

Migrating a CMVC or VisualAge TeamConnection family to ClearCase/ClearQuest

Mark D. Dunn, Angel Rivera, Kevin Postreich

TR 29.3470

Draft: 16-Oct-2001

DISCLAIMER:

This technical report is not an official publication from the VisualAge TeamConnection support group. The authors are solely responsible for its contents.

IBM Software Solutions
Research Triangle Park, North Carolina, USA

© Copyright 2001, IBM Corp.

Migration from CMVC/TeamConnection to CC/CQ

ABSTRACT

This technical report provides a procedure and tools that aid in the migration of an IBM VisualAge TeamConnection or IBM CMVC family to Rational ClearCase/ClearQuest. The objective is to provide a structured process, to reduce errors and to save time by selecting only appropriate data for the migration.

ITIRC KEYWORDS

- Migration
- VisualAge TeamConnection
- CMVC
- ClearCase
- ClearQuest

ABOUT THE AUTHORS

Mark D. Dunn

Mr. Dunn is a advisory software engineer, formerly with the VisualAge TeamConnection development group. He rejoined IBM in 1996 and was in charge of development and support of TeamConnection migration. He is currently a developer with the Test Tools Automation group.

Mr. Dunn has a B.S. degree in Biology from Western Kentucky University and a B.A. degree in Computer and Information Systems from Florida Atlantic University.

Angel Rivera

Mr. Rivera is an advisory software engineer and is the serviceability architect and RAS focal point for IBM e-Business Tools.

He joined IBM in 1989 and since then had worked in the development and support of library systems. He was the technical lead for CMVC Version 2 for many years. He was also the team lead of the technical support team for VisualAge TeamConnection.

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

Kevin Postreich

Mr. Postreich is a staff software engineer with the IBM WebSphere group. He joined IBM in 1980 as an electronic engineer in Charlotte, North Carolina. He relocated to RTP as an MVS systems programmer in 1987. Later, he joined the VisualAge TeamConnection development/test group and was a member of the customer support team specializing in database migrations.

Mr. Postreich is currently a member of the WebSphere Studio development and level 3 customer support teams.

Migration from CMVC/TeamConnection to CC/CQ

ABSTRACT	- 3 -
ITIRC KEYWORDS	- 3 -
ABOUT THE AUTHORS	- 4 -
Introduction	- 9 -
Overview of the migration tool	- 9 -
Important migration caveats	- 10 -
Restrictions on the CMVC client/server	- 10 -
Disclaimers	- 10 -
Acknowledgments	- 10 -
How to get the most up to date version of this technical report	- 11 -
How to get the most up to date version of the migration tools	- 11 -
Overview of the migration process	- 12 -
General suggestions	- 12 -
Notes for practice migrations	- 12 -
Preparing the migration plan	- 13 -
Important assumptions about the migration	- 13 -
Important recommendations on the scope of the migration of Files	- 13 -
Planning considerations	- 13 -
Time considerations	- 15 -
Important planning note when importing parts:	- 15 -
Day by Day Plan	- 16 -
Day: CC/CQ - 1 day (family servers)	- 16 -
Day: CC/CQ - 1 day (clients)	- 16 -
Day: CC/CQ - 1 day (users and usability)	- 16 -
Day: CC/CQ - 1 day (test migration)	- 17 -
Day: CC/CQ (server side)	- 17 -
Day: CC/CQ (client side for the test migration)	- 17 -

Migration from CMVC/TeamConnection to CC/CQ

Day: CC/CQ + 1 day (server side)	17 -
Prepare the source family for the migration	18 -
Checklist for preparing the source family for migration	18 -
What to do to the source family before migration	19 -
CMVC note for migrating files: need to create additional table with Committed Levels	19 -
Prepare the target CC/CQ	20 -
Checklist for creating a new CC/CQ	20 -
Environment setup for running clearexport	21 -
Performing the migration	22 -
Checklist for migrating the source family to the target CC/CQ	22 -
Performing the actual migration and logging results	22 -
Migrating users, defects and features to ClearQuest	22 -
Exporting data for ClearQuest into an export file	22 -
Importing data into ClearQuest from an export file	24 -
Importing users	24 -
Importing defects or features	25 -
Migrating parts into ClearCase	26 -
Exporting parts with version history for ClearCase into an export file	26 -
Example of clearexport_tc	29 -
Notes on creating a partspec file	30 -
Exporting parts without version history for ClearCase into an export file	31 -
Step 1. Ensure that the target directory is empty.	31 -
Step 2. Extract the current release from TeamConnection/CMVC.	31 -
Step 3. Run the ClearCase clearexport_ffile utility.	31 -
Importing parts into ClearCase	33 -
Step 1. Running the ClearCase clearimport utility on UNIX.	33 -
Step 1-A. Running the ClearCase clearimport utility on Windows NT.	34 -
Step 2. Adding Labels to the Imported Directories	35 -

Migration from CMVC/TeamConnection to CC/CQ

Hints and tips for migrating the data	- 35 -
Take note of the time that it took to perform the test migrations	- 35 -
ClearExportGui: Java-based GUI for Windows users	- 36 -
Invoking the tool	- 36 -
Usage notes	- 36 -
Exporting Users, Defects, Features	- 37 -
Exporting Files (Parts)	- 38 -
Verification of the new CC/CQ environment	- 40 -
Checklist for verifying the new CC/CQ environment	- 40 -
Post-migration tasks for the new CC/CQ environment	- 41 -
Checklist for performing post-migration tasks for the new CC/CQ environment	- 41 -
Appendix A: Mappings of defects, features and parts	- 42 -
Defect and feature mapping	- 42 -
Appendix B: Getting the correct versions of parts	- 44 -
Getting the correct versions of parts from TeamConnection	- 44 -
Getting the correct versions of files from CMVC	- 45 -
Appendix C: How to get the files mentioned in this document	- 46 -
FTP site for VisualAge TeamConnection	- 46 -
Obtaining Info-ZIP	- 46 -
How to unzip files	- 46 -
Brief description of the migration tools	- 47 -
VA TC migration tool	- 47 -
CMVC migration tool	- 47 -
Front end GUI for Windows	- 47 -
CMVC Korn shell scripts to prepare the table with file versions in committed Levels ...	- 47 -

Migration from CMVC/TeamConnection to CC/CQ

Scripts to facilitate the query of DBMSs	- 47 -
Miscellaneous tools	- 47 -
Copyrights, Trademarks and Service marks	- 48 -

Migration from CMVC/TeamConnection to CC/CQ

Introduction

This technical report provides a procedure and tools that aid in the migration of an IBM VisualAge TeamConnection or IBM CMVC family to Rational ClearCase/ClearQuest. The objective is to provide a structured process, to reduce errors and to save time by selecting only appropriate data for the migration.

The objectives of this technical report are:

- To describe a structured process.
- To provide checklists to help you execute each migration step.
- To provide tools for migration and hints on how to use them.
- To reduce errors and to save time by selecting only appropriate data for the migration.
- To provide a tool is provided to help you verify the success of the migration.

Appendix A provides the mapping between the main attributes of objects from TeamConnection/CMVC and CC/CQ.

Appendix B provides some technical explanations on how the correct versions of the parts/files from the source family are obtained by the migration tools.

Finally, we provide the details on the tools used in this document and how to get them (see *How to get the files mentioned in this document* at the end of this document).

