How to do migration tasks with CMVC Second Edition

Document Number TR 29.2232

Angel Rivera, Lee Perlov, and Clifford Meyers

TeamConnection/CMVC Development
IBM Software Solutions
Research Triangle Park, North Carolina
Copyright (C) 1997 IBM
All rights reserved.

ABSTRACT

This technical report provides useful information for CMVC family administrators that want to do the following migration tasks related to CMVC:

Changing the name of an existing CMVC family.
Migrating a CMVC family from one host to another.
Migrating a CMVC family from one DB2 instance to another.
Migrating data from library systems other than SCCS into CMVC.
Miscellaneous migration issues.
Preparing to migrate to TeamConnection.
Migration to CMVC 2.3.1 (Year 2000 compliant)

ITIRC KEYWORDS

CMVC

TeamConnection

Migration

Migrate

Second Edition, CMVC 2.3.1.1 (CMVC 2.3.0.25)

Added the section on how to migrate from CMVC 2.3.0 (which is not compliant with the Year 2000 because the years are stored with only 2 digits) to CMVC 2.3.1 which is Year 2000 compliant (the years are now stored with 4 digits).

First revision, CMVC 2.3.0.23

Converted from a FAQ previously distributed with CMVC to a Technical Report and updated based on the experiences of the CMVC development team and the ISSC CMVC support organization in Research Triangle Park, North Carolina.

ABOUT THE AUTHORS

ANGEL RIVERA

Mr. Rivera is an Advisory Software Engineer and team lead for the CMVC development. 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 a Staff Programmer in the 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.

CLIFFORD J. MEYERS

Mr. Meyers is an advisory programmer and the technical team lead for ISSC Distributed Configuration Management Services. He joined IBM in 1985 as support for AIX/370 and AIX/ESA before accepting his current assignment in 1993.

The ISSC organization in RTP provides consulting and support services for over 7,000 internal and external customers.

Cliff was also involved in the development of the original Motif GUI for CMVC and all of the tools in the appendices of this document.

CONTENTS

ABSTRACT . . . iii
ITIRC KEYWORDS . . . iii

ABOUT THE AUTHORS . . . v
Angel Rivera . . . v
Lee R. Perlov . . . v
Clifford J. Meyers . . . v

Figures . . . viii

Background on CMVC migration . . . 1
Overall recommendations . . . 1
CMVC 2.3.1 provides support for the Year 2000 . . . 1
Documentation from CMVC Version 2 Release 3 about migration . . . 2
What version of CMVC do I have? . . . 2
Example of version information from the CMVC server . . . 3
Example of version information from the CMVC UNIX GUI . . . 3

Changing the name of a CMVC family . . . 5
Scenario . . . 5
Recommendation . . . 5

Migration of a CMVC family from one host to another . . . 9
Scenario . . . 9
Recommendation . . . 9
Suggested approach . . . 9
Things to do on your current system, Host A . . . 10
Things to do on your new system, Host B . . . 10
Error 0011-111 when starting the cmvcd, after migration . . . 11
How to add a host list entry directly into the database? . . . 12

Migrating data from library systems into CMVC . . . 15
Brief explanation of SCCS . . . 15
Explanation of migration from SCCS to CMVC . . . 15
What problems can be encountered with the Import and Migration tools? . . . 16
Changes to the error handling of file.migrate and file.import . . . 17
How SCCS archives relate to CMVC releases . . . 18
Migrating from another library to SCCS . . . 18

Migrating from RCS to SCCS . . . 18
Bringing PVCS files under CMVC control . . . 18

Migration tips when moving from one version of CMVC to another . . . 21
Migrating to CMVC 2.3.1 (which supports the Year 2000) . . . 21
AIX 4: necessary links if using the en_US locale . . . 22
Warning when migrating a DB2 family from AIX 3 to AIX 4 . . . 22
Moving a family from a DB2 instance to a new one . . . 23
Migration from CMVC 2.2 using Oracle 6 to CMVC 2.3 using DB2 . . . 24
Migrating from Oracle 6 in AIX 3.2.5 to Oracle 7.1 in AIX 4.1 . . . 24
Do not migrate to Oracle 7.3 because it is not supported . . . 25
Migrating CMVC from Informix to DB2 . . . 25
0010-256 error messages after migrating from CMVC V1.1 to V2.x . . . 25
Fixing CMVC V1.1 to V2.x migration problems with "chcfg" . . . 26

Preparing to migrate to TeamConnection . . . 27
Why TeamConnection? . . . 27
Migration from CMVC to TeamConnection . . . 28

Migration shell scripts . . . 29
Obtaining the shell scripts . . . 29
IBM Intranet . . . 29
Internet . . . 29
Shell scripts to prepare the migration from CMVC to TeamConnection . . . 30
Complete a release prior to TeamConnection migration . . . 30
Reassign user's work and delete users . . . 30
Delete a release from a family . . . 30
Miscellaneous shell scripts . . . 31
Import a release from directory structure . . . 31

Copyrights, Trademarks and Service Marks . . . 33

FIGURES

1. Determine the version of CMVC in AIX, SunOS 4.1.3 and Solaris 2.x . . . 3
2. Determine the version of CMVC in HP-UX . . . 3
3. Script to create and run SQL instructions to drop indexes . . . 8

BACKGROUND ON CMVC MIGRATION

This technical report provides useful information for CMVC family administrators that want to do certain migration tasks related to CMVC.

This technical report covers new information that has been gathered by CMVC development and ISSC CMVC Support since the publishing of the CMVC 2.3 documentation.

