXTRACTS THAT IMPACT OTHER USERS

The extraction of current versions may impact the performance of other CMVC daemons:

- Extracting *-current* instead of committed for large releases; this option issues a query that uses all available memory, thus, slowing down the server.

  You can minimize the impact by extracting a committed level, because a "map" file is used instead and it uses less resources. For more information about the map file see “How to use the map file for a committed level” on page 24.

In turn, activities that are being performed by other CMVC daemons may impact the performance of the current extraction:

- Not enough daemons running; therefore, other users have to wait to perform CMVC actions.
- Overly generic queries such as "1=1" for Files: sometimes novice users learn that "1=1" is a good way to obtain all the instances for an object, such as all users and all components.

  However, this query is NOT a good one to obtain information about the files because it will retrieve all the files for all releases. We have found that such query does not really provide useful information and it takes a very long time to perform, slowing down the other CMVC daemons.

VERIFY THE FILE PERMISSIONS AND OWNERSHIP OF CMVCD

A system administrator saw the cmvcd owned by root with setuid bit on and worried about a security impact, so she changed the file to owner "bin", group "bin", and removed the setuid bit. Afterwards, the following error message was returned on each release extract (in the syslog)

```
   0010-063 with chown() ...
```

Do not change the file permissions or the ownership of any of the CMVC executables.

The cmvcd process changes the owner to the id of the user issuing commands like Release -extract so that the proper ownership can be applied to files. Without the root ownership and the setuid bit on, this change of owners fails.

Use the following commands to find out the file permissions for cmvcd (in HP-UX use 'whence' instead of 'which'):

```
$ which cmvcd
# cd to the appropriate directory, such as the default one
$ cd /usr/lpp/cmvc/bin
$ ls -l cmvcd
```

The CMVC server daemon, cmvcd, should have the following settings in AIX (notice the s-bit and the ownership by root:system):

```
-rwsr-xr-x   1 root     system      /usr/lpp/cmvc/bin/cmvcd
```

;p. In case the s-bit is not set, as user 'root' do the following to set it:

```
$ chmod u+s /usr/lpp/cmvc/bin/cmvcd
or
$ chmod 4755 /usr/lpp/cmvc/bin/cmvcd
```

In case the file ownership is not correct, as user 'root' do the following:

```
chown root:system /usr/lpp/cmvc/bin/cmvcd
```

All other CMVC executables should be:

```
-rwxr-xr-x   1 bin      bin        /usr/lpp/cmvc/bin/Report*
```

MISCELLANEOUS SYSTEM RELATED PROBLEMS

The previous sections showed some main problems that may occur frequently. This section describes less common problems:

- Using a Netware file server (older versions of CMVC cannot change the name of read-only files).

- Writing to AFS/DFS space after the token has expired, or when the Access Control Lists are not properly set.
- Mounting NFS file system as read-only.
- The NFS daemon in the client workstation is not reliable. See "What to do when everything else fails?" on page 23.
- The client workstation does not have an NFS daemon (such as OS/2, Windows 95 and Windows NT). See "How to use File -extract as a brute-force approach" on page 23.

Specifying shared memory instead of TCP/IP in sqlhosts (Informix 7)

Question:

A level extract hangs, even though all the files have been extracted successfully.

Answer:

Specify TCP/IP and not shared memory for sqlhosts in Informix 7.

To avoid problems with unreleased shared-memory when performing some CMVC tasks, such as release and level extract, it is critical to NOT use the shared-memory connection for the Informix 7 databases and to use the TCP/IP connection. For example the file /usr/informix/etc/sqlhosts below shows one incorrect entry (which is commented out) and one correct entry:

```
# oem_hp03 onipcshm       oem-hp03        online3
oem_hp03 onsoctcp       oem-hp03        online3
```

Note: The value in the first column is the value to use with the environment variable INFORMIXSERVER which must be specified in the .profile of the CMVC family administrator.

ANALYSIS AND DIAGNOSTICS OF THE PROBLEM

This chapter provides some suggestions on how to perform the analysis of the extraction problem and how to better diagnose it.

WHAT ERROR MESSAGE IS SHOWN?

Most error messages are explained in the *CMVC System Administration and Installation* Version 2.3 manual.

However, in some cases the error may not appear in the manual or the explanation might not be sufficient. Therefore, "Release/Level/File -extract error messages" on page 51 provides an extended explanation for most of the error messages related to the extract function.

IS THE PROBLEM REPEATABLE OR INTERMITTENT?

- A repeatable problem indicates a possible file problem (for example, a corrupted SCCS archive), or constraints of memory and/or disk. See "Handling performance problems" on page 33 for details.
- Intermittent problems frequently indicate network problems.

WHAT VERSION OF THE CMVCD DAEMON IS RUNNING?

To find out, issue one of the following command from the family account:

- Option 1:

  ```
  what 'whence cmvcd' | grep cmvc
  ```

  Note: Where ' is the accent grave, in the upper left corner in the English keyboard.

- Option 2 (only when using Korn shell):

  ```
  what $(whence cmvcd) | grep cmvc
  ```

Use this command to identify the release and version of any CMVC executable program. This is most often used for the server daemon: cmvcd.

We have improved the handling of release and level extract in our latest refreshes of the code.

WHAT IS THE TARGET MACHINE (THE ONE RUNNING THE NFS SERVER DAEMON)?

- Is it a Unix system or something slower (such as PC)?

  If using a PC, then you may need to use the new options explained in “How to slow down a release/level extract” on page 18.
- How is the file system exported?

  Ensure that the CMVC server host is authorized to mount the exported file system.
- Is the file system read-write?

  Ensure that the CMVC server host can read and write into the exported file system.
- Is the CMVC server machine authorized to perform "root" operations?

  The CMVC server host needs to have this type of access in order to write files using another user id.
- What version of NFS is being used?

  NFS Version 3 uses TCP, whereas NFS Version 2 uses UDP. In order to use the TCP connection, the parties involved need to use NFS Version 3.

IS THE SYSLOG DAEMON RUNNING?

It is highly recommended that the syslog facility be enabled and configured properly because the syslog is the place where the error messages from CMVC server daemon, the database and the operating system will be stored.

To enable and configure the syslog facility see the technical report "CMVC frequently asked questions: handling OEM platforms and databases", section "Where are the error messages from the CMVC daemon and databases?".

If the syslog facility is enabled, what are the error messages in the syslog file? You can issue the following commands to find out:

1. Determine where is the syslog located by looking at the file /etc/syslog.conf and find out the complete path name for the file that handles warnings or errors. Find the entry for warning or error, such as:

   ```
   *.warning  /var/spool/syslog
   ```

   Let's assume that it is the default location for the syslog in AIX: /var/spool/syslog

2. Then look at the bottom of the syslog for possible related entries:

   ```
   tail -10 /var/spool/syslog
   ```

   Where -10 (the default, actually) is the number of lines to show from the bottom of the file.

If using AIX, is there any corresponding errors logged in `errpt`?

ARE THERE ERRORS REPORTED BY THE NFS SERVER DAEMON ON THE CLIENT?

In OS/2, the NFS daemon runs in its own window and any errors are displayed there.

In other operating systems, the NFS daemons usually report problems via the syslog (for example, NFSD losing packets).

DOES IT ALWAYS FAIL ON THE SAME FILE?

If the extract always fails on the same file, is it possible to extract the file using `File -extract`?

If the `File -extract` fails, it may be due to:

- Lack of space in /tmp or the file system of the family account.

  If the file is binary, the user needs roughly 3 times the size of the file in /tmp.

  If the file is text, the user needs roughly 2 times the size of the file in /tmp.

- A corrupted SCCS archive. Check the archive with:

  ```
  login cmvcFamily
  cd $HOME
  admin -h vc/../../../../s... (as specified in error message).
  ```

  For more details, see the technical report: **CMVC frequently asked questions: version control and database synchronization issues**.

IS THERE A ROUTER HOP INVOLVED?

Routers can fill up and consequently they can drop packets. Use a network load monitor or the error detection tools for the router to make sure that there is no data loss.

Try extracting locally (that is, use `-node nomount`) to see if the problem goes away. For more information on this feature, please see “Information about using NOMOUNT” on page 29.

IS THE NFS DAEMON SERVER A SLOW ONE?

A slow NFS daemon (for example, the current IBM OS/2 TCP/IP daemon) is slower than the ones on a typical Unix system, and thus, it might be necessary to slow down the NFS client at the Unix system in order to give the slow NFS daemon time to catch up. For more information on this feature, please see “How to slow down a release/level extract” on page 18.

ARE THE USERS STARTING/KILLING RELEASE -EXTRACT COMMANDS?

In some cases, there are users who start a `Release -extract` and then due to the slow performance, the users assume that the process is hung and then kill the extract process in the CMVC client.

When the user kills the extract on the client side, this action does NOT kill the corresponding process on the server, which will continue running as if nothing had happened and will terminate when it finishes the request.

Therefore, it is possible that these users may repeat this process of starting and killing multiple release extracts, eventually using up all the daemons on the server. As a result, no one else will be able to do any work with CMVC.

Ask the system administrator to run the CMVC command `monitor` to see how many daemons are busy running extracts and obtain their process id (psId):

1. It is possible to use `kill -1 psId` to recycle those daemons that are doing an extract for which the user has canceled the corresponding client process.
2. If the signal -1 does not recycle the daemon, try to raise the signal to `kill -15 psId` to gracefully kill that particular daemon.
3. If that does not work, then try the signal -9 to force the kill.
4. if that does not work, then stop the complete family and restart it.
5. if that does not work, then shutdown and restart the host.

IS IT POSSIBLE THAT MOUNT COMMAND IS NOT IN THE RIGHT LOCATION?

In some cases, like AIX 4, it is necessary to do extra configuration for the `mount` command in order to be found by CMVC.

Note: In AIX 4, the compatibility package (bos.compat) must be installed in order to add a symbolic link that makes mount accessible as /etc/mount because the real location is now /usr/sbin/mount.

IS THE DATE FORMAT CORRECT IN THE FAMILY DATABASE?

The newer release of CMVC, 2.3.1 or later, uses a new date string format (4-digit year, instead of 2-digit year format).

The family database will have to be converted to use 4-digit year format, if you migrated from an old version CMVC to CMVC 2.3.1 or later. Otherwise, the extract will give different results, because the string comparisons for the date (such as, lastUpdate date, "97/05/01" and "1997/05/01") are different.