Overview of the migration tool

The migration tools will help migrate the following four types of objects from TeamConnection or CMVC:

- Users
- Defects
- Features
- Parts/Files (either the latest version or with version history)

For each type of object, this tool creates export files which are imported into ClearCase/ClearQuest using the import tools available from Rational. Users, Defects and Features are imported into ClearQuest; Parts/Files are imported into ClearCase.

The migration tool is very flexible and allows the person performing the migration to specify which parts/files and which releases to export part/file versions from, and subsequently import in a VOB in ClearCase.

Migration from CMVC/TeamConnection to CC/CQ

It also allows for incremental migration of parts/files. This would allow the customer to export part/file versions, import them into ClearCase, run tests over several weeks, then to export only the changes since the first migration, and import them into ClearCase.

Important migration caveats

In some instances, ClearCase and ClearQuest do not support the data structures found in TeamConnection/CMVC. Consequently, before migrating your data, you should be aware of the following migration limitations:

- The ClearCase import utility does not convert directory versions. However, after your files are migrated, you can use ClearCase to version directories.
- The migration of deleted files is not supported.
- When you migrate renamed files, both the new and old names will have the release labels attached to them.
- You cannot migrate the entire context of a driver/level. You can only migrate the changed parts/files in a driver.

Restrictions on the CMVC client/server

The CMVC Report -general command is used by the migration utilities, and this function was added to some client/servers in version 2.3.1.2. The following are valid platforms:

- CMVC client for AIX, HP-UX and Solaris
- CMVC server for AIX using DB2, Informix, or Oracle.
- CMVC server for HP-UX using Informix

The following are not valid platforms:

- CMVC client for Windows
- CMVC server for AIX using Sybase
- CMVC server for HP-UX using Oracle
- CMVC server for Solaris using Oracle or Sybase

Disclaimers

* This technical report is not an official publication from the VisualAge TeamConnection or CMVC support groups. The authors are solely responsible for its contents.

* The tools mentioned in this technical report are provided “as-is” with no warranties of any kind. Furthermore, these tools are NOT supported by IBM. Use the tools at your own risk.

Acknowledgments

Migration from CMVC/TeamConnection to CC/CQ

We would like to acknowledge the contribution of the following co-worker:

- Bruce Korn, who assisted with the Migration Users Guide

How to get the most up to date version of this technical report

The most up to date version of this technical report can be obtained from the following IBM ftp site:

[Ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/trmig-cccq.pdf](ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/trmig-cccq.pdf)

How to get the most up to date version of the migration tools

To get the most up to date version of the migration tools (remember: they are provided “as-is”), see the Appendix C: How to get the files mentioned in this document.

Overview of the migration process

General suggestions

Here is a broad overview of the steps needed to migrate a VisualAge TeamConnection or CMVC family to ClearCase/ClearQuest (CC/CQ). The ordering of the chapters in this technical report reflect this migration sequence:

1. Prepare the migration plan.
 - We strongly suggest a rigorous plan for migrating families to ClearCase/ClearQuest that requires the participation of your entire team.
 - Train users on CC/CQ prior to migration. Key differences between TeamConnection and these products are identified in the following URL:
http://www.rational.com/products/clearcase/cc_table.jsp
 - For more information on the IBM/Rational Alliance for Software Configuration Management (SCM) tools, and what are some migration considerations, see the following URL:
<http://www.rational.com/products/clearcase/ccibm-faq.jsp>
2. Gather all materials needed for planning and migration.
3. Prepare the VisualAge TeamConnection or CMVC family for migration (source family).
4. Prepare the new CC/CQ environment (target environment), in order to have a place for the data to be migrated from the source family.
5. Perform the migration. It is recommended that you perform a practice migration before the real migration.
6. Verify the new target environment.
7. Perform post-migration tasks on the new family.

Notes for practice migrations

- Your team should extensively exercise your new target environment after migrating from the source family for a very limited period of time, such as one full day.
- At the end of testing, have a quick meeting to assess the results and refine the plan for the final migration.
- If the practice migration went fine, then discard this testing target environment.
- Finally, perform the migration again for real, and start using your new target environment.

Preparing the migration plan

This chapter provides several checklists that can be used to determine the progress of the migration process.

Notes:

- The Estimated Time for those tasks that take more than 5 minutes has been rounded up to the nearest ½ hour.
- It assumes that all the prerequisites have been completed.

	Description of task	Est. Time	Comment
	Preliminary planning activities.	½ day	See the <i>Planning Considerations</i> in this chapter.
	Prepare day-to-day plan.	½ day	See <i>Day by Day Plan</i> in this chapter

Important assumptions about the migration

Throughout this document, we will assume the following:

- You should plan to do practice migrations because there are things you need to find out for yourself.
- You should include in your plan enough time to do backups of the source VisualAge TeamConnection Version or CMVC family and of the target CC/CQ environment at several points in the migration (such as just before the migration, after all the data except the files have been migrated, and after all the files have been migrated).

Important recommendations on the scope of the migration of Files

When migrating Files, we recommend to ONLY migrate the latest committed version of files in the release. This method reduces the migration time considerably, and makes for a simpler and less cumbersome migration.

Planning considerations

Moving your data from a VisualAge TeamConnection or CMVC family to CC/CQ will require careful planning. These are the questions you need to ask yourself:

- Will all of the releases from my current source family be migrated, or only some of them?

Migration from CMVC/TeamConnection to CC/CQ

- If not all of the releases will be migrated, are there any releases that need to be kept in the source family as reference only, while other releases will be migrated to be used in future development?
- How important is the history of each file change? Could the history of each file change be left in the source family (which means that this family needs to be active) as a reference family and migrate only the latest file change to CC/CQ? If only the last file change will be migrated, then migrating only the latest file change will save time during the migration and disk space with the new target environment.
- You need to determine for the migration of Defects/Features, determining whether to use the default mapping and using the standard schema which comes with ClearQuest, or to select no mapping to take place and create your own schema/forms in ClearQuest.
- You need to determine for the migration of Parts/Files, how to divide your source code trees into multiple VOBs in ClearCase.
- You need to estimate the time need for the migration and finding a window that is appropriate for your team. The source TeamConnection/CMVC family should not be updated during the export of the specified parts/files and releases. Although the export occurs relatively quickly (approximately 1-2 seconds per part/file exported), the Import into ClearCase takes approximately 5-15 seconds per version of part/file.
- You need to take into account that the entire context of a driver/level can not be viewed collectively once migrated into ClearCase. Only the parts/files changed in a context can be viewed. One Customer worked around this by exporting each of the important Levels separately and importing them into separate VOBs.
- Is there enough disk space to keep both the source family and the new environment in CC/CQ at the same time in the same host?
- Is there enough CPU to run both the source family and the new environment in CC/CQ at the same time in the same host? You need to take into account the resources that are needed to run TeamConnection, CMVC and the database management system such DB2 Universal Database, and CC/CQ.
- Decide how you are going to stage your migration from the source family to CC/CQ: if you have multiple releases or even multiple families, then the migration may take more than one day of work, so you need to be ready to run both the source family and CC/CQ in parallel while all the families are being migrated.
- If the source family needs to be active for normal work after the migration, then it might be necessary to make a clone of the source family and migrate it to version CC/CQ from a cloned family. In this case, we recommend that the source family should be taken off-line during the cloning process.

Migration from CMVC/TeamConnection to CC/CQ

After the cloning process is done, both families (production and clone) can be started. In that way critical work could be done in the production family while the cloned family is used in the migration process (and it is not accessible to normal users of the production family).