OVERALL RECOMMENDATIONS

It is strongly recommended that you follow these hints:

Make a complete backup of your CMVC family (file system and database) BEFORE you start the migration process.

In case that the migration process does not go fine, you can restore the baseline backup.

Make a complete backup of your CMVC family (file system and database) AFTER each important step in the migration process.

If the migration process involves several major steps, by making a backup after each major step, you could restore the CMVC family before the previous step if it does not go fine.

Try to break the migration process into manageable steps, in that way you can reduce the number of variables.

For example, do not try to migrate a CMVC family from one CMVC version in one operating system version and database version to another CMVC version in another operating system version and database version (too many variables!).

CMVC 2.3.1 PROVIDES SUPPORT FOR THE YEAR 2000

CMVC 2.3.1 is a new version-release-modification of CMVC (branched from CMVC 2.3.0.24) which uses 4 digits to represent the year data instead of the 2 digits used in 2.3.0 and previous versions. Thus, CMVC 2.3.1 provides support for the Year 2000.

For details on migrating a family on CMVC 2.3.0 or previous, to CMVC 2.3.1, see “Migrating to CMVC 2.3.1 (which supports the Year 2000)” on page 21.

CMVC 2.3.0 will continue to use 2 digits to represent the year data.

DOCUMENTATION FROM CMVC VERSION 2 RELEASE 3 ABOUT MIGRATION

The primary source for CMVC documentation on migration issues is in the "CMVC Server Administration and Installation: Version 2 Release 3" manual (IBM publication number SC09-1631-02). In this technical report we refer to this manual, as the CMVC Server manual.

The Version 2.3 of the CMVC Server manual contains many corrections to earlier releases, thus, use only the 2.3 version of this manual and do not use earlier releases.

This technical report does not cover the following migration issues which are covered by the CMVC Server manual:

1. Chapter 16, "Bringing SCCS Files under CMVC Control" describes how can migrate into the CMVC environment the text files that are already being developed with SCCS commands.
2. Appendix B, "Migrating to CMVC Version 2.3" covers the task on how to migrate from older versions of CMVC to the current version.
3. Appendix C, "Converting Existing CMVC Server from ORACLE 6 to ORACLE 7" describes how to migrate from ORACLE 6 (which is not longer supported in CMVC 2.3) to ORACLE 7.
4. Appendix D, "Migrating to CMVC Server V.2.3.0 for DB2" covers the task on how to migrate from an existing CMVC family that is not using DB2 to a CMVC family that uses DB2.

WHAT VERSION OF CMVC DO I HAVE?

Because of changes in functionality between CMVC 2.1, 2.2 and 2.3, it is important to know exactly which version of CMVC you are using.

Starting with CMVC version 2.3, the appropriate version information is imbedded in the executable code and it allows you to determine what version of CMVC you are running.

The syntax for determining the version of a CMVC executable file is shown below. The CMVC server, cmvcd, is used as an example, but any other CMVC executable could be used.

For AIX, SunOS and Solaris, use the following:

```
what `which cmvcd` e grep cmvc
```

Figure 1. Determine the version of CMVC in AIX, SunOS 4.1.3 and Solaris 2.x

For HP-UX use the following:

```
what `whence cmvcd` e grep cmvc
```

Figure 2. Determine the version of CMVC in HP-UX

See the following sections for examples of the output.

If the above procedure does not return a string, then you are running CMVC Version 1.x, or CMVC 2.1 or CMVC 2.2. Only by looking at the date stamps of the files might be possible to determine the version. For example, if the date is November 1993, then you have the last CMVC 2.2 version.

Note: The CMVC server, CMVC client, and CMVC GUI are numbered independently. For this reason, in this technical report we focus on the version of the CMVC server, cmvcd.

Example of version information from the CMVC server

An example of the version information that can be obtained from the CMVC server is shown below:

```
 cmvc       CMVC Version 2.3. .22, (C) Copyright IBM Corp.,1993,1996
 cmvc       5765-2 7, 5765-2 2, 5765-397, 5622- 63
cmvc       96/ 6/24  SID=1.26.1.52
 cmvc       Platform: AIX 4
cmvc       Database:  DB2
```

Notice the version of the platform and of the database.

Example of version information from the CMVC UNIX GUI

An example of the version information that can be obtained from the CMVC UNIX GUI is shown below:

```
cmvc        CMVC Version 2.3. .21, (C) Copyright IBM Corp.,1993,1996
cmvc        5765-2 7, 5765-2 2, 5765-397, 5622- 63
cmvc        96/ 5/13  SID=1.1.3.65
cmvc        X11  R5
cmvc        Motif  1.2
```

Notice the version of the Motif and X11 toolkits used to build the executable.

CHANGING THE NAME OF A CMVC FAMILY

SCENARIO

1. Current CMVC family is on host A using a database supported by CMVC (as an example we will use Oracle 7).
2. We want to keep the CMVC family in the same host, but we want to change its name.

RECOMMENDATION

Changing the name of a CMVC family is rather straight forward, but there are a couple of things to watch out for, so here is our recommended process.

1. Save and drop the current copy of the database

   This is a simple alternative to actually renaming tables, indexes, etc. in the existing instance.

   a. Export database (sample syntax is provided in the CMVC Server manual).
   b. Execute: rmdb
   c. Drop of extra table and index space (pointed to by environment variables): ORACLE_TBLSP and ORACLE_NDXSP

2. Change the name of the family in /etc/passwd:

   In AIX, this can be done via smit.