Note: A tool, dbSetDate has been provided in CMVC 2.3.1 to convert the old CMVC family databases to the new date format that uses 4-digit year format for all dates.

IS CRLF CONVERSION NECESSARY?

Prior to 2.3.0.22, only the OS/2 and Windows CMVC client had support for the -crlf flag with Release/Level -extract to request the CRLF conversion. Starting with CMVC 2.3.0.22 (that includes 2.3.1.x), the -crlf flag is now available from the Unix CMVC line commands Release/Level.

If the files extracted have some strange characters at the end of the line such as ^M and/or at the end of the file such as ^Z, this indicates that CRLF (Carriage Return, Line Feed) conversion is needed. You can use the tool **fixcrlf** to perform this conversion. For more information see “CR (carriage return) and LF (line feed)” on page 65.

In general, it is necessary to request CRLF conversion (by specifying -crlf at the Release/Level -extract command line) when you want to extract release/level from an OS2 or Windows platform to a Unix platform.

Notes:

1. The release or level -extract command with -crlf option will be slower because the conversion has to be performed on each file in the release/level.
2. The corresponding change in the CMVC GUI for Unix has NOT been made yet; that is, you cannot specify the -crlf flag from the CMVC GUI for Unix.

WRAPPING THE MOUNT COMMAND

In some instances, it might be necessary to deal directly with the `mount` command to find out more details on the problem. It is possible to create a command that will be a wrapper to the real mount command for debugging or other purposes:

1. Locate the directory where the mount command resides, and change to it.
2. Rename the mount command (to be restored later).
3. Create a shell script in place of `mount`.

   You can take a look at the sample mentioned in the appendix “Sample mount script” on page 59.
4. Call the renamed `mount` from the new mount script.
5. The script must have the same permissions as the previous `mount` command
6. This is not a long term solution because the error handling is not always correctly handled in a script.

COPYING A FAMILY TO RECREATE A PROBLEM

In some extreme cases, it might be necessary to send a CMVC family to the CMVC support team for debugging.

Refer to the technical report TR 29.2232 "How to do migration tasks with CMVC" and to the *CMVC Server Administration and Installation, Version 2 Release 3* manual for details on how to export a database (see Appendix C).

ADJUSTING FOR INTERMITTENT FAILURES

We have learned from painful experience that even well managed networks can be intermittently unreliable due to NFS. As a result, we have provided several means of adapting the behavior of CMVC. This chapter describes the use of NFS, some common problems associated with this usage and some alternatives on how to overcome these problems.

BY DEFAULT, CMVC USES NFS TO MOUNT A FILE SYSTEM

When extracting a single file from CMVC, a socket connection is made between the client and the server. The file contents is transferred through that socket. When the file has been extracted, the socket is closed.

This is fine for one file or a small set of files. However, the overhead associated with opening and closing a socket for every extracted file can add up for a large number of files. Thus, for extracting releases or levels, sockets are not used.

Instead of sockets, Network File System (NFS) is used. The client exports the desired directory and the CMVC server mounts it. This remote directory is seen as local by the CMVC server. Files are extracted into this mounted directory without the need of a socket between two systems.

The NFS daemons that handled the exported local files are more robust on Unix than on OS/2 or Windows. These non-Unix NFS daemons have caused significant problems for extracting files:

1. PC's run slower than their Unix counterparts. Even when the operating system is not slower, limited multiprocessing capabilities (that is, running lots of processes through time sharing), device driver performance as well as smaller device driver buffers impact the overall performance.
2. Extract actions on slower machines cause buffers to fill up and overflow in routers and bridges, resulting in some packets being dropped (lost).
3. After packets are lost, the UDP protocol used by NFS has trouble recovering gracefully.

Other issues when using NFS with OS/2 and Windows clients

When older NFS software uses the UDP protocol instead of TCP, there is no guarantee that a data packet will reach intact the intended destination. Since NFS simply does not work as well on PCs, this means that extracts to PCs fail more often. Here are the most common failures:

1. A request to create a file on the PC is received by the PC and the file is opened, but the packet containing the file handle is lost by the router or by the PC.

   After timing out, NFS resends the request to open the new file, however, on the second try the file already exists and the PC returns an error message that implies that the file system is full.

2. When the CR/LF conversion occurs, there is so much Input/Output associated with parsing and rewriting a file (one line at a time), that packets are lost and the file cannot be written. The problem is identified as a failure of CR/LF conversion.

CHANGING YOUR SYSTEM CONFIGURATION

It is sometimes possible to overcome NFS problems with the changes described below.

Information about the NFSD and BIOD daemons

NFSD is the daemon that handles the requests from the client related to the file system. The BIOD daemon runs on all NFS client systems. When a user on a client wants to read or write to a file on a server, the BIOD daemon sends the request to the server.

Depending on the traffic load on your system, the number of daemons (NFSD and BIOD) can be changed. When the extract is very slow, you may want to increase the number of daemons (See your System Administrator for help).

CMVC_MOUNT_UNIX_PARMS and CMVC_MOUNT_OS2_PARMS

It is possible to override the default NFS mount parameters for Release/Level -extract. Many of these variable can influence the reliability and performance of NFS during an extract.

The CMVC_MOUNT_UNIX_PARMS and CMVC_MOUNT_OS2_PARMS variables (introduced in 2.3.0.24 and thus available in 2.3.1.x) override the default NFS mount parameters for Release/Level -extract. By default, they are not set.

Below are some statements that can be used to specify these new variables:

```
# Change mount options when Unix clients mount file systems
export CMVC_MOUNT_Unix_PARMS="-o hard,intr,rw"

# Change mount options when OS/2 or Windows clients mount file systems
export CMVC_MOUNT_OS2_PARMS="-o hard,intr,rw,retry=4,timeo=14,\
retrans=6,rsize=4096,wsize=4096"
```

Here are the default values used by the CMVC servers; please note that the actual value occupies one line.

HP/UX

.

CMVC_MOUNT_Unix_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6"
CMVC_MOUNT_OS2_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6,rsize=4096,wsize=4096"

SunOS:

CMVC_MOUNT_Unix_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6"
CMVC_MOUNT_OS2_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6,rsize =4096,wsize=4096"

Solaris:

.

CMVC_MOUNT_Unix_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6"
CMVC_MOUNT_OS2_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6,rsize =4096,wsize=4096"

AIX 4:

.

CMVC_MOUNT_Unix_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6"
CMVC_MOUNT_OS2_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6,rsize =4096,wsize=4096"

AIX 3:

.

CMVC_MOUNT_Unix_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6"
CMVC_MOUNT_OS2_PARMS "-o hard,intr,rw,retry=4,timeo=14,retrans=6,rsize =4096,wsize=4096"

Finally, since the above strings are compiled into cmvcd, you can see them with a command like:

```
strings cmvcd | grep "hard,intr"
```

HOW TO SLOW DOWN A RELEASE/LEVEL EXTRACT

Question:

What is the new function to slow down a release/level extract?

Answer:

The extract function for Release and Level has been expanded to include an option to slow down Release/Level extracts for less reliable networks and networks that include slow clients (such as OS/2 and Windows workstations with slow hard disks and/or slow processors). This function is useful in cases when packets are lost by the NFS server daemon on workstations, bridges or routers.

The CMVC_EXTRACT_RELIABILITY environment variable

The CMVC_EXTRACT_RELIABILITY environment variable can be set to YES (yes) to indicate that you want the reliable way of doing Release/Level extracts.

By setting CMVC_EXTRACT_RELIABILITY to NO (no), the behavior is changed to default behavior and the following CMVC environment variables will be ignored.

The CMVC_EXTRACT_SLEEP environment variable

The CMVC_EXTRACT_SLEEP environment variable is ONLY available at the CMVC server when using CMVC_EXTRACT_RELIABILITY=YES. It is set to a numeric value from 0 to 2000 (in units of milliseconds).

It is important to notice that the delay specified by CMVC_EXTRACT_SLEEP is applied at multiple points during a Release/Level -extract.

For example, setting CMVC_EXTRACT_SLEEP=100 (or 1/10th of a second) will result in many 1/10th second delays for each file extracted. While this can add minutes to the total time required for a Release/Level -extract, it also reduces the likelihood of network problems.

The CMVC_EXTRACT_SLEEP_FS environment variable

The CMVC_EXTRACT_SLEEP_FS environment variable is ONLY available at the CMVC server when using CMVC_EXTRACT_RELIABILITY=YES.

Often, network problems on Release/Level -extract only occur on a select group of client workstations (such as workstations running the CMVC client on Windows). For this reason the CMVC_EXTRACT_SLEEP_FS environment variable can be set to the following values:

- DOS

  This is the default. It is used for file systems designated by drive:\path.

- UNIX

  It is used for Unix type file systems.

- ALL

  It is used for all NFS mounted file systems.

If CMVC_EXTRACT_SLEEP is set, but CMVC_EXTRACT_SLEEP_FS is not set, the default value for CMVC_EXTRACT_SLEEP_FS=DOS.

The CMVC_EXTRACT_SLEEP does not apply in the following situations:

- CMVC_EXTRACT_SLEEP is ignored when "node=nomount".

  Because the NFS is not used, there is no need for the delay.

- If CMVC_EXTRACT_SLEEP=0 the delay is disabled.

EXTRACTING FASTER, WITHOUT NFS

With minor changes in your process, it is possible to avoid NFS entirely and greatly increase the speed and reliability of your extracts.

Using argument "-node NOMOUNT"

Release/Level -extract can specify the argument "-node NOMOUNT". This writes the files to the family server without using NFS, which dramatically reduces extract times.

However, you will need to transfer the files from the server to your client! Here are some methods to accomplish it:

1. Use an alternative network file system. You can perform the equivalent to an NFS mount by using Novell Netware, OS/2 Warp Server, AFS, DCE/DFS, etc. prior to issuing a Release/Level with "-extract -node NOMOUNT".

   This should improve the reliability of the extract and deliver the files to your client in one step.

2. You can extract to a file system on the server, then use a utility such as "tar" or "Info Zip" to bundle the files, send them from the server to the client in a single step, then unbundle the files on the client. For more information see “How to package and compress files” on page 63.

   Since CR/LF conversion then becomes an issue, CMVC provides the "fixcrlf" tool from the IBM software ftp site as well as internally to IBM through the CMVC Service/Development Home Page. This tool is included in CMVC 2.3.1.3 or later in /usr/lpp/cmvc/samples. For more information see “CR (carriage return) and LF (line feed)” on page 65.