NOTE: Any work done in the production family after the cloning will not be migrated during the migration process described in this document and it must be migrated manually later on.

- After the migration, the family administrator could create a user exit for update actions to indicate that the family is only available in read-only mode and that any update operations need to be done in CC/CQ. One example of such user exit is shown below.

```
#!/usr/bin/ksh
# Name: generic
RC=0
COUNT=`print ${*} | grep -c v3`
if [ ${COUNT} -gt 0 ]
then RC=1
    print ""
    print -u2 "<<STATUS>> please use CC/CQ."
fi
exit ${RC}
```

This generic user exit is defined \$TC_HOME/config/userExit for all update actions as follows (this is just one example):

```
WorkAreaUndo    0          generic
```

The answers to these questions are critical because they can help you determine whether you will need to:

- Clone any TeamConnection/CMVC families from your current host to another host that has more resources and/or you want to continue working for a short period of time with some releases in the source family after migration.
- Add disk space, move around disk space, etc.
- Limit the amount of data to be migrated to save time or disk space; for example, get a snapshot of a release instead of the entire file change history.

Time considerations

Time spent migrating your families is time that your families are down, therefore the less data you migrate the less down time for your families. This is one of the best reasons to migrate only the releases you plan to continue developing. Leaving the other releases in the source family will get your target CC/CQ up and available much faster.

In order to save time and effort, once you have migrated the active releases, put the source family in read-only mode and remove it from your backup list. There is no reason to backup a family that does not change.

Migration from CMVC/TeamConnection to CC/CQ

Important planning note when importing parts:

Unix: The step to import parts was tested on a 2 way SMP AIX machine with 1 gigabyte of RAM and two 266 MHz processors. It took approximately 8 hours to import 8400 parts. The time it will take for you to complete this step will vary based on number of parts and versions you import and can take up to 15 seconds per version. This can translate into several days of processing time for large systems.

Windows NT: The step to import parts was tested on a Windows NT machine with a 400 MHz processor and 128M of RAM. It took approximately 8 hours to import 8400 parts. The time it will take for you to complete this step will vary based on number of parts and versions you import and can take up to 15 seconds per version. This can translate into several days of processing time for large systems.

Day by Day Plan

You have to begin by selecting an appropriate day in which all the members of your team will test the migrated family. Select an appropriate target date and call it **CC/CQ day**. Based on that target date there are several activities that need to be done prior, during and after that date.

Day: CC/CQ - 1 day (family servers)

Although we assume that the target CC/CQ is installed on a different host than the source family, it is possible to have both versions on the same host. However, we recommend using different hosts. Running both versions of the product on the same host will result in conflicts if the proper environment settings are not carefully maintained.

Source system:

- Ensure that the source family is up and running.

Target system:

- Install CC/CQ.
- Install the corresponding CMVC or VisualAge TeamConnection client.
- Get the migration tools mentioned in this document. See the Appendix C *“How to get the files mentioned in this document”* at the end of this document.
- Setup the proper environment variables for the migration process.
- Plan the backup and restore strategy for your new CC/CQ.

Day: CC/CQ - 1 day (clients)

- Ensure that everybody in the team has installed the CC/CQ clients.
- (Optional) Remove the client for TeamConnection/CMVC.

Day: CC/CQ - 1 day (users and usability)

Migration from CMVC/TeamConnection to CC/CQ

- Schedule and deliver training sessions to provide the essentials of how to use the new CC/CQ.

Day: CC/CQ - 1 day (test migration)

- Migrate the latest snapshot of the source family into the new test target CC/CQ.
- The new target CC/CQ is started and a migration test is done by selected team members to ensure that so far the migration went fine and that the family is ready for CC/CQ day.
- Rehearse the backup and restore strategy for the new test family.
- All team members ensure that they are ready for CC/CQ day: they should be able to connect to the new target CC/CQ from the connectivity point of view.

Day: CC/CQ (server side)

- The target CC/CQ administrator and selected members of the team should monitor the new environment, paying particular attention to not running out of disk space.
- At the end of the day, the target CC/CQ will be shut down in order to perform a backup and then do a restore, to test your backup/restore strategy.

Day: CC/CQ (client side for the test migration)

- Every member of the team will spend time interacting with the test target CC/CQ. First with the specific assignments and then later on in an exploratory mode.
- Once there are enough parts/files migrated, you should try to perform a "normal" build.
- At the end of the day, you should have a short meeting to discuss the results and if necessary, plan to fix the main problems and if necessary repeat the test.
- Once you and your team are comfortable with the results of the test migration, proceed to perform the real migration.

Day: CC/CQ + 1 day (server side)

- The test CC/CQ should be deleted in order to prepare the next test (if needed) or the final migration.
- Perform the final migration for the source family by repeating the tasks described in **Day: CC/CQ**. Keep in mind that this time the migration is not a test anymore; it is the final migration.
- Start the new CC/CQ.
- Perform the verification of the new CC/CQ.

Prepare the source family for the migration

Checklist for preparing the source family for migration

Source system: These activities are done in the host system where the source TeamConnection or CMVC family is located. This system will be called the **source system**.

Description of task	Est. Time	Comments
Backup the source family database.	1 hr	See VisualAge TeamConnection Administrator's Guide or the corresponding DBMS manuals.
Restart the VA TC source family by running: teamcd <i>familyName N</i> Restart the CMVC source family by running: cmvcd <i>familyName N</i> Where <i>N</i> is the number of daemons (at least 2).	5 min	The family should be active while migrating.
Run the script processCommittedLevelsForRelease to create table with committed versions of the files in a level.	1 hr	A new table is needed for migration purposes.
Disable the automated shell scripts that might affect the availability of the source family or the database system. This is to prevent a possible interruption of the migration.	½ hr	For example, if you start the migration in the evening, it may take several hours to migrate the parts. Thus, you want to avoid an automatic daily backup cron job at midnight to interrupt the migration.
(Optional) Make changes to the naming conventions of files.	30 min	Take advantage of the transition to the new family to improve the usage of TeamConnection.
Add a host list for a TeamConnection/CMVC superuser who will issue client commands from the new machine.	5 min	
Ensure that all parts are unlocked.	30 min	The parts must be unlocked for migration.

Migration from CMVC/TeamConnection to CC/CO

What to do to the source family before migration

There are many preparation tasks that can be performed in the source family prior to the migration:

- You can take advantage of this transition to make changes to the naming conventions.
- There are things that are not migrated but that should be cleaned up before migration. For example, ensure that all parts are unlocked (keep a note of the parts that you need to unlock, in order to notify the users who had a lock on those parts).

CMVC note for migrating files: need to create additional table with Committed Levels

CMVC keeps the information of the particular versions of the files that are in a committed level in a file named a “map file” which is located in a directory structure which is based on the release and level name. That is, this information is NOT stored in the database for the family. This means that this information cannot be accessed by a CMVC client.

In order to facilitate the migration of files from CMVC, it is important to run the script **processCommittedLevelsForRelease** and **createCommittedVersions** (which is included with the migration utilities) which will create a table in the CMVC family, read the map file for the desired release-levels and populate the table with the appropriate information.

Once the table is created and populated, the client can use the Report -general using this new table and get the appropriate version of the committed files for a level in order to extract them.

This table is ONLY exploited by the migration utilities. The CMVC server does not exploit this table at all. Thus, if you commit a level after running this script, then you will need to run the script again in order for the table to reflect the new committed level.