3. Change the ownership of all files in the family account.

   a. Log in as "root"
   b. Issue the following commands:

      ```
      $ mv /home/FAMILY_NAME /home/NEW_FAM_NAME
      $ find /home/NEW_FAM_NAME -exec chown NEW_FAM_NAME {} \;
      ```

   c. Log out of root

4. Update the family ".profile"

   Update the environment variables that reference the family name, such as CMVC_FAMILY, CMVC_HOME (if used), and PATH.

   Also, this would be a good time to compare your .profile to our sample profile in /usr/lpp/cmvc/install/profile.<db>, where <db> is a database name.

5. Find all "p." files in the vc tree and change the family name in these files

   a. Log into the family account
   b. Issue the following command:

```
$ find /home/NEW_FAM_NAME/vc -type -name "p. " -print e
   while read file
  do
        awk 'BEGIN{print "%s/'${OLD_FAMILY}'/'${FAMILY}'/";       \
        print "wq"}' /dev/null e ex ${file} > /dev/null 2>&1
 done
```

c. If you would prefer to save the old copies of each file, you can use the following command:

```
$ for i in  find /home/NEW_FAM_NAME/vc -type -name "p. " -print
  do
        mv $i ${i}.old
        sed "s/${OLD_FAMILY}/${NEW_FAMILY}/g" ${i}.old > $i
        # rm ${i}.old
  done
```

6. Create the new database using Oracle:

   a. Issue: mkdb

   Note: If you originally issued mkdb -d to create the database you will either need to do it here, or perform the step 6e below.

   b. Create new table space and new index space.

   c. Update the .profile to use the new index and table space.

   d. Delete the preloaded table entries created by mkdb:

```
$ sqlplus family/passwd
> delete from users;
> delete from hosts;
> delete from config;
> delete from interest;
> delete from authority;
> delete from sequence;
> delete from components;
> delete from compmembers;
> delete from levels;
> delete from releases;
> delete from versions;
> delete from cfgrelproc;
> delete from cfgcomproc;
> commit;
> quit;
```

   e. (optionally) Run chfield for customized configurable fields. Do not use the default option (-source) when you have customized the configurable fields for a table.

```
chfield -object Defect
chfield -object Feature
chfield -object File
chfield -object User
```

f. Temporarily drop the indexes to speed up the import of data. The following shell script can be used to create the shell script **drop.index.sh**:

g. Import data.

   Please consult your database documentation for the procedure to import data. CMVC provides the command to use with Oracle 7 in step 5 of "Importing to Oracle 7" in Appendix C of the CMVC Server manual. If problems occur on a particular table, consult the database documentation for the procedure to re-try just that table. If problems occur importing the Defects, Features, Files or Users tables, see the step 6e on page 6. If all else fails, drop the table and then import it again (the table will be imported into the default table space).

h. Recreating indexes.

   You need to run a slightly modified version of the index.db utility in /usr/lpp/cmvc/install (or the one appropriate for your database). Please consult your database documentation for procedures. CMVC provides a script that modifies the index.db to use with Oracle 7 in step 6 of "Importing to Oracle 7" in Appendix C of the CMVC Server manual.

The preceding procedures can also be used for migration of CMVC families when it is desired to change operating systems, versions of CMVC, and/or versions of a database in one step. However, it takes longer than following the procedures documented in the CMVC Server manual. This procedure is particularly useful when you wish to move data from the system (default) database to separate database files.

Note: Other databases use different syntax, for example:

Access DB2/6000 sql prompt with: db2
Access Informix sql prompt with: isql
Access Sybase sql prompt with: esql
Connect in DB2/6000 is performed with: connect to <family>
Connect in Informix is performed with: connect <family>
For database backup procedures see the appropriate operating system manuals.

```
#!/bin/ksh
OUTPUT=drop.index.sh
# Create a file with instructions for converting the CMVC script
# used to create indexes into one that drops them.
# It has Oracle specific syntax.
cat <<EOF!    > sed.list
s/unique //
s/create/drop/
s/$/;/
EOF!
# The file index.db is specific to Oracle.    There are others for
# the other databases supported by CMVC.
grep create $CMVC_HOME/install/index.db e sed -f sed.list > $OUTPUT
echo "commit;
quit;" >> $OUTPUT
# Run SQL script to drop indexes
sqlplus $ORACLE_DBA @${OUTPUT}
rm $OUTPUT sed.list
exit
```

Figure 3. Script to create and run SQL instructions to drop indexes

MIGRATION OF A CMVC FAMILY FROM ONE HOST TO ANOTHER

SCENARIO

1. Current CMVC server is on Host A using a database supported by CMVC.
2. We installed the same versions of CMVC and of the database in Host B (this machine could have a different operating system).
3. We want to put the CMVC database on the new system, Host B.

RECOMMENDATION

When moving a family from one host to another there are some common problems that we have tried to capture in this example. These are our suggestions for making this task easier:

The key to making this move smooth is to make the two machines as similar as possible. For example, keep the same version of the operating system, the same version of the database, the same operating parameters, etc. Wait until the family is moved before upgrading your new system.

A related recommendation is to reduce the number of variables between the migration steps, and to avoid introducing too many variables in one single step. That is, in order to make the old Host A as similar as possible to the new Host B, take smaller steps and ensure that CMVC is operational after each step; furthermore, make a complete backup of the CMVC family after a successful step. In case of problems, you will need to back out to the previous small step instead of all the way back to the very beginning.

Make sure that all of the CMVC characteristics are the same. Specifically, keep the same port number for the family in /etc/services, the same family name (usually an entry in /etc/hosts), etc. Again, you can change these things later, after the family has been successfully moved.