Notes:

1. The uid/gid will be set to the client id on the server, or the family ID if the client id is not defined.

   This is the same as the default behavior for the mount option.

2. If the users attempt to set the uid or gid, they will receive a warning and the specified values will be ignored.

3. User must specify the attribute "-node nomount" or "-node NOMOUNT" in order to get this optional behavior.

4. The "-root directory" must be accessible to the userid of the client, that is, the uid/gid on the server must allow access.

For more details on this function, see “Information about using NOMOUNT” on page 29.

CMVC_EXTRACT_MOUNT_NAME

You have the option of moving your mount point out of /tmp. In cases where the file system is being used by many processes, distributing the file I/O across file systems can improve performance and reliability.

Starting with CMVC 2.3.0.22 (and thus, included in 2.3.1.x), it is possible to specify the temporary directory to be used by CMVC Release or Level extracts, by using the new CMVC environment variable:

```
export CMVC_EXTRACT_MOUNT_NAME=/tmp
```

The value of this variable is a path name for the desired temporary file directory. By default, if this variable is not specified or if the specified path name is not accessible, then /tmp is used.

REMOTE POSSIBILITY: MULTIPLE LAN ADAPTER CARDS IN YOUR SYSTEM

Thanks to Tom Bing for his following suggestion:

In case that everything looks fine for a release extract, but still NFS fails to write into the target system, one remote possibility is that the target host may have multiple LAN adapter cards and each one has a different IP address. Thus, the host might be known with one address but the actual NFS handshake might be done with the other address. In that case, then you need to add all these IP addresses as valid hosts that can write into the export file system.

Follow the procedure in “HP-UX: using the ping -o command to provide trace” to find out if this is the problem.

HP-UX: using the ping -o command to provide trace

The ping command in HP-UX provides the following option for displayed some tracing information:

```
$ ping -o oem-ppc1
PING oem-ppc1.raleigh.ibm.com: (9.67.163.37): 56 data bytes
64 bytes from 9.67.163.37: icmp_seq=0 ttl=254 time=70 ms
64 bytes from 9.67.163.37: icmp_seq=1 ttl=254 time=7 ms
Ctrl-C
--- oem-ppc1.raleigh.ibm.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
2 packets sent via:
   9.37.199.97   - oem-ppc1.raleigh.ibm.com
   9.37.193.53   - oem-hp10.raleigh.ibm.com
```

Notice the last 3 lines of the output. If the IP address shown in the last line is NOT the same as the main IP address of your box, then you have a 2nd LAN card that has its own IP address and thus, you need to add it to the list of hosts that can write into the file.

THE FINAL OPTION

In cases where NFS, or another network file system sharing mechanism will not work, and it is not acceptable for users to log directly onto the family server, it is possible to do a query to identify the files in a release and run a script to extract the files one at a time.

Since each extract indicates success or failure, it is possible to retry on each file and guarantee that the files are transferred. See “What to do when everything else fails?” on page 23 for more details.

WHAT TO DO WHEN EVERYTHING ELSE FAILS?

This chapter describes what to do when the troubleshooting techniques and workarounds described so far do not seem to work.

HOW TO USE FILE -EXTRACT AS A BRUTE-FORCE APPROACH

Question:

How to use File -extract as a brute-force approach.

Answer:

As a workaround to Release/Level -extract problems, it is possible to extract files from a release/level without using NFS. Instead, a brute-force approach can be used to perform a single File -extract for each file (which does not use NFS).

This approach can also be used in situations in which the client workstation does not have an NFS daemon (or a reliable one), such as OS/2, Windows 95 or Windows NT.

These tools are available in Korn shell scripts for Unix and in REXX for the Intel platforms.

The total extraction time will be longer than a pure Release/Level extract, due to the accumulation of the overhead of dealing with a repeated invocation of the CMVC File command.

The following 2 samples allow you to extract the equivalent of the latest committed level (the same as the committed release) and then extract the delta of the uncommitted level.

1. Extract all the committed files for a release. See “Extracting the committed versions of the files for a release.” on page 59.

   This is the equivalent of doing the full extract of the latest committed level for a release.

2. Extract all the file changes for an uncommitted level. See “Extracting all the files for an uncommitted level” on page 60.

   This is the equivalent of doing a delta extract of the latest uncommitted level for a release.

   This command is usually done after the committed versions of the files in a release have been extracted.

The above 2 files are all that you need, but for completeness sake we include the following supplemental samples:

- Extract all the file changes for a committed level. See “Extracting all the files for a committed level” on page 61.

  This is the equivalent of doing a delta extract of a committed level for a release.

- Extract all the current files for a release. See “Extracting the current versions of the files for a release.” on page 60.
- Extract the latest file changes for a committed track. See “Extract all the files associated with a committed track” on page 61.
- Extract all the files in a component for a specific release. See “Extracting all the files in a component for a single release” on page 61.
- Extract all the files in a component for all releases. See “Extracting all the files in a component for all releases” on page 62.

HOW TO USE THE MAP FILE FOR A COMMITTED LEVEL

Question:

How to use the map file for a committed level to extract files.

Answer:

There is no single easy query that can give you all the files that can be extracted when doing a "Level -extract -full". For this reason, the CMVC server stores this information in map files. There is one such map file for each committed level for each release. They are located in:

```
${FAMILY_HOME}/maps/${CMVC_RELEASE}/${LEVEL_NAME}
```

The format of the maps file is:

```
fileName  versionId fileId  changeType
```

Level -extract -full is a complicated process, but here are the basics:

1. The CMVC server reads the list of the files to extract for a level in a release from $HOME/maps/releaseName/levelName.
2. Based on the contents of the map file, the CMVC server queries the database to identify which files to extract.
3. The CMVC server performs the File -extract -version command to extract each file.

Note: If the file has been destroyed, you cannot extract it.

The file /usr/lpp/cmvc/samples/extractLevelUsingMapFile was added in CMVC 2.3.1.3, and it is shown below. You can experiment with this samples to extract a committed level by using directly its map file. You can to run this script from the CMVC family user id, in order to access the map files.

Warning: This is a simplistic example and does not cover all scenarios, such as destroyed files.

```
#!/usr/bin/ksh

#  The following could be made as input parameters
export RELEASE=r1
export LEVEL=l1

export mapsFile=$HOME/maps/$RELEASE/$LEVEL

while read line
do
   FILENAME= echo $line | cut -d' ' -f1      # field 1 is fileName
   VERSIONID= echo $line | cut -d' ' -f2     # field 2 is version id
   FILEID= echo $line | cut -d' ' -f3        # field 3 is file id

   Report -view ChangeView -where "fileId=$FILEID and \
     versionId=$VERSIONID and releaseName='$RELEASE' " -raw |
      awk -F"|" '{system("/usr/lpp/cmvc/bin/File -extract " $5 \
      " -relative /yourAFSDir -release " $1 " -version " $4)} '

   if [ $? -ne 0 ]
   then
      echo "This may have failed because it is a destroyed file"
      echo "Error found in Report -view ChangeView or File -extract"
      terminate
   fi
done < $mapsFile

exit

/* end of file */
```

Figure 1. Sample script to handle map files for a manual level extract

Notes:

1. The Report -view ChangeView command will return duplicate fileName/SID combinations if multiple tracks were specified upon checkin of a file. So, you may need to sort the output from Report -view ChangeView before using "awk". It is possible that the following will work but may not be too eloquent:

```
Report -view ChangeView -where "fileId=$FILEID and \
        versionId=$VERSIONID and release  Name='$RELEASE' " -raw |
        sed "s/\|/:/g" | sort -u -k4,5 -t: |
        awk -F":" ' {system("/usr/lpp/cmvc/bin/File -extract " $5 \
        " -relative /yourAFSDir -release " $1 " -version " $4)} '
```

2. For destroyed files, the File -extract will not work since the File entry in the database is destroyed. Therefore the only thing you could do is issue a native SCCS command to get the file from the CMVC server. This is NOT recommended because it is not a good idea to allow all users to issue native SCCS commands from the CMVC server.

VisualAge TeamConnection note: VisualAge TeamConnection has integrated the map file data into the database, thus allowing users direct access to the version information for each level (Driver in VisualAge TeamConnection).

HOW TO DIVIDE THE MAP FILE TO FIND OUT AN OFFENDING FILE

If there is an incorrect entry in the map file, then the following error will be issued:

```
0010-023 The extraction of level LEVEL1 associated with release
         RELEASE1 failed. The internal map for the level is not
         valid or the level map file is not loaded on the
         CMVC server disk.
```

If the map file exists and looks OK, then probably one entry in the map file is for a file that is not valid for the level. One way to find out the offending entry is to do the following that is a typical binary search:

1. Make a backup copy of the original map file.
2. Divide the map file by half. For example, if the total number of files in the original file is 8, then have a file with the first 4 and another with the other 4.
3. Do a level extract using the first map file. If it fails, then you know that the offending line is in that first map file.
4. Repeat the process and keep reducing the lines in the map file to be used until you find the offending file.
5. In one customer case, the offending file went thru a complex series of actions and it was linked to several releases and to several uncommitted levels. For some reason it was committed only in level 72 but it was shown in level 69 and it was causing problems during the extraction of 69.

HOW TO RECREATE LOST MAP FILES

Question:

Some map files were deleted by mistake from the CMVC family account. Can they be recreated manually?

Answer:

The map contains entries with the following format:

```
pathName    versionId   fileId     type
```

There are easy ways to reproduce a lost map file. One possibility that was proposed by the CMVC support group in Austin is (thanks to Gary Warner for his feedback!):

1. You could take the map file from the previously committed level.
2. Remove any lines with type='delete'.
3. Change the remaining types to "noDelta".
4. Get a list of the changes from the level with the lost map file.

   For each file, find the highest version (if multiple updates).
5. Now you need the version's id and the file's id. You probably have to query the data base directly.
6. If the file is not new in the missing map file, then it has the same id, so the query could be:

   ```
   select sourceid from Files where id=xxx
   ```
7. If the file is new, then the query could be:

   ```
   select id,sourceId from FileView where pathName='XXXXXXXX'
   ```

   This assumes that the file has not been renamed in the meantime.
8. This gets the sourceId which is needed to find the version's id:

   ```
   select id from Versions where sourceId=XXX and sid='X.Y'
   ```
9. If the file is new in this level, the type will be "create" or "link".

   If a file is deleted, the type is "delete".

   Otherwise, the type will be "delta".

   If the file was renamed, the type is "rename". This type takes precedence over other types (by the way, the field pathName in the map file is the new name).