To migrate files from CMVC, do the following:

- A) Login as a CMVC family superuser id.
- B) Copy the Korn Shell scripts called `processCommittedLevelsForRelease` and `createCommittedVersions` to a directory in your path, such as `$HOME/bin`.
- C) Execute the following Korn Shell Script:

```
processCommittedLevelsForRelease db2 <release name>
```

Note that you must execute `processCommittedLevelsForRelease` for each release you want to export.

Migration from CMVC/TeamConnection to CC/CQ

Prepare the target CC/CQ

Checklist for creating a new CC/CQ

Target system: These activities are done in the host system where the new CC/CQ will be located. This system will be called the **target system**. There are some entries in the table that do not have an estimated time; the reason is that there is a wide range of possibilities and we rather do not specify a time that might be misleading to the reader.

Description of task	Est. Time	Comment
Install the software for CC and for CQ.		See the corresponding installation manuals.
Perform the appropriate administration and configuration tasks for CC/CQ.		See the corresponding manuals.
Install the TeamConnection or CMVC client.	1 hr	The clearexport utility will make use of TeamConnection or CMVC client commands.
Ensure that the source family can be reached. For TeamConnection, use: teamc report -testServer -family sourceFamily where <i>sourceFamily</i> is the v2.0.9 family You should see a message saying that the family server is active. For CMVC, use: Report -testServer -family sourceFamily	5 min	If this command fails, then clearexport will fail too. It is better to fix the problem now.
Make a backup of the new CC/CQ environment. It is faster to restore a newly created CC/CQ environment from a backup than to recreate it.	½ hr	See the corresponding administration manuals.
Setup the environment variables needed for the TeamConnection migration utilities.	10 min	

The migration utility, **clearexport**, will use VisualAge TeamConnection/CMVC client commands. It is important to have the proper version of the client installed in the target machine:

- The VisualAge TeamConnection client Version 3.0.3 or higher.
- The CMVC client Version 2.3.1.2 or higher.

Migration from CMVC/TeamConnection to CC/CQ

Environment setup for running clearexport

Before running **clearexport** it is necessary to setup the following environment variables (shown here in Korn shell format for Unix):

1. Login to the new CC/CQ environment that will be the target for the migration.
2. If using TeamConnection, update the environment:
 - `export TC_FAMILY=targetFamilyName`
 - `export TC_BECOME=superUserId`
1. If using CMVC, update the environment:
 - `export CMVC_FAMILY=targetFamilyName`
 - `export CMVC_BECOME=superUserId`

Performing the migration

Checklist for migrating the source family to the target CC/CQ

Target system: These activities are done in the host system where the new CC/CQ environment will be located. This system will be called the **target system**.

	Description of task	Est. Time	Comment
	Disable the automated shell scripts that might affect the availability of the target CC/CQ. This is to prevent a possible interruption of the migration.	½ hr	For example, if you start the migration in the evening, it may take several hours to migrate the parts. Thus, you want to avoid an automatic daily backup cron job at midnight to interrupt the migration.
	Run clearexport to export users, defects and features from TeamConnection/CMVC into export files.		
	Import the users, defects and features from the export files into ClearQuest.		

Performing the actual migration and logging results

The `clearexport*` tool creates the export files from CMVC or VA TC. However, the real migration is done using the CC/CQ commands; during this step is where the logging takes place.

For users of Windows platforms, we provide an optional Java-based GUI tool, named “ClearExportGui” which can be used as a front-end to specify the desired options to the `clearexport*`. The GUI tool will then invoke the export file and drive the migration steps. For more information, see the chapter “ClearExportGui: Java-based GUI for Windows users”.

Migrating users, defects and features to ClearQuest

Migrating users, defects, and features from TeamConnection/CMVC to ClearQuest is a two step process:

1. Export the data from TeamConnection/CMVC into 3 different export files.
2. Import the data from these export files into ClearQuest.

Exporting data for ClearQuest into an export file

Migration from CMVC/TeamConnection to CC/CQ

You can migrate users, defects and features using the **clearexport_tc** tool (for TeamConnection data) or the **clearexport_cmvc** tool (for CMVC data). Before using either of these tools, note the following:

The command syntax for `clearexport_tc` and `clearexport_cmvc` is as follows:

```
clearexport_tc (or clearexport_cmvc)
  <-users | -defects | -features>
  [-o outputFile][-where sqlquery]
  [-map | -nomap]
```

where:

- `-users | -defects | -features` is the data you are exporting
- `outputFile` is the name of the output file (this is used by the ClearQuest import utility)
- `sqlquery` is the SQL query string to the TeamConnection/CMVC server
- `-map` or `-nomap` is a switch that turns on TeamConnection (or CMVC) mapping.

For example you could export all your TeamConnection users by running the following command:

```
clearexport_tc -users
```

Or, you could export all users in a domain called `ny.myco.com` into a file called `myusers.txt` by running the following command (all on the same line):

```
clearexport_tc -users -o myusers.txt -where "address like '%ny.myco.com'"
```

After creating an output file for users, you then run the `clearexport` tool to create an output file for defects and then an output file for features.

If you do not specify the name of an output file, the following defaults are used:

<u>Parameter</u>	<u>Default Output File</u>
<code>-users</code>	<code>usersExport.txt</code>
<code>-defects</code>	<code>defectsExport.txt</code>
<code>-features</code>	<code>featuresExport.txt</code>

If you do not specify a value for the `-where` switch, the following defaults are used:

<u>Parameter</u>	<u>Value for -where</u>
<code>-users</code>	<code>"1=1 and dropDate is null"</code>
<code>-defect</code>	<code>"1=1 order by name"</code>
<code>-features</code>	<code>"1=1 order by name"</code>

Migration from CMVC/TeamConnection to CC/CQ

By default, the export command does not export deleted users. If you want to migrate deleted users, use the following command when you export users:

```
clearexport_tc -users -where "1=1"
```

If you do not specify the `-map` or `-nomap` parameter, the command defaults to `-map`. When mapping occurs, TeamConnection/CMVC fields are mapped to the ClearQuest DefectTracking schema fields. For more details on the mapping, see the Appendix A “Mappings of defects, features and parts”.

If you specify the `-nomap` option, all fields found in the defect or feature will be written to the export file without mapping. However, you would have to customize a ClearQuest schema to accept this type of data (see ClearQuest documentation for information on customizing schemas). All configurable field data is migrated when specifying the `-nomap` option.

Importing data into ClearQuest from an export file

This section instructs you how to import users, defects and features into ClearQuest using the export files that you created when you exported data from TeamConnection/CMVC.

Note that because the ClearQuest fields of “submitter” and “owner” reference users defined in ClearQuest, you must first import the users from TeamConnection/CMVC before importing defects or features. If you attempt to import defects or features before importing users, you will get an error. This also applies to those defects that reference deleted users.

Importing users

To import users, complete the following steps:

1. Start the ClearQuest Designer.
2. From the Tools pull-down, select User Administration.
3. Select the Import push button.
4. Specify the name of the output file that contains your user data and then select the Open push button.

When importing users, please note the following:

- All users are imported with access to all databases.
- Deleted users are not migrated.
- There is no such thing as a deleted users in ClearQuest. If you elect to migrate deleted users they will become valid users when imported into ClearQuest.

TeamConnection and CMVC superusers are imported into CQ with the following attributes:

Migration from CMVC/TeamConnection to CC/CQ

is_superuser = TRUE
is_appbuilder = TRUE
is_packagehacker = FALSE
is_user_maint = TRUE

All other users are imported as follows:

is_superuser = FALSE
is_appbuilder = FALSE
is_packagehacker = FALSE
is_user_maint = FALSE

Importing defects or features

Before importing defects or features, make sure you have first imported users.