SUGGESTED APPROACH

Things to do on your current system, Host A

1. Make any necessary hostlist changes for the new server, such as adding a hostlist for the superuser. If you do not do this now, then you will have to create a hostlist directly into the database after the migration.
2. Check the value of LANG and run the locale command. The new family account in Host B will need to match these values.
3. Stop the CMVC family. We suggest to use the "/usr/lpp/cmvc/samples/stopCMVC" sample.
4. Using the database utilities, make a backup of the CMVC family database in Host A. Your database manual will provide the details on how to do this. CMVC provides a sample backup utility for DB2/6000 in "/usr/lpp/cmvc/samples/backupCMVC".

   Place the backup file in the HOME directory of the CMVC family in Host A.
5. Use the utility "tar" (the utility "backup" is also available on AIX) to make a tar of the entire CMVC family account (including the backup of the CMVC database).

Note: By the way, the above sequence is also the recommended sequence for your normal daily backup of both the CMVC database and the family account in order to keep them synchronized.

Things to do on your new system, Host B

1. If using DB2, create a new DB2 instance owner and start the DB2 instance.
2. Create in Host B, a userid with exactly the same values as the userid of the existing CMVC family in Host A, for example same userid, same groupid, same directory, etc...

   Note: You need to give root CONNECT authority to the family database in order to allow the CMVC daemons to start properly. Alternately, you can just give root DBADM authority.
3. Make sure that LANG environment variable is the same as in Host A. Also run the locale command to make sure that the correct locale is installed.

   If any LC_ variables have a value of 'C', the locale may not be installed and the DB2 restore may not work.
4. Create a new CMVC family with the same setup as the old family:

   Perform "mkfamily".
   Update the /etc/host and /etc/services to reflect the new family.

   Note: In most cases, it is not necessary to perform "mkdb" because the restoring of the database backup will perform all of the same functions.

5. Untar (or use utility "restore" if you used "backup" in AIX) the complete CMVC family account obtained from Host A into Host B. This will overlay the different configuration files and replace the version control tree ($FAMILY_HOME/vc).
6. Restore the CMVC database in its proper directory, which should be identical as in Host A; for example, if using DB2, it must be in the new database instance.
7. Reorganize the indexes for the database. This step is optional but highly recommended. This is a good point in the sequence to reorganize and update the indexes for the database, to improve performance.

   If using DB2, use the command "db2reorg".
8. If using DB2, then it is necessary to run the "db2bind <family>" command in order to bind the executable code to the DB2 instance.
9. Test the migrated CMVC family. Issue several CMVC commands to verify that the users, components, files, etc... are correct.
10. Make a backup of the new CMVC family.
11. Any user who perform level or release extracts will have to NFS export the target directory to the new server.
12. Either update the nameserver entry for the new family name, or ask all users to update their /etc/hosts files.
13. If a user is using the VM client, you may need to install/start the vmkld daemon on the new server and create additional host list entries for each user.

Note: Do NOT use the CMVC cmvcarchive and cmvcrestore utilities for backups. These tools are designed to archive old releases and/or levels that will not be needed in the current family. If this archived information is needed, it can be restored into a new, hopefully temporary, family.

ERROR 0011-111 WHEN STARTING THE CMVCD, AFTER MIGRATION

Question:

After migrating a CMVC family, the error 0011-111 is shown when starting the cmvcd daemons:

```
 11-111
The CMVC server software cmvcd daemon cannot start.
The value of lastSerial where name = 'general' found in the Sequence
table is smaller than some of the id found in the Components,
Defects, Files, Levels, Path, Releases, Tracks, Users or Versions
table.  One of the above mentioned tables has been corruptted.
```

Answer:

This is a very serious message. The Sequence table is extremely important to CMVC because it provides the next number/id to be used for the internal ids for the objects. One main cause of this corruption of the Sequence table or related tables is when the user directly alters the database bypassing CMVC; another possible cause is an error during the migration.

Our recommendation at this stage is to take a backup now of both the database and of the file system of the family. Then, use one of the previous backups that were OK and reestablish your family database and file system. If at this stage the server is OK, then keep using it. If the changes between the last successful backup and the current corrupted database are very small, our recommendation is to keep using the reestablished family and incorporate the small changes back into CMVC.

HOW TO ADD A HOST LIST ENTRY DIRECTLY INTO THE DATABASE?

Question:

How to add a host list entry directly into the database?

Answer:

In some rare cases during installaiton or migration, it might be necessary to add a host list for the CMVC Superuser (which by default has the internal userid of 1) by using directly the SQL statements; by the way, this is the method used by the 'mkdb' when creating the CMVC database. The symptom of this problem is that the message 0010-057 is shown.

The sequence is shown below:

1. Logon as the CMVC family administrator
2. Connect to the database, such as:

   db2 connect to <family>
3. Go into interative SQL; in DB2 (command line processor) type:

   db2
4. Issue the following insert command (one single line)

   ```
   insert into Hosts (userId, name, login, address) \
           values (1,'$CLIENT_HOSTNAME', '$CLIENT_LOGIN', )
   ```

   The variables CLIENT_HOSTNAME and CLIENT_LOGIN represent the hostname and the login from where the superuser will login.

The CLIENT_HOSTNAME should be the string that is the complete name returned by the command "host". For example, if your host is "carcps06", then do "host carcps06" and you will get something similar to:

```
carcps 6.raleigh.ibm.com is 9.67.238.18
```

Then, the CLIENT_HOSTNAME should be "carcps06.raleigh.ibm.com".