   Any files unchanged by the level retain the "noDelta" type.

FOR ADVANCED USERS: CREATE A TABLE TO STORE THE CONTENTS OF MAP FILES

Starting with CMVC 2.3.1, we provide the new function "Report -general" in which the users can specify a given table and a given query. Then this command passes the request directly to the database and then the database process the request and provides the results thru CMVC. Thus, CMVC client users can access the database without being logged in the host where the CMVC family server is running.

This new function can be exploited by advanced users as follows: a new table can be added to the database of the CMVC family and a user exit on Level Commit (exit 3) will insert the contents of the resulting $HOME/map file for the committed level into this new table. Then, the users can exploit the CMVC Report -general on this new table to get the information on the map files.

INFORMATION ABOUT USING NOMOUNT

This chapter describes the use of the new 2.3.x function for performing a Release or Level extract to a local file system (in the CMVC family host), bypassing NFS.

CAN A RELEASE/LEVEL EXTRACT BYPASS NFS?

Question:

What is the new function to avoid using NFS on release/level extract?

Answer:

The extract function for Release and Level has been expanded to include an option to not perform an NFS mount to extract the files (that is, to extract to the CMVC server machine without performing an NFS mount).

In this option the extraction will always be to a local directory in the CMVC server host. Because if NFS is not used, then the CMVC server will not be able to access remote directories for the extraction.

The default function is to continue using NFS for mounting the target directories for the extraction.

The specification for the new no-mount capability can be done as follows:

- From the client, specify the keyword NOMOUNT as the destination host (for example, -node nomount).

  For the -root attribute, specify a directory that is not available via NFS mount. This option allows users to selectively indicate which extractions will not use an NFS mount.

- From the server, set the CMVC environment variable, CMVC_NFS_DISABLE=YES (the default is NO).

  This option allows family administrators to prevent the use of an NFS mount during extractions.

The details of the new behavior are shown in the following sections.

Details on the Mount option

- If a user specifies a uid (the numeric user identification in /etc/passwd) and/or gid (the numeric group identification in /etc/passwd and /etc/group), all extracted files and directories will receive that uid and/or gid.
- The defaults for uid and gid are:
  - If the client userid (the user name you use when logging on, such as testclt) exists on the server, the server will look up the uid and gid on the server and use them.
  - If the id of the user exists on both the client and the server machines, but have different uids and gids on those machines, the extract will be done, but the userid on the client will not own the files.

    For example:

    ```
    testclt on client_wsn has uid 113 and gid 14
    testclt on server_wsn has uid 110 and gid 10
    testfam on server_wsn has uid 115 and gid 15
    ```

    The extracted files will have uid 110 and gid 10 (from server)
  - If the client userid is NOT defined on the server, then the family uid and gid will be used (for example, uid 115, gid 15).
- File permissions filtering:

  The setuid bit will always be filtered. For example:

  ```
  rwsrw-rw- becomes rwxrw-rw-
  rwSrw-rw- becomes rw-rw-rw-
  ```

Details on the NOMOUNT option

- Uid/gid will be set to client id on the server, or family ID if the client id is not defined. This is the same as the default behavior for the mount option.
- If the user attempts to set the uid or gid, they will receive a warning, and the specified values will be ignored.
- User must specify a -node of "nomount" or "NOMOUNT" in order to get this optional behavior.
- The -root directory must be accessible to the userid of the client, that is, the uid/gid on the server must allow access.

Details on CMVC_NFS_DISABLE=YES option

- The family administrator must define and export the CMVC_NFS_DISABLE variable with the value either "YES" or "yes", in the family account. This must be done before the daemons are started.

  If this variable is not specified, then the default is NO.

- Once CMVC_NFS_DISABLE is set to yes, all users will then receive the NOMOUNT behavior:

  – If the user specifies any valid -node other than "NOMOUNT" or "nomount", they will receive a warning that the -node is being ignored and told that their files will be extracted to the server.

  – If the "become id" (set with the -become attribute) does not exist in the server's /etc/passwd file, the family account id will be used for the extracted files (that is, the family account will own the files).

    If the "become id" does exist, that user id will own the files.

  – If the user account has a different uid on the server (such as, 123 on client-a and 125 on server-b), the uid on the server will be used (that is, 125).

Implications of AFS/NIS/NIS+

- Distributed file systems allow the user's space to be accessible by the server in order to extract directly to the server. Uid and gid must be common on all file systems.
- Some configurations of NIS+ may cause problems with userid resolution.
- Automounting schemes may force users to do an extra step so that their directories will be visible on the server (such as klog in AFS).
- Automounting problems should be addressable through user exits.

HANDLING PERFORMANCE PROBLEMS

The first step to determine whether the problem is related to the performance of the Unix system is to use the `vmstat` tool to monitor system behavior. Some possible performance problems are:

- Running out of CPU cycles (becoming CPU bound)
- Running out of real memory or paging space (thrashing)

USING VMSTAT TO FIND PERFORMANCE PROBLEMS

The `vmstat` command can be used to determine both the CPU utilization and the real memory availability in a Unix system You generally run vmstat in one window, while running line commands in another window. You can also set this up as an `at` job, redirecting the output to a file. The goal is to make sure you catch the system behavior while the CMVC command and database operations are in progress.

vmstat usage

The vmstat command is invoked as follows:

```
vmstat frequency [repetition]
```

Notes:

1. frequency is how often the system will be checked. You generally use "1", for every second.
2. repetition is optional. It is the number of times to repeat the command. Without this parameter, vmstat will run until the process is killed.
3. Ignore the first line of output which is an average of the system performance from the first day the box was used.

Interpreting the vmstat output

This is what to look for when running vmstat. You can find more detailed information in the AIX commands reference or the *AIX 3.2 Performance Monitoring and Tuning Guide*:

Column
heading **Description and possible interpretation**

avm available memory.

If this number gets too low, processes will be swapped out. This is one of the most expensive operations in Unix. The relative cost of recovering memory is shown in paging activity and context switches.

fre free memory list.

This is an arbitrary value, but when the number of pages becomes too low (again, an arbitrary value), the "initd" process runs garbage collection. This will slow your system for a short time.

po page out.

These operations move pages containing dormant processes to paging space. Depending on how efficiently paging space is set up, this is either slow or very slow. Ideally, you want paging space on several of your least active disks (so that the least active disk can receive the pages). Also, to improve performance it helps to have the disks on separate SCSI controller cards (since a single SCSI card can send data to only one device at a time). Ideally, this number is always 0.

cs context switches.

Each context switch is a change of the active process. It includes changing from user state to system state when an application requests access to a privileged device (for example, an application makes a **system()** call to write the file). The more processes you run and the less efficiently these processes make use of system calls, the higher this number. Ideally, this number is at or below 100.

r process run queue.

This is the number of processes waiting for access to the CPU. Ideally this is never greater than 1 or 2. It is a gross indication that the system is not keeping up with the user's demands.

id cpu idle.

This the amount of time that the CPU has nothing to do. A healthy system has at least 20 percent idle. You would probably like about 50 percent idle so that you can handle the rough times.

wa cpu wait on device.

This number indicated the percentage of time the cpu is waiting for a device to complete a task. If this number is routinely high, you could have too much data on a single disk (so lots of writes have to be serialized) or have lots of delays on your network.

sy cpu system utilization.

A further indication of system problems is when "sy" is greater then "us" (cpu user process utilization). This is usually an indication that the system is not well configured or that the application was written inefficiently (for example, using unbuffered write commands).

Following leads provided by vmstat

If you see any unusual activity with vmstat, also look at the error report log with the `errpt` command. Some of the system problems are very visible here, for example:

- "paging space low, killing process ..."

  Paging space can be checked on AIX using `lsps -a`. There should be roughly twice as much paging space as real memory (information available from SMIT), and it should be distributed across disks.

- "token ring error ..."

  Check netstat to see if there are network errors. Also, ask the system administrator to check for the physical integrity of the network components (such as adapter cards, cables, etc.).

USING NETLS DEBUGGING TOOLS THAT USE CMVC LICENSES

In some occasions, if the NetLS configuration is not correct, the CMVC clients may fail to execute properly because they cannot reach the NetLS server, or they cannot obtain a valid license, or the performance of the execution of the CMVC command may seem to be too slow.

To help with the diagnostics and the debugging of these NetLS configuration problems, the following executable files are provided which can be used to contact the NetLS server and request a token for CMVC. Each program uses the same logic as a normal CMVC line command, but it will display a lot of information that will allow the user to better diagnose the problem with the NetLS configuration.

Each program does not use the message catalog from CMVC, and thus, it is an independent program. Actually, each program does not require that CMVC should be installed at all.

The testing NetLS programs in /usr/lpp/cmvc/bin (Client code) are:

1. test.netls.with.queues

   This program uses the NetLS processing done by CMVC in 2.2, 2.3.0.0 up to 2.3.0.20.

2. test.netls.no.queues

   This program uses the NetLS processing done by CMVC since 2.3.0.21, in which the previous waiting approach for NetLS was replaced by "softstop". That is, if all the licenses for CMVC are taken, the next user to request a license is not placed in a queue, instead, a token is given and an entry is added into the NetLS server log to indicate this situation.

These tools provide time stamps that can help you to determine the overhead associated with getting a license from NetLS. If the overhead is large (more than 1 or 2 seconds), you may have do the procedure mentioned in the technical report "CMVC frequently asked questions: license management with NetlS/iFOR" to optimize the NetLS configuration.

RUNNING THE FILE -EXTRACT COMMAND

This chapter describes important concepts and techniques for running the File -extract command.

SPECIFYING THE FULL PATH NAME OF A CMVC FILE USING CMVC_TOP

Question:

Specifying the full path name of a CMVC file using CMVC_TOP

Answer:

The material in this section is based on Chapter 18, Part, "VisualAge TeamConnection Commands Reference".

There are a number of ways to specify the full path of a CMVC file:

- Specify the file name (which includes the path name and base name) and take the path from your current working directory.
- Use the `-relative` flag to specify a path name that is outside your current working directory.

  In this case, the values for the `-relative` flag and the file name are concatenated.
- If your workstation directory structure matches the way files are named, you can set the CMVC_TOP variable to the leading portion of your current working directory and specify only the base name in the actions of the `File` command.

  CMVC subtracts the contents of CMVC_TOP from the current working directory.

  This option might take some time to understand, but it can save you a great deal of typing when using line commands.