It is important to understand the following about importing features:

- TeamConnection and CMVC features are imported as severity 5 defects.
- The default DefectTracking ClearQuest schema defines a severity 5 defect as an enhancement request (which is the equivalent of a TeamConnection or CMVC feature)

To import defects or features, start the ClearCase Import Tool and complete the following steps:

1. In the Setup Import Files window, specify the name of your export file and the name of an error log file in the appropriate fields. Then select the Next push button.
2. In the Setup Destination window, do the following:
 - Select "Defect" in the Select Destination Entity field.
 - Select "State" in the Imported field with state information field.
 - Select "old_id" in the ClearQuest field containing original id field.
 - Make sure the Import duplicates check box is not checked.
 - Select the Next push button.
3. In the Setup Field Mapping window, select the Next push button. (Do not make any changes in any of the fields.)
4. In the Perform Import window, select the Import push button.

Migration from CMVC/TeamConnection to CC/CO

Migrating parts into ClearCase

There are two ways to migrate parts into ClearCase. You can:

- Migrate parts with version history
- Migrate parts without version history

Important: You can only migrate parts with version history if you use the CMVC server for AIX using DB2, Informix or Oracle; or the CMVC server for HP-UX using Informix. For other CMVC servers you must migrate parts without version history. See the section, "Exporting parts without versions history for ClearCase into an export file" for instructions.

Note that after the parts are migrated, you will not be able to view an entire driver/level; you will only be able to view the parts that were changed in a driver/level.

Exporting parts with version history for ClearCase into an export file

To migrate parts with version history into ClearCase, you first export part information from TeamConnection/CMVC into an export file. This export file contains an index of the parts that are to be extracted from TeamConnection/CMVC. You then use this file as input to the ClearCase import utility, which extracts the parts from TeamConnection/CMVC and imports them into ClearCase. Consequently, you need to have a TeamConnection or CMVC client running for the export and import phase of the migration.

The tools you use to export parts from TeamConnection/CMVC to an export file are named **clearexport_tc** (for TeamConnection) and **clearexport_cmvc** (for CMVC). To ensure performance, these tools export only the latest part versions committed to a release or driver (or CMVC level). This includes all release level versions of the part. Consequently, the tools will not export:

- Workarea versions.
- Parts that are checked out.
- Configurable fields on parts.
- Metadata for the parts (such as remarks or change history that links parts to defects).

After creating an export file, you then use this file as input to ClearCase's **clearimport** tool, which extracts part information from TeamConnection/CMVC and imports the parts into different ClearCase VOBs (Versioned Object Bases). The instructions for using the ClearCase import tool are provided in the section entitled "Importing parts into ClearCase".

You should not attempt to import export files that contain identical parts and part versions into the same VOB; if you do, you will get an error message. The ClearCase import tool expects part versions to be in chronological order. If the same part appears in two export files, and the change date on the version

Migration from CMVC/TeamConnection to CC/CO

you are importing is earlier than the change date on a version previously migrated, an error will occur. Otherwise, newer versions are added either to the end of the mainline branch for each part or to a separate branch where appropriate.

When migrating, you must specify the release or releases you want parts migrated from. If you want parts from only one release, then you must specify that release on the command line:

1. Place the release names in a file
2. Use the "-relspec" command line option
3. Point to the release specification file when invoking `clearexport_tc` (or `clearexport_cmvc`).

You can specify a list of parts that you want migrated from each specified release by doing the following:

1. Create a list of parts in a file
2. Specify the "-partspec" command line option
3. Point to the file containing the list of parts

If you do not specify a part specification file, then part version information for all parts in the specified release(s) will be exported.

You can specify on the command line (using the `-partindex n` option) the number of parts to be processed at one time. The parts found in the part specification file can include SQL wildcard characters ('%' or '_'). Using wildcards will slow down performance, however, due to the "like" clause in the query to TeamConnection/CMVC.

You can also use the "-where whereClause" option to specify what parts versions should be placed in the export file. When the where clause option is used, the release, release specification and part specification are ignored. This where clause is an "all or nothing" clause that must retrieve all the part versions the user wants migrated. The where clause is used to build TeamConnection PartFullView. Note that the `-where` clause is not supported in `clearexport_cmvc`.

It is recommended that you execute the `teamc Report -view PartFullView` command outside of the migration tool first, adjusting the query to retrieve just the part versions you want migrated. Once the query selects the desired versions, you can then use this option of the migration tool to select the versions to be migrated. For more details on how the tools get the correct version of parts/files from TeamConnection/CMVC, see the Appendix B "Getting the correct versions of parts".

A single revision will in most cases have multiple labels attached to it to indicate which release and/or driver versions it represents. After part versions are migrated to ClearCase, you will be able to query the ClearCase database for all the parts found in a specific TeamConnection or CMVC release or driver (or CMVC level).

Migration from CMVC/TeamConnection to CC/CO

The syntax to export part data from TeamConnection is as follows:

```
clearexport_tc -parts [-release myRelease OR -relspec releaseFile.txt]
[-partspec partspecFile.txt] [-where whereClause] [-partindex nn]
[-o outputFile.txt] [-since "yyyy/mm/dd"] [-crlf] [-target UNIX | NT]
```

The syntax to export part data in CMVC is as follows:

```
clearexport_cmvc -parts [-release myRelease OR -relspec releaseFile.txt]
[-partspec partspecFile.txt] [-partindex nn]
[-o outputFile.txt] [-since "yyyy/mm/dd"] [-crlf] [-componentlabels]
[-sidlabels] [-target UNIX | NT]
```

You must specify either a where clause (to select the part versions from TeamConnection PartFullView command) or specify the release or releases (using the -release myRelease option or -relspec myReleaseSpecFile.txt, which lists the releases to be searched for the parts). Note that the -where clause is not supported in clearexport_cmvc.

You can also specify the parts you want to export by listing them in the part specification file. SQL wildcards ('%' and '_') are acceptable in the part specification file.

The following describes the command parameters in more details:

-partspec partspecFile.txt

This file contains the pathnames of the parts (including optional SQL wildcards) that must be migrated. The pathnames must be one per line, and must be the first item found on the line. It must be delimited by either a blank, carriage return/linefeed, or '|'.

-release myRelease

This specifies the release to migrate parts from. Only one release may be specified. For multiple releases, use the -relspec option.

-relspec releaseFile.txt

This file contains the names of the releases from which part versions are migrated. There must be one release per line, and the release name must be the first item on the line. The releases must be delimited by a blank, carriage return/linefeed, or '|'.

-where whereClause

This specifies the where clause that will be used in a TeamConnection Report -view PartFullView command to select the versions of parts to be migrated. When you specify the where clause, the -release, -relspec, and -partspec switches are ignored. Every part version that needs to be migrated must be selected with the where clause.

WARNING: If you use the -where clause, do not use the SQL order by clause, because the column order for the command is set internally.

Migration from CMVC/TeamConnection to CC/CQ

`-o outputFile.txt`

The export file name. If this is not indicated, the output name will be `partExport.txt`.

`-partindex nn`

Specifies the number of parts to process at one time (when using the `-partspec` option only). The higher the number, the better the performance. However, there is a limit based on the length of the path names of the parts in the part specification file. If you do not specify a value, the default value of 20 is used.