5. Commit the transaction (otherwise the value will not be seen by CMVC), by doing:

   ```
   commit work
   ```

6. Query the Hosts table to ensure that the data is correct:

   ```
   select  from Hosts
   ```

7. Terminate the DB session:

   ```
   terminate
   ```

MIGRATING DATA FROM LIBRARY SYSTEMS INTO CMVC

BRIEF EXPLANATION OF SCCS

The SCCS library system is provided free with most Unix operating systems. It does not have tracking, it cannot handle binary files and it does not provide a GUI; therefore this tool is easy to outgrow. That is why CMVC provides a migration from SCCS.

Of course, there are many other simple library systems whose users would benefit from CMVC.

For a useful shell script see “Import a release from directory structure” on page 31

Explanation of migration from SCCS to CMVC

Note: The following explanation was given by Gary Warner, Software Tools Support, IBM Austin.

All of the migration scripts for SCCS files (Filemap, Fileimport, Filemigrate, and xecit) are found in the /usr/lpp/cmvc/bin directory on the CMVC client.

You can use these SCCS files provided you import or migrate them into the CMVC development environment. The SCCS File Import and Migration Tools provide a means by which CMVC users can bring SCCS text files into the CMVC development environment.

To use these tools, the destination CMVC family must use SCCS as its underlying version control mechanism.

The SCCS File Import and Migration tools are documented in the 'Bringing SCCS Files Under CMVC Control' section of the IBM AIX CMVC/6000 Administration and Installation guide. Refer to the discussion in this section for the advantages/disadvantages of each method.

When SCCS files are imported, their version number in the CMVC development environment becomes 1.1. We recommend that users keep a copy of their map files so that they can look back to which version of each file they imported.

When SCCS files are migrated, all of the versions of the file are brought into the CMVC development environment. The user specifies which version is to be known as the current version in a release. The user does not have to assign each branch of an SCCS file to a release; they can do this later.

When SCCS files are migrated into the CMVC development environment, the version history and remarks will be captured. If a CMVC user id exists for the user who created the SCCS version, then this information will be retained. If a CMVC user id does not exist for the user who created the SCCS version, then the information will be recorded as the user who is performing the migration.

Common files are supported with the import and migration tools if all releases that are to contain common files are processed by the Fileimport or Filemigrate scripts simultaneously. This is accomplished by supplying more than one map file name as parameters to these scripts.

The Migrate command is used to bring an SCCS file into CMVC. This is essentially File -create with the command and action changed to Migrate -migrate and with an additional -version flag.

The File -create command creates an s. file at version 1.1 from the input file. The Migrate command creates all versions in an input s. file and makes one of them the "current" version.

The Migrate command uses a shell script named sclean which is in the CMVC bin directory (with cmvcd) to "clean" the file and to tell the Migrate command what versions are in it. The "cleaning" task means to remove any flags that restrict access of the file to certain logins, and sets a flag to allow multiple gets for edit.

The sclean script writes the version information to stdout which is then read by the Migrate command. This information is:

#versions
deltanum previous SID date time username
remark lines, if any
<a blank line>
repeat for other versions in deltanum order.

The sclean script gets this stuff from the header of the SCCS file. The SCCS utility numbers each delta as it is created. This is "deltanum". The "previous" is deltanum of the version from which the given version was created, with previous = 0 for the first delta. The "username" is the login which created the delta. If this matches a CMVC login, Migrate uses that login as the creator. If it does not, the user issuing the Migrate command gets the credit.

WHAT PROBLEMS CAN BE ENCOUNTERED WITH THE IMPORT AND MIGRATION TOOLS?

Question:

What problems can be encountered with the Import and Migration tools?

Answer:

Some common situations that cause problems with these migration tools are shown below:

The file.import or file.migrate files do not have execute permission.
The Filemap, Fileimport or Filemigrate scripts are being run by a user who has insufficient AIX user authority to create files within the current directory or read the SCCS files in the directories in which they reside.
The scripts are being run from a client that does not have the CMVC client.
The components and releases have not been created in the CMVC environment yet.
The files being imported or migrated already exist in the CMVC environment.
The user running the import or migration scripts has insufficient CMVC access authority (FileAdd and FileLink are required).
A defect or feature is in the wrong state.
A track or a fix record is in the wrong state.
The CMVC server's vc tree is full.

CHANGES TO THE ERROR HANDLING OF FILE.MIGRATE AND FILE.IMPORT

CMVC 2.2 and initial versions of CMVC 2.3 made use of a utility for performing error checking on the file.migrate and file.import script files. However, most users forgot to use xecit and were unable to determine whether or not their data was properly loaded into the CMVC family. As a result, later versions of CMVC 2.3 now puts the error checking directly into the file.migrate and file.import script files.

The current file.migrate and file.import script files, generated from FileMigrate and FileImport, include error handling. If an error occurs, then a file.migrate.err or file.import.err file will be generated and will identify the errors. Except in extreme cases, this file will be very small, if it is created at all.

If a file.migrate.err or file.import.err file is generated, you can use the "xecit" command to execute each line in these files and to capture the errors in an executable file like file.migrate.err or file.import.err. The utility xecit is documented on pages 131 and 132 in the CMVC Server manual.

You may find xecit useful for any script where you do not want to include the error checking code in it.

HOW SCCS ARCHIVES RELATE TO CMVC RELEASES

It has been asked why a single SCCS archive cannot always be migrated into CMVC in one release. The basic answer is that CMVC can only migrate one branch of the version tree in an SCCS archive in a single release.