The example that follows starts with the simplest case and moves to the more complex. Throughout the example, assume that the environment variables for CMVC_FAMILY, CMVC_RELEASE and CMVC_BECOME have been set.

Working from your current working directory

The main assumption for this example, is that the user is NOT using a directory/file structure that mimics the path names of the files stored in CMVC.

Initial checkout of txfile.c

A user is working with a file named txfile.c. The path name and base name for this file is src/lib/txfile.c; the current working directory on the client workstation is /robot. There is no other file in this release named txfile.c.

1. The current working directory is: /robot
2. The user has not set CMVC_TOP yet.
3. The user enters the command:

   ```
   File -checkout txfile.c
   ```

4. The family server identifies that the base name of txfile.c is unique in the release: src/lib/txfile.c
5. The file is checked out in the workstation as: /robot/txfile.c

Initial checkout of msg.h

Next, the user needs to check out src/lib/msg.h. Let's assume that other files named msg.h exist in the release, thus, the user must specify the entire path name.

1. The current working directory is: /robot
2. The user has not set CMVC_TOP yet.
3. The user enters the command:

   ```
   File -checkout src/lib/msg.h
   ```

4. The family server identifies that the base name of msg.h is not unique in the release.
5. The file is checked out in the workstation as: /robot/src/lib/msg.h

Miscellaneous notes

If you create a file without specifying `-relative` or setting the CMVC_TOP variable, then a file with the same name must exist in the current working directory. For example, you use the following command:

```
File -create src/lib/read.me
```

In this case, read.me must exist in your current directory; CMVC does not search for the part in /src/lib.

Using the -relative flag

Check out again the file txfile.c from another directory

The next day, the user's current working directory is /server. But the user wants to check out txfile.c to the /robot directory. The user specifies the `-relative` flag to do this:

1. The current working directory is: /server
2. The user has not set CMVC_TOP yet.
3. The user enters the command:

   ```
   File -checkout txfile.c -relative /robot
   ```
4. The family server identifies that the base name of txfile.c is unique in the release: src/lib/txfile.c
5. The file is checked out in the workstation as: /robot/src/lib/txfile.c

Check out again the file msg.h

Similar scenario, but this time with the file msg.h which is not unique in the release, and thus, it is necessary to specify the complete path name.

1. The current working directory is: /server
2. The user has not set CMVC_TOP yet.
3. The user enters the command:

   ```
   File -checkout src/lib/msg.h -relative /robot
   ```
4. The family server identifies that the base name of msg.h is not unique in the release.
5. The file is checked out in the workstation as: /robot/src/lib/msg.h

Using the CMVC_TOP variable

A few days later, the user has checked out many files from the release into subdirectories of /robot, using a directory structure that mimics the names of CMVC files and for convenience, the user sets the CMVC_TOP variable to /robot.

Now when working with files, the user can specify just the base name of the files in his current working directory, as in this example:

1. The current working directory is: /robot/src/lib
2. The user has set CMVC_TOP to /robot
3. The user enters the command:
   ```
   File -checkout txfile.c
   ```
4. The family server identifies that the base name of txfile.c is unique in the release: src/lib/txfile.c
5. The file is checked out in the workstation as: /robot/src/lib/txfile.c

The additional information added to the leading portion of the path is formed by subtracting the value of CMVC_TOP from the current working directory. In other words, the CMVC_TOP variable tells CMVC which portion of the current working directory path belongs to the client workstation.

This command is completely equivalent to the following command if the CMVC_TOP variable is not set:

```
File -checkout src/lib/server/tx/txfile.c -relative /robot
```

DOES CMVC USE FORWARD OR BACKWARD SLASHES WHEN HANDLING THE FILE PATH NAME?

Question:

Does CMVC use forward or backward slashes when handling the file path name?

Answer:

The CMVC client and server will take into account the usage of the slashes, both backward (for DOS, OS/2 and Windows) and forward (for Unix). The file will be treated according to the target platform. That is, you can extract the file tmp\file.dos into a Unix directory and the end result will be a file in tmp/file.dos. The reverse is also true, you can extract the file tmp/file.aix into a DOS directory and the end result will be a file in tmp\file.aix.

You can use the GUI to specify the relative directory to store the code. In the Unix GUI it is called "Set Directory", in the OS/2 and Windows GUI is called "Relative directory" on the first settings page.

When using the line commands the CMVC_TOP environment variable can be used. For more details on this topic, please see page 51 in the CMVC Commands Reference, Chapter 10, Files. Actually, we improved the explanation in the "VisualAge TeamConnection Commands Reference", page 98, Chapter 18, Parts (same concept).

An example of the relative directory is shown next: if the user specifies the relative directory to C:\pages, and the file tmp\file.dos is extracted, then the file will be located in C:\pages\tmp\file.dos.

Another point to consider is the difference between "path name" and "base name" when doing queries. The base name is the file name with NO directory information, such as file.dos, while the path name is the complete file name with the directory information, such as tmp/file.dos.

Note: The CMVC server normalizes the path names inside the CMVC database with the forward slash '/' and not with the backward slash '\'. The consequence is that in the Filter dialogs, you must specify the path names with the forward slash; otherwise, the result of the query will be empty.

UNIX: FILE COMMAND ADDS (DOT) FILES AT EXTRACT OR CHECK-OUT

Question:

Unix: File command adds (dot) files at extract or check-out

Answer:

To preserve any potential changes that users could have made to the files on their workstation, the CMVC File command creates a copy of a file being extracted or checked out, and this copy is renamed with a prefix of . (dot) in order to keep it hidden during a normal "ls" query. In order to see these dot files, you have to issue the command "ls -a".

INTEL: FILE COMMAND ADDS $ TO THE FILE SUFFIX AT EXTRACT OR CHECK-OUT

Question:

File command adds $ to the file suffix at extract or check-out when using the client for OS/2 or for Windows.

Answer:

To preserve any potential changes that users could have made to the files on their workstation, the CMVC File command creates a copy of a file being extracted or checked out, and this copy is renamed with a suffix that begins with a $ (dollar).

RESTRICTION ON FILE EXTRACTS INTO DOS OR OS/2 FAT DRIVES

Question:

Restriction on file extracts into DOS or OS/2 FAT drives

Answer:

Files from the CMVC server that do not conform to the 8.3 naming convention (file name longer than 8 characters or file extensions longer than 3 characters) cannot be extracted to a FAT drive.

Note: Because Windows 95 or Windows NT have a way of handling long file names even in FAT drives, you can use the CMVC client for Windows 32-bit to extract long file names into these operating systems.

USING CMVC WITH AFS

This chapter describes important details on using AFS file systems to extract releases and levels from CMVC.

WHAT ARE THE GENERAL GUIDELINES FOR USING CMVC WITH AFS?

Question:

What are the general guidelines for using CMVC with AFS?

Answer:

CMVC supports use of AFS space for user's files. Here are some suggestions and limitations.

- For File -checkin, -checkout, -extract, etc., it is necessary to have an active AFS token. CMVC will only write into AFS space that the user is currently authorized to write.
- If a token expires it may be necessary to restart the CMVC GUI after performing a "klog".
- Release -extract and Level -extract benefit from a new feature in CMVC 2.3: nomount.
  - Using "nomount", the user extracts into their own AFS space on the CMVC server machine without performing an NFS mount.
  - The UID and GID of the extracted files are set to the ID of the user on the server, therefore it is important for each user to have the same UID and GID on all machines in the AFS cell when using "nomount".
  - Since the extraction is occurring on the server machine, it is necessary to have an active AFS token on that machine in order for the extract to work. This can be accomplished with a user exit that issues the "klog" command.
  - In lieu of issuing a "klog" command, the directory where files are to be extracted can have permissions set to "system:anyuser all". While this is somewhat of a security exposure, it provides for greater flexibility.
  - It is possible to restrict all users to use only the "nomount" mode for extracts, by using the following environment variable in the .profile in the family account:

    ```
    export CMVC_NFS_DISABLE=YES
    ```

  - It is important to specify the target directory correctly, especially if this command is run this from an OS/2 client. The name of the target directory has to be the name that the

CMVC server knows and the slashes have to be forward-slashes (as they would appear if they had been entered on an AIX box).

- The CMVC family account should not be placed in AFS space. There is an exposure to the reliability of CMVC if there is any chance of an AFS token expiring while the CMVC daemons are running.

  Also, some databases cannot run in AFS space since AFS does not support write verification.

- For additional details on using AFS with CMVC, see “How to setup a CMVC family to do release extracts into AFS space?.”
- For the details on how to setup a CMVC family to use AFS, see “What are some additional considerations when using CMVC with AFS?” on page 45.
- For the details on the usage of Release/Level -extracts with the -nomount option, see “Information about using NOMOUNT” on page 29.

HOW TO SETUP A CMVC FAMILY TO DO RELEASE EXTRACTS INTO AFS SPACE?

Question:

How to setup a CMVC family to do release extracts into AFS space?

Answer:

Thanks to Chris Cawthorne for his feedback in this topic.

Several sites that use CMVC perform level/release extracts to AFS without any problems.

You can only do this on versions of CMVC which allow you to specify NOMOUNT as the node for the extract; this function was added in CMVC 2.3.

The setup of the CMVC family is as follows:

- An AFS id is created with the same name and uid as the CMVC family.
- The AFS password is the same as the Unix login password and as the CMVC database password such as DB2_PASS or ORACLE_PASS.
- The AFS id is given proper access in the access control list (ACL), such as write-access to the directory that the code will be extracted into.
- In the .profile of the CMVC family userid, the following statement is added in order to perform klogs in to AFS:

```
/usr/afsws/bin/klog $CMVC_FAMILY -password $DB2_PASS
```

- Every night a backup job is run which performs the "su" command to the family userid.

  This job runs the .profile so the AFS token is refreshed at least once a day.

  Alternatively, you could add a user exit on the release and level extract commands to klog in to AFS. In theory, this is more reliable but it has not been necessary so far.

WHAT ARE SOME ADDITIONAL CONSIDERATIONS WHEN USING CMVC WITH AFS?

Question:

What are some additional considerations when using CMVC with AFS?

Answer:

Some additional considerations when using CMVC with AFS are shown below:

- Prior to the extract, use a user exit to issue the "klog" command to the CMVC server specifying the appropriate AFS user. In order for the CMVC server to do this, either:
  - The passwords for each account would need to be stored in the family account for use with klog, or
  - The CMVC server would need to be an AFS administrative user with the authority to klog to any other user in the group.