`-since yyyy/mm/dd`

Specifies the date (in `yyyy/mm/dd` format) after which part version information will be extracted from a TeamConnection Report `-view PartFullView` or a CMVC Report `-view FileView` command. This parameter is specifically designed for incremental exports. If you use the `-since` parameter for two sequential exports, make sure you use the same `-partspec` and `-relspec` for both exports.

`-crlf`

Adds the `-crlf` (carriage return line feed) option. This is useful if you are migrating from AIX to NT or the reverse.

`-componentlabels`

This adds the component labels to the part versions. This is only supported in `clearexport_cmvc`.

`-sidlabels`

This adds the sid labels to the part versions. This is only supported in `clearexport_cmvc`.

`-target UNIX | NT`

This creates the correct file extract command for the target operating system. For example, if you specify NT with the `-target` switch, the extract command will add `cmd /c` to the extract command (for example, `cmd/c teamc part -extract ...`). If you specify UNIX with the `-target` switch, the extract command will not add anything to the extract command; (for example, `teamc part -extract...`)

Example of `clearexport_tc`

The following is an example of a command used in TeamConnection that exports the parts listed in `myprtspec.txt` in the releases listed in `myrelspec.txt` (releases v2 and v3) that are dated March 22, 1999 or later. The command is specified in a single line, but for formatting purposes, it is shown in multiple lines:

```
clearexport_tc.exe -parts -partspec myprtspec.txt -since "1999/03/22"  
-o myexport.out -relspec myrelspec.txt
```

Where the contents of `myprtspec.txt` is the following. Note the use of wildcards in this file:

Migration from CMVC/TeamConnection to CC/CQ

```
src/migrate/%  
src/model/fhcfile.cpp  
src/java/%
```

Where the contents of myrelspec.txt is:

```
v2  
v3
```

After you have produced an export file, you can then use this file to import data into ClearCase. Instructions for doing this can be found in the section entitled, "Importing parts into ClearCase."

Notes on creating a partspec file

In the above example, the partspec file (myprtspec.txt) was created by typing the information into a txt file. However, you can create a partspec file using the TeamConnection/CMVC report command. For example, in TeamConnection running on UNIX, you could create a partspec file as in the following example:

```
teamc report -general partfullview -select "nupathname" \  
-where "releasename='v303' and compname='migrate' \  
order by nupathname" | uniq -c | awk '{ print $2 }' > myprtspec.txt
```

You could write a similar kind of script in the Intel environment as well.

Migration from CMVC/TeamConnection to CC/CQ

Exporting parts without version history for ClearCase into an export file

This section is provided for the following customers:

- Customers who are running a UNIX server on a combination that does not support the command Report -general: AIX on Sybase; HP-UX on Oracle; Solaris on Oracle/Sybase.
- Customers who have other reasons for wanting to migrate parts without version history.

This method of exporting parts has the following limitations:

- It only migrates the part contents and places it into ClearCase. It does not migrate metadata.
- It can be repeated for different releases, but not for the same release.

The basic steps for migrating parts without version history are as follows:

1. Ensure that the target directory is empty.
2. Extract the current release from TeamConnection/CMVC into the target directory.
3. Run the ClearCase utility `clearexport_ffile`. This produces an export file containing release information.
4. Run the ClearCase utility `clearimport` to process the export file and create ClearCase elements.

The following describes the above steps in detail:

Step 1. Ensure that the target directory is empty.

The target directory is shown as `myExtractPath` in the examples below. If you have used the target directory before, it might be possible that some backup files (`.file` or `file.$bak` or `file._bak`) are in the directory. These backup files should not be migrated.

Step 2. Extract the current release from TeamConnection/CMVC.

Make sure you replace the bracketed items in the statements below with your own information.

A. To perform a release extract in TeamConnection, issue the following command:

```
teamc Release -extract <myRelease> -root <myExtractPath> -nokeys
```

B. To perform a release extract in CMVC, issue the following command:

```
Release -extract <myRelease> -node <myNode> -root <myExtractPath>  
[ -uid myUid -gid myGid ] -nokeys
```

Step 3. Run the ClearCase `clearexport_ffile` utility.

Migration from CMVC/TeamConnection to CC/CQ

This creates an export file containing part information in a release.

Before running this utility, you should first change your current working directory to the root directory of the release extract (myExtractPath above). Also make sure you have logged into the correct host (that is, where the release was extracted to) with the appropriate access rights. To do this, issue the following commands:

```
cd <myExtractPath>
clearexport_ffile -r -o <myExportFile>
```

After you have produced an export file, you can then use this file to import data into ClearCase. Instructions for doing this can be found in the next section called "Importing parts into ClearCase".

Migration from CMVC/TeamConnection to CC/CQ

Importing parts into ClearCase

After you create an export file, you use this file as input to the ClearCase import utility.

If your export file contains version history, the import utility uses the information in the file to extract the part information from TeamConnection/CMVC and to import the parts into ClearCase. Consequently you must be running a TeamConnection/CMVC client when importing parts.

If your export file does not contain version history, no part extraction is necessary and you do not need to be running a TeamConnection/CMVC client.

Important notes:

- If you are using UNIX, complete Step 1 and skip Step 1-A.
- If you are using Windows NT, skip Step 1 and complete Step 1-A.

Step 1. Running the ClearCase clearimport utility on UNIX.

If you are using UNIX, complete the following steps to run the ClearCase clearimport utility:

The steps involved in using the clearimport utility to import data include using ClearCase commands to create a view, create a VOB, and mount a VOB. When that is complete, you then run the clearimport command, which reads your export file and creates ClearCase elements for each file and directory in the TeamConnection (or CMVC) release extract. To complete these tasks, issue the following commands (make sure you replace the bracketed items with you own information):

```
mkdir mountOverDir
cleartool mkvob -nc -public -tag <myVOBTag> vobStorageDirectory
cleartool mkview -tag <myViewTag> viewStorageDirectory
cleartool setview <myViewTag>
cleartool mount <myVOBTag>
cd <myVOBTag>
clearimport -verbose <myExportFile> -pcase -nsetevent -identical
```

Notes:

- The -identical flag ensures that identical versions of the same part in CMVC are imported.
- When running the mkvob command to create a public VOB, you may need to supply a password to add the VOB to the registry.

Skip Step 1-A and proceed to Step 2.

Migration from CMVC/TeamConnection to CC/CQ

Step 1-A. Running the ClearCase clearimport utility on Windows NT.

If you are using Windows NT, complete the following steps to run the ClearCase clearimport utility. Make sure you replace the bracketed items with your own information.

1. Create a directory for the VOB storage location. For example: d:\cc\views\vob
2. Using Windows Explorer, make the directory shared by completing the following steps:
 - Right click on the directory and select properties
 - Click on the sharing tab and make the directory shared.
3. Create a new VOB using the ClearCase Home Base application by completing the following steps:
 - Click on the VOBs tag.
 - Provide the name of the VOB.
 - Enter the shared name of the storage location using the following format:

`\\cc\views\vob\vobName.vbs`
4. Mount the VOB using the ClearCase Home Base application by completing the following steps:
 - Select Mount a VOB.
 - If the VOB is in the list, then it must be mounted. Otherwise it is already mounted.
 - Click on the VOB, and then click OK.
5. Create a view using the ClearCase Home Base application by completing the following steps:
 - Click on the Views tag.
 - Select create a dynamic view.
 - Provide the view name and storage location. For example: \\cc\views\viewName.vws
6. Start the view using the ClearCase Home Base application by clicking on the views tab.
7. Run the clearimport utility by completing the following steps:
 - Issue the NET USE command to determine the shared storage location of the VOB (vobTag) .
 - Change the directory to the vobTag location .
 - Run the command: clearimport -verbose <myExportFile> -identical

Note: The -identical flag ensures that identical versions of the same part in CMVC are imported.