For example, if s.foo archive for the foo file has file versions 1.1, 1.2, 1.3 and 1.2.1.1, then a conscious decision was made that 2 different file versions would be created as successors to version 1.2. A CMVC release can contain version 1.1, 1.2 and 1.3, but would not be able to contain 1.2.1.1 because it is conflicting with version 1.3. Therefore, a second release would need to be migrated that would contain versions 1.1, 1.2 and 1.2.1.1. In both cases, versions 1.1 and 1.2 could be linked between both releases.

In either case, the file foo would be common to both releases because CMVC would internally use one archive to store the versions.

MIGRATING FROM ANOTHER LIBRARY TO SCCS

The easiest way to get from these libraries to CMVC is by migrating first to SCCS, then from SCCS to CMVC.

Migrating from RCS to SCCS

For example, it has been reported to us that there is an RCS to SCCS conversion tool available at: ftp://www.cyclic.com/pub/cvs/contrib/cvs-1.8.5 as part of the package cvs-1.8.5.tar.gz. If you use gnu zip to unzip to file, you will find a Korn shell script for converting from RCS to SCCS in the contrib directory. You can get to this site by using http://www.cyclic.com.

Bringing PVCS files under CMVC control

Question:

Is there a recommended way to bring PVCS files (on OS/2) under the control of CMVC (on AIX)?

Answer:

In order to provide some advice for PVCS migration, it is important to understand the migration from SCCS:

There are several scripts which are used to help build the proper arguments for the CMVC Migrate commands to migrate files from an SCCS source tree.

The CMVC Migrate command is used to bring an SCCS file into CMVC. This is essentially "File -create" with the command and the action changed to "Migrate -migrate" and with an additional "-version" flag.

The CMVC "File -create" command creates an "s." file at version "1.1" from the input file. The CMVC "Migrate -migrate" command creates all versions in an input "s." file and makes one of them the "current" version.

The CMVC Migrate command uses a CMVC shell script named "sclean" to "clean" the file and tell Migrate what versions are in it. The term "cleaning" means to remove any flags restricting access of the file to certain logins and to set a flag to allow multiple gets for edit.

The script sclean writes the following version information to stdout which is then read by the Migrate command:

#versions
deltanum previous SID date time username
remark lines, if any
<a blank line>
repeat for other versions in deltanum order.

The script sclean gets this stuff from the header of the SCCS file. SCCS numbers each delta as it is created and this is "deltanum". The field "previous" is deltanum of the version from which the given version was created, with previous = 0 for the first delta. The field "username" is the login which created the delta. If this matches a cmvc login, then Migrate uses that login as the creator. If it does not, then the user that is issuing the Migrate command gets the credit.

So, if you can make a PVCS version of sclean and put it in the same directory as sclean, then you should be able to use the Migrate command to bring PVCS files into CMVC.

MIGRATION TIPS WHEN MOVING FROM ONE VERSION OF CMVC TO ANOTHER

The following tips will make it easier to perform several migration tasks related to CMVC.

Please follow the recommendations described in “Overall recommendations” on page 1.

MIGRATING TO CMVC 2.3.1 (WHICH SUPPORTS THE YEAR 2000)

The CMVC 2.3.1 version-release-modification provides support for the Year 2000 by using 4 digits instead of 2 to represent the year. For more information, see “CMVC 2.3.1 provides support for the Year 2000” on page 1.

To migrate a family that uses CMVC 2.3.0 or previous version, please follow these steps:

1. Make a backup as recommended in “Overall recommendations” on page 1.
2. Migrate CMVC to 2.3.1:

   If migrating from CMVC 1.x, 2.1, or 2.2, then follow the procedure explained in the Appendix B of the CMVC Server 2.3 manual.

   Then continue with step 3.

   If migrating from CMVC 2.3, then it is just necessary to install CMVC 2.3.1.

   If using DB2, you need to perform the "db2bind" utility for each family.

   Then continue with step 3.
3. Run the 2.3.1 migration shell script in order to modify all the instances for the date in the database from "yy/mm/dd" to "yyyy/mm/dd":

   /usr/lpp/cmvc/install/dbSetDate <database> <family> <logfile>

   Where <database> is ORACLE7 (for ORACLE 7)
   INFORMIX (for Informix)
   DB2 (for DB2)
   SYBASE (for Sybase)
   Where <family> is the name of your family
   Where <logfile> is the name of a file to append the SQL output

   For example, to update the date for a "cmvctest" family that uses DB2 and to store the SQL output in the file "sql.out" do the following:

   dbSetDate DB2 cmvctest sql.out

4. Perform a quick sanity check for the migrated families; some tests that you can do are to verify the new format for the year (yyyy) are:

   List all users and verify the columns with dates.
   View one user and verify the dates.
   From a Open List (Filter) Users window, issue a query that lists all users created after a certain date, such as:

   ```
   addDate > '1997/ 1/ 1'
   ```

   List other CMVC objects such as defects, files, releases, components and verify the dates.
   Do a release extract using the -date flag.
   If possible, do a release link using the -date flag.

   You could create a dummy release just to do the link and then undo the links and delete the release.