  We have not looked into the details of implementing the above options. However, this should work in any properly configured AFS environment and should allow all files to be written to the user's file system with the user as the owner, instead of 32766 (AFS's anyuser id).

- If the code from AIX /etc/login is used instead of the AFS /etc/login code, then it is possible for the CMVC server to "share" the AFS token with a user of the same ID on the same workstation. In other words, if user "joe" has done klog on the family server workstation and invokes Release -extract, then when the CMVC daemon (cmvcd) performs a setuid to become "joe", it would be using the same AFS token.

  Note: The telnet command uses something other than /etc/login, whereas the rlogin command uses /etc/login. Therefore, use rlogin to access the family account.

- If the family account and each user are in the same AFS authentication group, and if the group permissions are set correctly in the user's file system, then the files will be extracted and owned by the family account.

This is not as good as any of the options mentioned above, but it is easier. The following command addresses the inconsistency with the file ownership, which is that the files are owned by the CMVC family account and not by the desired user (represented by xxx below):

```
find . -exec chown xxx {} \;
```

The following are some miscellaneous notes from conversations with people from Transarc.

- If you do not use the AFS "login" command and a user issues the "klog" command on your server, when the cmvcd daemons later issues the function "setuid" to change the user id to that user, then both the user and the cmvcd daemon can "share" the token.

  This is considered a security breach by Transarc, but it might be a useful side effect when dealing with extractions.

- Make sure that all parent directories have the following access specification:

  ```
  system:anyuser rl
  ```

  The directory to extract into needs to have enough authority for cmvcd to extract file.

- Check if there are operating system issues (for example, AFS works OK with Solaris clients, but not with SunOS ones).

- Check the permissions of any associated directories, such as by using "ls -ld ." and "fs listacl .".

- Use the "tokens" command to see what tokens the user and the CMVC family have on the server machine.

- Verify that AFS and your name server are using the same ID for the user. You can use "id user" and "pts exam user".

- Verify which version of "login" are you using:

  ```
  strings  which login  | grep Base
  ```

  To determine this, look for AFS in /etc/security/user and /etc/security/login.cfg

- Verify if the version of "getty" that you are using is a symbolic link to "tsm":

  ```
  ls -l /etc/getty
  ```

- Verify if the cmvcd executable is in AFS space, and its file permissions:

  ```
   -rwsr-xr-x   1 root     system      /usr/lpp/cmvc/bin/cmvcd*
  ```

- Verify that the AFS cell is active:

  ```
  fs getcellstatus -cell CELLNAME
  ```

- As user "root" you can perform the following command to find out more about the users:

  ```
  kdump -users
  ```

- Manually create a file (without the AFS tokens) and check the file's ownership.

- Run "rxdebug <machine> 7001 -version" on a few machines.
- Perform tracing:

  ```
  fs trace
  ```

HOW TO SOLVE PROBLEMS WITH AUTHENTICATION WHEN USING CMVC WITH AFS?

Question:

How to solve problems with authentication when using CMVC with AFS?

Answer:

Thanks to George Brandman for his feedback in this topic.

The following information was provided by Mary Ann DelBusso from Transarc. It deals with authentication problems when using AFS with applications such as CMVC that perform a setuid to an AFS user and then write files into AFS space.

The scenario is that the files that are extracted do not have the proper file ownership and two users are involved "george" and "wds".

- While both users (such as "george" and "wds") have tokens on the server machine (such as "rs_aix32"), then you as user "root" should perform the following:

  ```
  kdump -users
  ```

- Then you as root should perform:

  ```
  groups
  ```

- Follow the instructions in “Running fstrace to diagnose some AFS problems” on page 48 and run "fstrace" 2 times: once on a successful run of CMVC for user "wds" and then again when CMVC fails for user "george". Both runs should be initiated from the same client machine. The tool "fstrace" should be run from the CMVC server where the files are actually written.

  An example of the diagnostic results from fstrace is shown in “Example of results from running fstrace” on page 49.

Running fstrace to diagnose some AFS problems

There is a program called "fstrace" (documented in the AFS 3.4 release notes) which may give some more information when encountering certain AFS authentication problems.

Copy "fstrace" to the local user file system (ufs) space. The tool fstrace is normally in /usr/afsws/etc and you can copy it into /usr/vice/etc.

Do not write the fstrace output logs to AFS space because you need to run it from outside AFS and it should be run at a time when no other users are accessing AFS on the machine.

Ensure that the /usr/vice/etc/C/afszcm.cat file exists and that it is the same version of the kernel extensions running on the client. This might be difficult to determine since the afszcm.cat file does not have a version string associated with it. If the file is out of date, you will see "raw" output for some of the lines in the fstraceLog file.

The file afszcm.cat contains formatting instructions for each type of fstrace log entry; when you run 'fstrace dump', it combines the log entries with the message strings in the file to form readable output.

Next, you need 2 sessions on a single AFS client machine in order to do the following (using csh):

1. In session 1, as user "root":

```
# Set the environment variables needed for using the afszcm.cat file
        setenv NLSPATH /usr/vice/etc/%L/%N
        setenv LANG C

# Do the initialization of the fstrace commands
        /usr/vice/etc/fstrace clear cm
        /usr/vice/etc/fstrace setlog cmfx -buffers 100

# Start the fstrace session and continuously dump the fstrace log
# to a file
        /usr/vice/etc/fstrace sets cm -active
        /usr/vice/etc/fstrace dump -follow cmfx -file /tmp/fstraceLog \
          -sleep 10
```

2. In session 2:

```
# After starting the fstrace, you can run CMVC.
# When finished, wait a few minutes, then continue in the
# root window...

# The "fstrace dump -follow" command will not return, you need to do a
# CONTROL-C to kill it.  You can also place this command in the
# background.  Just make sure that you kill this when you want to stop
# the fstrace logging.
```

```
   # You can run the following command to make the fstrace logging
   # inactive.
           /usr/vice/etc/fstrace sets cm -inactive
```

If you are interested in reading more about the fstrace facility, you can check out the article that introduced it in the Cache Update newsletter from Transarc. It was in the January 1994 edition which you can access at /afs/transarc.com/public/afsug/newsletter/jan94

Example of results from running fstrace

The output from the "groups" command is shown below and it gives a clue as to why this works for user id "wds" but not for user "george":

```
Groups for george on sax3
-------------------------
33536 32533 cc7639 ccm_root

Groups for wds on sax
---------------------------
company aplus mstk floppy idt ccm_root
```

The user "george" is a member of 2 groups "33536" and "32533" while user "wds" is not a member of these groups. The two numbered groups mean that the AFS tokens for "george" are associated with a PAG. Unlike "wds", it means that the tokens for "george" are NOT shared with other processes under the same UID. For CMVC to work, the UID-based tokens are required.

There are two ways that you could have gotten the PAG: either you (the user id "george") is running the AFS version of login or is specifically executing the "pagsh" command prior to running "klog". The userid "george" needs to check .login, .cshrc, .profile or whatever is appropriate. If the userid "george" does not see the pagsh being run, then somehow, the user is getting the AFS version of login while the user "wds" is not.

In this particular case, the userid "george" was using telnet and the other user was using rlogin. Telnet was associated with a PAG while rlogin was not.

RELEASE/LEVEL/FILE -EXTRACT ERROR MESSAGES

The following are error messages commonly seen during Release/Level/File -extract, and suggestions on what to do.

MESSAGES RELATED TO FILE SYSTEM SPACE AND FILE PERMISSIONS

The following two error messages indicate that the file system runs out of space or does not have the proper file permissions.

- msgFileBadVCPut

  ```
  0010-045 An error occurred when the CMVC server software tried to
           check in file %1$s to the
           version control file repository.

           This error also occurs when a text file has some kind of
           data corruption, such as an incomplete last line without EOF.
           Verify that the file is complete, such as by using vi and then
           saving the file to ensure that the EOF marker is added.
           Also, you may try to create the file as binary instead of text.
  ```

  Another possibility besides the ones mentioned in the 2nd paragraph of the above message is that the free space in /tmp is not large enough to accommodate the actual file plus the temporary copy. For more details see on page 53.

  A remote possibility is that the customer is testing CMVC and has created a CMVC family with "mkdb" or "mkdb -d", created some files, and then deleted the family database by using "rmdb", but the user forgot to issue "rmfamily" to eliminate the vc tree. Then, the user tried to create again the family with "mkdb" or "mkdb -d" but the old vc tree is still around and will not match the database when a new file is being created, which in this case the CMVC server will issue the 0010-045 when creating a file. To fix this problem, simply do "rmdb", followed by "rmfamily" and recreate the family again.

- msgErrno

  ```
  0010-063 Error, %1$s, occurred when the CMVC server software attempted
  to process the %2$s() function.
  ```

  Usually this is a problem with fork, pipe, or other system command. For more information see the technical report: **CMVC frequently asked questions: cmvcd, notifyd, cmvcarchive/restore, user exits, server errors, database issues** , section "Error message: 0010-063".

- msgExtractBadDir

  ```
  0010-326 The CMVC server software cannot create directory %1$s
           relative to the destination extraction directory.
  ```

  Check that the parent directory permits the CMVC server software to write to it. Also check for the existence of the destination extraction directory and its access permissions. If the destination extraction directory exists, ensure that its permissions allow the CMVC server software to write to it.

- msgErrno3

  ```
  0010-350 The error (%1$s) occurred when the CMVC server software
           processed function %2$s() on the file with path name %3$s.
  ```

  Usually, open, read, or write, failures. Check that the path name, the file permissions, and the directory permissions are correct.

- msgExtractChmodFailed

  ```
  0010-380 The chmod request on file %1$s
           to a mode of %2$o failed during the extraction.
  ```

  The file is extracted, however, the permissions cannot be changed. Check if the server has permission to do that. You can change the permissions manually.

- msgExtractRemoveNoWriteAccess

  ```
  0010-489 The CMVC server software does not have write access to the
           directory /ms/dev/releng/scmtool/1.0/src on node NOMOUNT.

           Check the permissions of the exported directory, make
           the necessary changes, and then retry your request.

           One workaround is to use the new environment variable
           CMVC_EXTRACT_MOUNT_NAME to specify a different mount point.
  ```

  When the CMVC server generates this error message, another message with more detailed information is also placed in the syslog, such as:

  ```
  0010-350 The error (Permission denied) ... function access() ...
  ```

  Or the following one:

  ```
  0010-063 with chown() ...
  ```

  In case of getting the 0010-063 error message, then it is very likely that the file permissions and/or ownership of "cmvcd" are incorrect. For information on how to fix the problem see “Verify the file permissions and ownership of cmvcd” on page 6.