Proceed to Step 2

Migration from CMVC/TeamConnection to CC/CQ

Step 2. Adding Labels to the Imported Directories

After completing Step 1 (or Step 1A), you will have two new files in the directory in which you ran the export operation:

- SavedDirsFile.txt
- SavedLablesFile.txt.

You need these files to add release, driver, and component labels to the clearcase directory structure. To add these labels, complete the following steps:

1. Change to your ClearCase VOB directory.
2. Copy SavedDirsFile.txt and SavedLabelsFile.txt to the ClearCase VOB directory.
3. Execute the following command:

```
makelabels SavedDirsFile.txt SavedLabelsFile.txt runfile.bat
```

This creates a batch file called runfile.bat that contains all the necessary clearcase mklable commands for the imported directories.

4. If you are running UNIX, make runfile.bat executable. Execute runfile.bat

Hints and tips for migrating the data

You are likely to perform the migration task several times, and thus, we suggest several tasks to make each migration easier and more reliable:

Take note of the time that it took to perform the test migrations

Take note of how long it takes you to perform the practice migration, so you know for how long the source family will need to be in maintenance mode.

ClearExportGui: Java-based GUI for Windows users

ClearExportGui is an optional user interface that is provided “as is”, without warranty or support. It was tested only on Windows. This utility is a Java Application that requires JDK 1.2 or later.

The `clearexport*` tool creates the export files from CMVC or VA TC. However, the real migration is done using the CC/CQ commands; during this step is where the logging takes place. For users of Windows platforms, we provide an optional Java-based GUI tool, named “ClearExportGui” which can be used as a front-end to specify the desired options to the `clearexport*`.

ClearExportGui is a simple graphical user interface to the `clearexport_tc` and `clearexport_cmvc` tools that are used to migrate VisualAge TeamConnection and CMVC databases to ClearCase/ClearQuest. This GUI provides the user with an alternative for using the command line interface and all of its parameters.

The interface has two basic panels:

- The first panel is used as input to the migration of Users, Defects, and Features.
- The second panel is used as input to the migration of Parts/Files.

Invoking the tool

ClearExportGui is contained in the package `clearExportGui`. This package is supplied as a jar file, “ClearExportGui.jar”, that should be added to your system CLASSPATH. Example:

```
set CLASSPATH=%CLASSPATH%;ClearExportGui.jar
```

To execute the application, enter the following from the command prompt:

```
java clearExportGui.ClearExportGui
```

The package and class names are case sensitive; if the class is not found, or if the method ‘main’ is not found when executing the application, verify that the correct case is used in the CLASSPATH and in the command that was entered.

Usage notes

- Some of the parameters used while exporting files are mutually exclusive. As a result, these fields will become disabled and grayed out if data is entered into one of these fields.
- While exporting Files, the Label SIDS and Label Components check boxes will only be available when exporting from CMVC. These options are not supported from TeamConnection. As a result, if TeamConnection is the selected source database, these options will be hidden.

Migration from CMVC/TeamConnection to CC/CQ

Exporting Users, Defects, Features

This section describes the dialog for the export of Users, Defects and Features.

Export Type

This radio button is used to select the type of data to export.

- Users – Export Users
- Defects – Export Defects
- Features – Export Features

Mapping

This radio button is used to indicate if the exported defects/features should be mapped to the default ClearQuest schema. The default value is “map”.

Export Source

This radio button is used to select the type of database that is being migrated.

Query String

Text field used to provide the selection criteria for the export. The field is initialized with the default query string for the type of object being exported.

Output File

Text field used to provide the name of a file that is generated by the `clearexport_tc` and `clearexport_cmvc` tools. Use the corresponding “Browse” button for a file dialog to select the file from the filesystem.

Input File

Text field used to provide the name of an input file to be used as input to the `clearexport_tc` and `clearexport_cmvc` tools. Use the corresponding “Browse” button for a file dialog to select the file from the filesystem.

Scrolling View Port

This scrollable view port is used to display the output of the tool.

Save As

This button is used to save the generated output to a file. A file dialog will appear.

Clear

This button will clear the output area.

Export

This button will launch the `clearexport_tc` or `clearexport_cmvc` tool.

Migration from CMVC/TeamConnection to CC/CQ

Exit

This button will exit the ClearExportGui application.

Migration from CMVC/TeamConnection to CC/CO

Exporting Files (Parts)

This section describes the dialog for the export of Users, Defects and Features.

Export Source

This radio button is used to select the type of database of the migration.

- CMVC - Migrating a CMVC database
- TeamConnection – Migrating a TeamConnection database.

Release

Text field to indicate a specific release to export.

RelSpec File

Text field used to provide the name of a file that contains a list of releases to export. Use the corresponding “Browse” button for a file dialog to select the file from the filesystem.

PartSpec File

Text field used to provide the name of a file that contains a list of parts to export. Use the corresponding “Browse” button for a file dialog to select the file from the filesystem.

Index

Text field used to indicate the number of objects to process at a time.

Update Since

Text field used to provide a date for which an incremental export should be done. Refer to the documentation on the `clearexport_tc` for exact syntax of the date.

OutFile

Text field used to provide the name of a file that is generated by the `clearexport_tc` and `clear-export_cmvc` tools. Use the corresponding “Browse” button for a file dialog to select the file from the filesystem.

Where

Text field used to provide a conditional selection of objects to be exported.

crlf

Checkbox used to indicate that the crlf should be done.

Label SIDs

This check box is used to indicate that labels should be added to all SIDs. This function is only available when exporting from CMVC.

Migration from CMVC/TeamConnection to CC/CQ

Label Components

This check box is used to indicate that labels should be added to all components. This function is only available when exporting from CMVC.

Scrolling View Port

This scrollable view port is used to display the output of the `clearexport_tc` and `clearexport_cmvc`.

Save As

This button is used to save the generated output to a file. A file dialog will launch.

Clear

This button will clear the output area.

Export

This button will launch the `clearexport_tc` or `clearexport_cmvc` tool.

Verification of the new CC/CQ environment

Checklist for verifying the new CC/CQ environment

Target system: These activities are done in the host system where the new CC/CQ environment will be located. This system will be called the **target system**.

	Description of task	Est. Time	Comment
	Review the output of export and import actions. There might be warnings or errors that you need to be aware of.	½ hr	
	Explore some objects in the new target family and verify that they look fine.	½ hr	Manual observation.

After the migration has been performed, it is necessary to verify that the new TeamConnection family has valid data. You can perform the following verifications:

1. First of all, you will need to review the report from the clearexport tool and the import actions, to see if there are any errors or warnings reported by the migration utility.
If there are errors, try to fix them.
2. Manually explore some objects, such as view a defect, display the contents of a file, etc.
 - To test the successful migration of file contents, it is recommended to manually extract releases from VA TC and CMVC and use scripts to compare (using “diff” command) all of the extracted files with those in a specified ClearCase view of the same release.
 - The only validation of defects was verifying that the correct number of objects were created in CQ. You can manually look through some of the defects to verify that all of the expected data was imported and the correct mapping was done based on the table in Appendix A.