5. In case that you have not done it before, add the new indexes to improve performance that were added in 2.3.0.18 and which are listed in the file:

   ```
   /usr/lpp/cmvc/doc/README.pubs
   ```

AIX 4: NECESSARY LINKS IF USING THE EN_US LOCALE

If you are going to use the "en_US" locale in AIX 4 (which is the the default), then it is suggested to do the following links because the only locale provided by CMVC is En_US:

1. Login as root
2. Do the following symbolic links:

   ```
   ln -s /usr/lib/nls/msg/En_US/cmvc.cat         /usr/lib/nls/msg/en_US/cmvc.cat
   ln -s /usr/lib/nls/msg/En_US/cmvchelp.cat /usr/lib/nls/msg/en_US/cmvchelp.cat
   ln -s /usr/lib/nls/msg/En_US/cmvcui.cat      /usr/lib/nls/msg/en_US/cmvcui.cat
   ```

WARNING WHEN MIGRATING A DB2 FAMILY FROM AIX 3 TO AIX 4

The following section was added to /usr/lpp/cmvc/doc/README.pubs:

g) New DB2_CODESET and DB2_TERRITORY variables

The mkdb CMVC installation utility (which in turn calls the mkrmchdf shell script) and the sample profile.db2 were changed to use codeset/territory for DB2 during the creation of a DB2 database for the CMVC family. The new variables are DB2_CODESET and DB2_TERRITORY.

In common situations this is not really needed, but when migrating CMVC DB2 families from AIX 3 to AIX 4 it is critical that these variables are in sync: in AIX 3 the default is En_US.IBM-850 but in AIX 4 the new default is en_US.ISO8859-1.

If the user does not specify these variables, they will default to the proper value according to the version of AIX.

See the sample /usr/lpp/cmvc/install/profile.db2 for actual usage of these variables.

In short, if you have a family created in AIX 3 that used the defaults:

```
LANG=En_US
DB2_CODESET="IBM-85 "
DB2_TERRITORY="En_US"
```

Then you need to install the locale En_US in AIX 4, and specify the above 3 variables when creating the database in AIX 4; do not use the default en_US locale in AIX 4 for that family.

MOVING A FAMILY FROM A DB2 INSTANCE TO A NEW ONE

The recommended sequence to move an existing family from one DB2 instance to another is as follows:

1. Create and start the new DB2 instance.
2. Login as the family account.
3. Stop the family and make a backup of the database by using "db2 backup database ...".
4. Change the family's DB2INSTANCE environment variable to point to the new instance, and change DB2_DBPATH to point to the new location of the family in the new instance.

   Logoff and login again to refresh the environment variables.
5. Make the family userid a member of the new instance's primary group in order to have SYSADM authority on the database.
6. Use "db2 restore database ..." to restore the database. If you do a test run first, you will have to uncatalog the new database before you can restore it again.
7. Restart the family

This sequence will leave all the old database files on the system and leave the old database cataloged on the old instance. At some stage you will want to delete all these old database files and to uncatalog the old database from the original instance.

MIGRATION FROM CMVC 2.2 USING ORACLE 6 TO CMVC 2.3 USING DB2

This note describes the overall migration scenario from CMVC 2.2 using Oracle 6 to CMVC 2.3 using DB2. For mode details, see the appendices B and D from the CMVC 2.3 Server manual.

1. Migrate from CMVC 2.2 for Oracle 6 to CMVC 2.2 for DB2.

   This is necessary because there is no support for Oracle 6 on CMVC 2.3, only for Oracle 7 (Oracle dropped support for version 6 in December 1994).

2. It is necessary to perform db2bind after the migration to DB2.
3. Verify that the migration is successful before continuing.
4. Upgrade from CMVC 2.2 for DB2 to the latest version of CMVC 2.3 for DB2.
5. It is necessary to perform db2bind in the new family.
6. Verify that the upgrade was successful.

MIGRATING FROM ORACLE 6 IN AIX 3.2.5 TO ORACLE 7.1 IN AIX 4.1

The migration route that we recommend is as follows:

1. Migrate from CMVC 2.1/2.2 using Oracle 6 on AIX 3.2.5 to CMVC 2.1/2.2 using Oracle 7.0 on AIX 3.2.5.

   Notes:

   a. Use the Appendix C of the CMVC 2.3 Server manual.

   b. There is no support in CMVC 2.3 for Oracle 6.

2. Migrate from CMVC 2.1/2.2 Oracle 7.0 on AIX 3.2.5 to CMVC 2.3 Oracle 7.0 on AIX 3.2.5.

   This is very easy. See Appendix B of the CMVC 2.3 Server manual.

3. Migrate from CMVC 2.3 Oracle 7.0 on AIX 3.2.5 to the latest version of CMVC 2.3 (2.3.0.22) Oracle 7.1 on AIX 3.2.5.
4. Migrate from CMVC 2.3.0.22 Oracle 7.1 on AIX 3.2.5 to CMVC 2.3.0.22 Oracle 7.1 on AIX 4.1.

DO NOT MIGRATE TO ORACLE 7.3 BECAUSE IT IS NOT SUPPORTED

We have learned that Oracle 7.3 is significantly different than Oracle 7.2/7.1/7.0, and many APIs used by CMVC are not available in 7.3. Because of this and the upcoming replacement of CMVC by its successor product, TeamConnection, it is unlikely that Oracle 7.3 will be supported by CMVC.

Thus, do not try to migrate CMVC to Oracle 7.3.

MIGRATING CMVC FROM INFORMIX TO DB2

Here is how we recommend making the migration from Informix on AIX 3.2 to DB2 on AIX 4. As in previous examples, we will call the current machine Host A and the new machine Host B.

1. Perform a complete backup of the current CMVC family database (under Informix) and file system on Host A. See the Informix documentation for specific directions.
2. Migrate to DB2 on Host A. However, do not migrate CMVC or AIX on Host A yet.