- msgExtractCannotOverwrite

  ```
  0010-499 File %1$s exists in the destination
           extraction directory with permissions that prevent the
           CMVC server software from overwriting it.
  ```

Extract the files to an empty directory or change the access permissions of the file or destination directory to permit the CMVC server software to overwrite the file during the extraction.

- msgClientNoFreeSpace

  ```
  0010-677 The CMVC action was aborted while attempting to extract
           file %1$s .
  ```

  The file system does not have enough available space.

- msgClientBadStatFile

  ```
  0010-678 The CMVC action was aborted because the CMVC server
           could not access the extracted file %1$s.
  ```

  The file system does not have enough available space.

- msgErrGetPWUIDFailed

  ```
  0010-802 The getpwuid() operating system subroutine failed when the
           CMVC client software or the CMVC server software tried to obtain
           the user's login information.
  ```

  CMVC only passes the login name, not the uid from the client to the server when doing a Release/Level -extract. Therefore, the login name needs to be translated by the server by using getpwuid(). If the login name is not known through NIS, AFS or /etc/passwd, then getpwuid gets an error. Thus far the only known problems have been when users are working in AFS space and their token expires or their ACLs are not set for the CMVC server to read/write the files. For more details on AFS, see “Using CMVC with AFS” on page 43.

- msgCrlfConvertFailed

  ```
  0010-817 The CRLF conversion failed in %1$s.
  ```

  There are several reasons to cause this failure:

  – Cannot open temporary file (out of /tmp space).
  – Cannot open the target file for write.
  – The system calls fread() or fwrite() failed.

- msgFileExtractTxFailed

  ```
  0010-886 An error occurred while transferring file, /blah/blah
           from the CMVC server. Check your local file system.
  ```

  When the CMVC server extracts a file, a temporary copy is placed in /tmp and then this copy is transferred into the final directory.

  Thus, if this error message occurs, then either there is no enough available space in /tmp or something is wrong with /tmp, in which case you can try the Unix command "fsck /tmp" to find out the integrity of /tmp; you can use the environment variable TMPDIR to define a different directory/file system for the location of the temporary variables.

MESSAGES RELATED TO THE FORMAT OF DATE AND TIME

The following error messages are related to the format of the specified date. You can use "-date yyyy/mm/dd" to indicate that files modified in a release/level after the specified date are to be extracted.

- msgExtractDateBadFormat

  ```
  0010-328 The format for the date, %1$s, is not valid.  The date
           must be in the format yyyy/mm/dd. For example,  1991/04/28
  ```

- msgExtractDateBadMonth

  ```
  0010-330 The value for month, %1$s, is not valid.  The valid
           range for month is 01 to 12.
  ```

- msgExtractDateBadDay

  ```
  0010-331 The value for day, %1$s, is not valid.  The valid
           range for day is 01 to 31.
  ```

- msgExtractDateBadSubString

  ```
  0010-461 The format of the date string, %1$s, is not valid.
  ```

CMVC 2.3.0 note: CMVC version 2.3.0 or previous, support 2-digit year format, that is, the format of the date is, yy/mm/dd.

MESSAGES RELATED TO NFS MOUNT AND FILE EXTRACTION

The following messages are related to NFS mount when a real node is specified (that is, not using "NOMOUNT"), and when files are extracted across NFS mounted file systems. Contact the family administrator to resolve the NFS problems.

- msgExtractVCErr

  ```
  0010-025 An error occurred when the CMVC server software tried to
           extract files from the version control file repository.
  ```

  It could be an open file failure, a file permission problem, and an SCCS or PVCS problem (in this case, run some SCCS or PVCS command manually to see if it is working properly).

- msgExtractMountErr

  ```
  0010-325 The CMVC server software cannot mount directory %1$s
           from node %2$s.  The CMVC server software requires
           mount and write access to the directory to perform
           a remote extraction.
  ```

Contact your system administrator to see if the NFSD is running and, if necessary, to manually mount/umount the directory or to export the directory with write access. Verify that indeed the CMVC family userid can write into that file system.

- msgUmountBad

```
0010-339 The CMVC server software cannot unmount the directory %1$s
         which was mounted from node %2$s.
```

Contact your system administrator to see if the NFSD is running and, if necessary, to manually mount/umount the directory or to export the directory.

- msgRemoveMountBad

```
0010-340 The CMVC server software cannot remove the directory %1$s.
         The extraction cleanup procedure cannot be completed.
```

Contact your system administrator to see if the NFSD is running and, if necessary, to manually mount/umount the directory or to export the directory.

- msgFileNetBadTransferToClient

```
0010-351 An error occurred when the CMVC server software tried to open
         a temporary file on the CMVC server prior to transferring
         a file from the CMVC server to the CMVC client.
```

Check to see if the /tmp (temporary directory) is out of space. Contact the family administrator for further problem resolution.

- msgExtractCannotUmount

```
0010-492 The unmount of directory %1$s failed.
```

Contact your system administrator to see if the NFSD is running and, if necessary, to manually mount/umount the directory or to export the directory.

- msgUndefinedVCType

```
0010-638 The version control type, %1$s, is not valid.
```

Verify that the environment variable "CMVC_VCTYPE" is defined and exported properly. SCCS and PVCS are the only supported version control systems.

- msgExtractBadHost

```
0010-673 Hostname %1$s is not a known host.
```

Check for proper command format and that the hostname is known to the CMVC server's nameserver.

- msgNFSDisableNoUidGid

```
0011-151 Warning: When CMVC_NFS_DISABLE is set to YES, or -node = NOMOUNT,
         -uid and -gid values are ignored. Extract is not interrupted.
```

This is just a warning and does not affect the extract process. It tells you that if you use NOMOUNT or set the environment variable CMVC_NFS_DISABLE=YES, the gid and uid are not used.

MESSAGES RELATED TO THE VERSION CONTROL SYSTEM

The following messages are related to your version control system (SCCS or PVCS). For more details, see the technical report: **CMVC frequently asked questions: version control and database synchronization issues**.

- msgVCGetFileCannotOpen

  ```
  0010-510 File %1$s could not be
           retrieved.  A temporary file could not be created
           because the CMVC server software does not have write access.
  ```

- msgVCGetFailed

  ```
  0010-511 An error occurred when the CMVC server software tried to
           get file %1$s.
  ```

- msgVCGetFileDirExists

  ```
  0010-595 A file and a subdirectory cannot have the same name.
           File %1$s cannot be extracted.
  ```

- msgVCGetFileBadWrite

  ```
  0010-604 File %1$s could not be
           retrieved. The CMVC server software can not create this
           file on the CMVC client, %2$s.
  ```

- msgBinVersionInvalidSID

  ```
  0011-106 An error occurred when the CMVC server software tried to
           get the file %s from the version control repository.
           An invalid version number was received.
  ```

- msgBinVersionMergeFailed

  ```
  0011-107 An error occurred when the CMVC server software tried to
           get the file from the version control repository. The
           server failed to merge the binary files, %s and %s.
  ```

- msgBinVersionMergeMallocFailed

  ```
  0011-155 binmerge could not allocate temporary memory.
  ```

  It may be necessary to stop some processes or otherwise reduce memory consumption before retrying.

- msgBinVersionMergeWriteFailed

  ```
  0011-156 binmerge could not write to a temporary file.
  ```

It may be necessary to free disk space before retrying.

- msgBinVersionMergeOpenFailed

  ```
  0011-157 binmerge could not open file %s.
  ```

  It may be necessary to free disk space or inodes. It is also possible that some versions of this file may be corrupted.

- msgBinVersionMergeFileCorrupt

  ```
  0011-158 binmerge detected a truncated data file: %s.
  ```

  It is likely that the current difference file is corrupt.

APPENDIX A. DESCRIPTION OF THE TOOLS AND HOW TO GET THEM

DESCRIPTION OF THE TOOLS EXPLICITLY PREPARED FOR THIS TR

Sample mount script

This sample mount script is used for debugging problems with the behavior of the real "mount" command during the execution of the CMVC `Release -extract` command.

Note: This is not to be used in a production environment. There are some return codes that are lost due to the interaction of this script with CMVC.

This shell script is a front end for the real /etc/mount command that is invoked by other programs, such as the CMVC server. The objective is to provide a way to specify different parameters that can be tailored by the system administrator in case that the network is busy.

See the header of the tool for the actual strings used by the CMVC server to build the command that will invoke the mount command. Notice that there are 2 versions of the parameters, one for OS/2 and another for Unix, it is necessary to identify in this shell script which version is used.

To use this sample script, do the following:

1. Rename the real mount command to something like "/etc/mountx"
2. Save this shell script as /etc/mount
3. Edit this script to use the proper positional parameter that is used by the CMVC server for the platform.

   See the section labeled "CHANGE HERE" inside the sample script.

Extracting the committed versions of the files for a release.

This tool is the equivalent of doing the full extract of the latest committed level for a release.

Syntax of the tool:

```
Korn: extractCommittedFilesForRelease \
              releaseName relativePathName

REXX: xFilRelC releaseName relativePathName

ENVIRONMENT VARIABLES:
                      CMVC_FAMILY
                      [CMVC_BECOME]

DESCRIPTION:  This sample command extracts all the committed versions
              of the files associated with a release and places them
              in the path specified by the relativePathName parameter.
```

Extracting the current versions of the files for a release.

This tool extracts the current versions of the files for a release.

Syntax of the tool:

```
Korn: extractCurrentFilesForRelease \
              releaseName relativePathName

ENVIRONMENT VARIABLES:
                      CMVC_FAMILY
                      [CMVC_BECOME]

DESCRIPTION:  This sample command extracts all the current versions
              of the files associated with a release and places them
              in the path specified by the relativePathName parameter.
```

Extracting all the files for an uncommitted level

This tool is the equivalent of doing a delta extract of the latest uncommitted level for a release. Usually it is performed after the committed files for a release have been extracted.

Syntax of the tool:

```
Korn: extractFilesForUncommittedLevel \
              releaseName levelName relativePathName

REXX: xFilLevU releaseName levelName relativePathName

ENVIRONMENT VARIABLES:
                      CMVC_FAMILY
                      [CMVC_BECOME]

DESCRIPTION:  This sample command extracts the files associated with
              an uncommitted level in the specified release
              and places them in the path specified by relativePathName.
```

Extracting all the files for a committed level

This tool is the equivalent of doing a delta extract of a committed level for a release. That is, only the committed file changes done in this level are extracted.