   Install DB2 and then migrate the database from Informix to DB2 as explained in the CMVC Server manual, Appendix D.
3. Verify that the CMVC family is working correctly under DB2 on Host A.
4. Make a complete backup of the CMVC family database (under DB2) and file system on Host A.
5. Migrate from Host A running DB2 on AIX 3.2.5 to Host B running DB2 on AIX 4.1.
6. Verify that the CMVC family is working correctly under DB2 on Host B.
7. Make a complete backup of the CMVC family database (under DB2) and file system on Host B (now under AIX 4).

0010-256 ERROR MESSAGES AFTER MIGRATING FROM CMVC V1.1 TO V2.X

There is a potential problem after migrating a CMVC family from V1.x to V2.x in which the command "Defect -configInfo" will not work (error message 0010-256 for txDefectConfig in the client side, and error 0010-636 with a database error indicating more than 1 row).

It might be possible in CMVC 1.x to have multiple defaults in one configuration item type (from config.ld) which were loaded into the Config table and this is the root problem, because in CMVC 2.2, if there is a default, it should be only 1 and no more.

In CMVC 1.1, there was no checking for this situation, and the migration step does not touch this table and does not reload it fresh from the source config.ld file.

Fixing CMVC V1.1 to V2.x migration problems with "chcfg"

Immediately after the migration is over, but before starting the CMVC daemons, it is necessary to update the file config.ld to avoid multiple defaults for an item, and then to use "chcfg" to reload the items for the Config table (which must be successful).

PREPARING TO MIGRATE TO TEAMCONNECTION

CMVC and it successor product TeamConnection have a lot in common but they are not compatible between each other. That is, you cannot use a CMVC client with a TeamConnection server, nor a TeamConnection client with a CMVC server.

This means that if you want to use TeamConnection, then you have to migrate your CMVC family into TeamConnection.

WHY TEAMCONNECTION?

CMVC is an industrial strength source code library with extensive support for tracking of changes through defects and features, and a configurable and extendible process model (by using user exits). However, the role of the development library is evolving.

TeamConnection is IBM's next generation of library management. Actually, TeamConnection is much more than a library. TeamConnection extends the CMVC process model to include a tightly integrated build facility, a highly customizable packaging facility, and support for network distribution of your
packaged code.

Furthermore, TeamConnection incorporates an object repository and API's to allow tools to store their data (that is, large grained objects such as files and small grained objects such as data field information created by 4GL languages like IBM's VisualAge Generator in TeamConnection.

These integrated tools can take advantage of TeamConnection's releases, defects, features, etc. so that all of the developer's data is managed and versioned together. This dramatically reduces the overhead of using a collection of tools in the development process.

Finally, the object repository interface allows for tools used by anyone in your entire development team (document writers, marketing specialists, testers, etc.) to manage and version their data in one place.

Now everyone's data can evolve without lots of manual translation. For example, a software architect's specification can be tracked as the software developer writes code to match the specifications. This transition is trackable and auditable.

MIGRATION FROM CMVC TO TEAMCONNECTION

The TeamConnection migration utility will use CMVC client commands. The only requirement from TeamConnection is that the CMVC client be installed on the same system as the TeamConnection server for a migration to occur.

In principle, any CMVC client, version 2.2 or later, will be able to support TeamConnection migration. However, we recommend that you upgrade your version of the CMVC server to the latest supported version of CMVC 2.3 before migrating to TeamConnection.

The following shell script samples will help you to prepare the CMVC family for migration:

“Complete a release prior to TeamConnection migration” on page 30
“Reassign user's work and delete users” on page 30
“Delete a release from a family” on page 30

MIGRATION SHELL SCRIPTS

OBTAINING THE SHELL SCRIPTS

The shell scripts 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://keithp.raleigh.ibm.com/~cmvcsupt

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

FTP

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

1. ftp keithp.raleigh.ibm.com
2. login as 'anonymous' and for password give your email address.
3. cd pub/cmvc/fixes/tr-tools
4. ascii
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/tr-tools
4. ascii
5. get <filename>
6. quit

SHELL SCRIPTS TO PREPARE THE MIGRATION FROM CMVC TO TEAMCONNECTION

Complete a release prior to TeamConnection migration

The name of the shell script is:

release.complete.ksh

This utility will complete work in a given release so that all file changes are committed.

Reassign user's work and delete users

The name of the shell script is:

reassign.work.ksh

This utility will reassign incomplete work from one active CMVC login to another active CMVC login. Incomplete work can be defined as:

defects owned/originated
features owned/originated
verification
components
releases
levels
environments
test records
size records
approval
approver
tracks
fix records
notifications deleted
access deleted
login deleted

Delete a release from a family

The name of the shell script is:

release.delete.ksh

This utility will delete work in a given release. It will also identify a list of vc (SCCS/PVCS) files that are not related to any other release that can be removed to save disk space.

MISCELLANEOUS SHELL SCRIPTS

Import a release from directory structure

The name of the shell script is:

file.import.ksh

This utility will load files into CMVC from a given root directory. It requires that the user should provide a mapfile that maps the pathnames of the files to the components where the files are to be loaded into.

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, OS/2, AIX, VisualAge Generator, CMVC, TeamConnection	IBM Corporation
UNIX, USL	UNIX System Laboratories, Inc.
OSF, OSF Motif	Open Software Foundation, Inc.
INFORMIX	Informix Inc.
ORACLE	Oracle Corp.
SYBASE	Sybase Inc.
Sun, SunOS, OpenWindows, Solaris	Sun Microsystems Inc.
HP, HP-UX	Hewlett-Packard Company

END OF DOCUMENT