Syntax of tool:

```
Korn: extractFilesForCommittedLevel \
               releaseName levelName relativePathName

REXX: xFilLevC releaseName levelName relativePathName

ENVIRONMENT VARIABLES:
                       CMVC_FAMILY
                       [CMVC_BECOME]

DESCRIPTION:  This sample command extracts the files associated with
              a committed level in the specified release
              and places them in the path specified by relativePathName.
```

Extract all the files associated with a committed track

This tool extracts all the files associated with a committed track.

Syntax of tool:

```
Korn: extractFilesForTrack \
               releaseName trackName relativePathName

REXX: xFilTrk  releaseName trackName relativePathName

ENVIRONMENT VARIABLES:
                       CMVC_FAMILY
                       [CMVC_BECOME]

DESCRIPTION:  This sample command extracts all the files associated
              with a specific track and places the files in the path
              specified by relativePathName.
```

Extracting all the files in a component for a single release

This tool extracts all the files associated with a specific component and a specific release.

Syntax of tool:

```
Korn: extractAllFilesForComponent \
      componentName releaseName relativePathName [committed]

ENVIRONMENT VARIABLES:
                       CMVC_FAMILY
                       [CMVC_BECOME]

DESCRIPTION:  This sample command extracts all the files associated with
              a specific component and a specific release.
              The files are placed in a directory specified by
              the relativePathName parameter.
              The default is to extract the current version.
```

Extracting all the files in a component for all releases

This tool extracts all the files associated with a specific component for all releases.

Syntax of tool:

```
Korn: extractAllFilesForComponent \
               componentName relativePathName [committed]

REXX: xFilComp componentName relativePathName [committed]

ENVIRONMENT VARIABLES:
                       CMVC_FAMILY
                       [CMVC_BECOME]

DESCRIPTION:  This sample command extracts all the files associated with
              a specific component. The files are placed in a directory
              that represents the release name to which the version of
              the file is associated.  This directory is created relative
              to the relativePathName parameter.
              The default is to extract the current version.
```

Obtaining the Tools

The tools described in this technical report can be downloaded as follows:

- From the IBM intranet (only for IBM employees).
- From the Internet (open to everyone).

IBM Intranet

Web Home Page: You can access the CMVC Service/Development Home Page at:

`http://tc-cmvc.raleigh.ibm.com/cmvc/`

From the index at the top of the page, select the section "Release extract tools".

FTP: You can download the code from our internal FTP site, by doing:

1. ftp tc-cmvc.raleigh.ibm.com
2. login as 'anonymous' and for password give your email address.
3. cd e:/cmvc/doc/tr
4. binary
5. get fileName
6. quit

Internet

Web Home Page: Not available.

FTP: You can download the code from our external FTP site, by doing:

1. ftp ftp.software.ibm.com
2. login as 'anonymous' and for password give your email address.
3. cd ps/products/cmvc/fixes
4. binary
5. get fileName
6. quit

DESCRIPTION OF AUXILIARY TOOLS

How to package and compress files

Sometimes it is necessary to package a set of files into a single one in order to perform a backup copy or to send them to another system. Furthermore, in order to save space, the file that contains the packaged files can be compressed.

All platforms: using zip and unzip from Info-Zip

The VisualAge TeamConnection team uses the Info-Zip "zip" and "unzip" tools to package compressed files (in which the files to be packaged are compressed first). The main advantages of Info-ZIP are:

- Compatibility: these tools are compatible with other ZIP programs.
- Portability: they are available in ALL the platforms that are supported by VisualAge TeamConnection.

  Furthermore, a zip file prepared in Unix can be unzipped in the correct format in Windows NT and vice versa. This is something that the other ZIP programs such as "gnuzip" or "gzip" cannot do.

Info-ZIP's software (zip, unzip and related utilities) is free and can be obtained for the desired platforms from various anonymous-ftp sites, including the URL:

```
ftp://ftp.uu.net:/pub/archiving/zip/
```

To compress and package files do the following:

1. Change to the directory where you have the files to be processed.
2. Issue the following command to compress and package:

   ```
   zip  -r  myfile.zip  fileList
   ```

   The -r flag indicates that if there are subdirectories, they need to be recursively traversed.

   The fileList can be a single file (file.c), a list of files (file.c file.h) or using wildcards (file*).
3. To only view the contents of the zip file (without actually unpackaging and uncompressing the files) do:

   ```
   unzip -l myfile.zip
   ```
4. To unpackage and uncompress the zip file do:

   ```
   unzip myfile.zip
   ```
5. To fine more information about these tools simply invoke them without any parameters.

Unix: using tar and compress

In case that you do not want to use the Info-Zip tools, you may want to do the following process to package and compress files:

1. Change to the directory where you have the files to be packaged.
2. Issue the tar command to package the files (without compression):

   ```
   tar -cvf file.tar  fileList
   ```

The fileList can be a single file (file.c), a list of files (file.c file.h) or using wildcards (file*).

3. Issue the compress command to reduce the size of the tar file:

   ```
   compress  file.tar
   ```

   The file will be compressed and the file name will be changed to reflect that it is compressed. The new name will have a suffix of uppercase Z at the end, such as "file.tar.Z".

4. Issue the uncompress command to restore the tar file to its original size:

   ```
   uncompress file.tar.Z
   ```

   The file will be uncompressed and the file name will be changed to its original name, such as "file.tar".

5. Issue the tar command to list the table of contents (without actually unpackaging the files)

   ```
   tar -tvf file.tar
   ```

6. Issue the tar command to unpackage (extract) the files:

   ```
   tar -xvf file.tar
   ```

CR (carriage return) and LF (line feed)

The end of a line of text in OS/2 or in Windows is represented by the character pair CR-LF (carriage return and line feed), whereas in Unix is represented simply by the character LF (line feed).

In CMVC all files are normalized to the Unix standard. This means that if a user creates a text file in CMVC, the CMVC server will ensure that the file has only LFs and not CRLFs. However, there are choices in the -extract action to request on-the-fly-conversions when extracting a file into an Intel platform.

In contrast, VisualAge TeamConnection does not convert the text files and they are stored as-is. The server will convert the text file according to the platform of the client (if the -crlf flag is specified).

The following file will be the source file to be used in the rest of the examples in this section:

```
This is line 1
This is line 2
This is line 3
```

If the source file is created from an OS/2 client and later on is extracted into a Unix client without CRLF conversion, then the resulting file will have the CR character at the end of each line and the file would look like:

```
This is line 1^M
This is line 2^M
This is line 3^M
```

Note: The ^ symbol is similar to an inverted V, located in the Shift position for the numeric key "6" in the alphanumeric keyboard (not in the numeric keyboard in the right side) for English for the USA.

If the source file is created from a Unix client and later on is extracted into an OS/2 client without CRLF conversion, then the resulting file will not have the CR character at the end of each line and the file would look like:

```
This is line 1
              This is line 2
                            This is line 3
```

Using the tool fixcrlf to fix CR-LF conversion problems

If you are encountering the CR-LF problems mentioned in “CR (carriage return) and LF (line feed)” on page 65 with your files when exchanging text files between Unix and Intel, then you can use the **fixcrlf** tool.

This tool can be obtained from the VisualAge TeamConnection ftp site:

1. ftp ftp.software.ibm.com
2. login anonymous
3. cd ps/products/teamconnection/tools
4. ascii
5. get fixcrlf.txt
6. binary
7. Download the corresponding file for your platform:
   - AIX: aixfixcrlf
   - HP-UX: hpfixcrlf
   - Solaris: fixcrlf.solaris
   - OS/2: fixcrlf2.exe
   - Windows: fixcrlfw.exe
8. quit

The file fixcrlf.txt provides more details about using the tool.

Also, you can simply invoke the tool with no parameters to find out how to use it.

APPENDIX B. BIBLIOGRAPHY

For more information on how to use CMVC, you can consult the following manuals and publications:

```
SC09-1596-01  IBM CMVC Client Installation and Configuration
SC09-1597-01  IBM CMVC User's Reference
SC09-1631-02  IBM CMVC Server Administration and Installation
SC09-1633-00  IBM CMVC Concepts
SC09-1635-01  IBM CMVC Commands Reference
```

The following Redbooks offer practical advice on CMVC:

```
GG24-4178-00  Did you say CMVC?
GG24-4345     CMVC: Customer's Perspective
```

HOW TO GET ELECTRONIC COPIES OF MANUALS AND TECHNICAL REPORTS

Many of the manuals and technical reports mentioned in this document can be downloaded as follows:

- From the IBM intranet (only for IBM employees).
- From the Internet (open to everyone).

IBM Intranet

Web Home Page

You can access the CMVC Service/Development Home Page at:

```
http://tc-cmvc.raleigh.ibm.com/cmvc
```

From the index at the top of the page, select:

- Technical Reports and related tools
- Copies of ALL the archived versions for the forums CMVC and CMVC6000
- Documentation in ASCII text files

FTP

You can download the code from our internal FTP site, by doing:

1. ftp tc-cmvc.raleigh.ibm.com
2. login as 'anonymous' and for password give your email address.
3. cd e:\cmvc\doc\tr
4. binary
5. get fileName
6. quit

Internet

Web Home Page

Not available.

FTP

You can download the code from our external FTP site, by doing:

1. ftp ftp.software.ibm.com
2. login as 'anonymous' and for password give your email address.
3. cd ps/products/cmvc/doc
4. binary
5. get fileName
6. quit

APPENDIX C. COPYRIGHTS, TRADEMARKS AND SERVICE MARKS

The following terms used in this technical report, are trademarks or service marks of the indicated companies:

TRADEMARK, REGISTERED TRADEMARK OR SERVICE MARK	COMPANY
IBM AIX OS/2 CMVC VisualAge TeamConnection	IBM Corporation
PostScript	Adobe Systems Incorporated
NetLS iForLS, iFor	Gradient Technologies, Inc.
HP-UX	Hewlett-Packard Company
Info-ZIP	Info-ZIP Group
Intel	Intel Corp.
Microsoft Windows Windows NT Windows 95 MS-DOS	Microsoft Corporation
Novell Netware	Novell Inc.
Oracle	Oracle Corp.
OSF DFS DCE	Open Software Foundation, Inc.
NFS SunOS Solaris	Sun Microsystems Inc.

AFS	Transarc Corp.
Unix	X/Open Co., Ltd.

END OF DOCUMENT