Migration from CMVC/TeamConnection to CC/CQ

Post-migration tasks for the new CC/CQ environment

Checklist for performing post-migration tasks for the new CC/CQ environment

Target system: These activities are done in the host system where the new CC/CQ environment is located. This system is called the **target system**.

	Description of task	Est. Time	Comment
	For both the source family and the target CC/CQ environment, update the administration system scripts or cron jobs to include the changes. You may need to enable back any cron jobs that were disable before the migration to avoid an interruption of the migration process.	1 hr	For example, to include the new CC/CQ environment with the daily backup scripts.
	May rewrite existing user exits used in VA TC or CMVC as CC triggers.	1 hr	Note: not all the existing user exits have logic that will apply to the new environment in CC.
	Make a complete backup of the new CC/CQ environment	1 hr	
	Notify the users that the new CC/CQ environment is ready to be used.	5 min	

Appendix A: Mappings of defects, features and parts

Defect and feature mapping

This section shows you the mapping that occurs when you migrate defects and features into ClearQuest with the mapping switch turned on.

When a defect is migrated from CMVC or VA TC to CQ, the defect name from CMVC or VA TC is stored in the field OLD_ID in CQ. CQ assigns a new defect number to the migrated defect.

The following TeamConnection and CMVC fields are automatically mapped into ClearQuest as follows:

<u>TeamConnection/CMVC Field</u>	<u>ClearQuest Field</u>
State	State
Abstract	Headline
AddDate	Submit_Date
OriginLogin	Submitter
OwnerLogin	Owner
Symptom	Symptoms
Priority	Priority
Severity	Severity
Notes	Notes_Log
Name	old_id

The TeamConnection and CMVC states are mapped as follows:

<u>TeamConnection/CMVC State</u>	<u>ClearQuest State</u>
Open	Submitted
Design	Submitted
Size	Submitted
Review	Submitted
Working	Open
Returned	Submitted
Verify	Resolved
Cancelled	Closed
Closed	Closed

The following TeamConnection and CMVC Priorities are mapped as follows:

<u>TeamConnection/CMVC Priority</u>	<u>ClearQuest Priority</u>
Mustfix	1-Resolve Immediately

Migration from CMVC/TeamConnection to CC/CO

Candidate	2-Give High Attention
Deferred	3-Normal Queue
Difficult	3-Normal Queue
Easy	3-Normal Queue
Moderate	3-Normal Queue
n/a	4-Low Priority

The following TeamConnection and CMVC Symptoms are mapped as follows:

<u>TeamConnection/CMVC Symptom</u>	<u>ClearQuest Symptom</u>
Lost_data	Data Loss
Core_dump	System Crash
Usability	Unfriendly Behavior
Function_needed	Missing Feature
Incorrect_io	Incorrect Operation
Install_failed	Installation Problem
Performance	Performance Problem
Intgr_problem	Unexpected Behavior
Build_failed	Unexpected Behavior
Non_standard	Unexpected Behavior
Not_to_spec	Unexpected Behavior
Obsolete_code	Unexpected Behavior
Plans_incorrect	Unexpected Behavior
Prog_suspended	Unexpected Behavior
Program_defect	Incorrect Operation
Reliability	Unexpected Behavior
Test_failed	Unexpected Behavior
Design_wrong	Unexpected Behavior
Docs_incorrect	Unexpected Behavior

Appendix B: Getting the correct versions of parts

This appendix provides details about the process we use to get the correct versions of parts from TeamConnection and CMVC.

Getting the correct versions of parts from TeamConnection

- The following activities are executed:
- The tools get the required part version records from the PartFullView command.
- All the part version records for a particular part are processed and placed in memory. This provides the candidates for actual versions that will be extracted and placed into ClearCase.
- The tools execute a RevisionView command and get a report for the same part, and store this information in memory. This report shows when the part was modified and committed to a release or driver.
- The tools match information from the two reports for each part to determine which revisions of the part to extract.
- The tools create entries in the export file that are used to create the version history tree during the import. The actual part extract command is executed during the import.

In most cases, each part revision will be a node on the main branch in ClearCase. However, when two (or more) part versions have the same parent, one will be placed on the ClearCase mainline branch, and the others will be placed on branches off the mainline.

Although a part may have many version records associated with it in TeamConnection, it will have fewer revision records. This is because each time a part is moved into a driver or up to the release, a new version record is generated, even if the part has not changed. However, a revision record will only be generated when the contents of the part are changed.

As mentioned earlier, only the revisions of a part will actually be extracted and placed into ClearCase during the import. However, every part version found in the PartFullView command will be represented in the export file, either by performing a part extract (when the part content actually changed), or by placing a label(s) on the part corresponding to the correct important version(s). Each part version will be labeled with its release name and driver name. The release name will be labeled:

TC-RELEASE-yourReleaseName

The drivers will be labeled:

TC-DRIVER-yourDriverName

Migration from CMVC/TeamConnection to CC/CQ

Getting the correct versions of files from CMVC

To get the correct versions of files from CMVC, the following activities are executed:

- The tools get the records for the desired files from the FileView command.
- The tools process all the records for one file, combining the FileView records with the records found in the CommittedVersions table (if any) for this file.
- The tools perform a query of the Versions table to get the correct versions of the files that need to be extracted and placed into ClearCase.
- The tools create the correct entries in the export file that will identify and extract the correct file version from CMVC during the import.

Each file version will be labeled with the appropriate Release and Level it was found in. Optionally, the customer can select to label the version with the component the version was found in and the SID associated with the file.

The release name, level name, component name (if selected) and SID (if selected) will be labeled as followed:

CMVC-RELEASE-yourReleaseName

CMVC-LEVEL-yourLevelName

CMVC-COMPONENT-yourComponentName

CMVC-SID-yourSIDName

Appendix C: How to get the files mentioned in this document

All the tools used here are available via the public Internet. The tools might be updated in the future. The tools are zipped into a single file called **mig-cccq-tools.zip** and it can be downloaded as follows:

FTP site for VisualAge TeamConnection

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

1. ftp ftp.software.ibm.com
2. login as **anonymous** and for password give your email address.
3. cd ps/products/teamconnection/papers
4. binary
5. get mig-cccq-tools.zip
6. quit

Obtaining Info-ZIP

The VisualAge TeamConnection team used 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.
- **Cross-platform:** A zip file prepared in Unix can be unzipped in the correct format in Windows NT and vice versa.

Info-ZIP's software 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/>

Because of the general value of these tools, it is recommended that you add the unzip and zip tools in a directory in the PATH that is accessible to all the users for the machine.

How to unzip files

- To only view the contents of the zip file (without actually unpackaging and uncompressing the files) do: **unzip -l mig-cccq-tools.zip**
- To unpackage and uncompress the zip file do: **unzip mig-cccq-tools.zip**

Migration from CMVC/TeamConnection to CC/CQ

Brief description of the migration tools

The following files are extracted when the above zip file is unzipped:

VA TC migration tool

For Windows: clearexport_tc.exe
For AIX: clearexport_tc
For HP-UX: clearexport_tc.hpux

CMVC migration tool

For Windows: clearexport_cmvc.exe
For AIX: clearexport_cmvc
For HP-UX: clearexport_cmvc.hpux

Front end GUI for Windows

clearExportGUI.jar

CMVC Korn shell scripts to prepare the table with file versions in committed Levels

processCommittedLevelsForRelease
createCommittedVersions

Scripts to facilitate the query of DBMSs

db2Script
informixScript
oracleScript

Miscellaneous tools

makelabels

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, VisualAge, TeamConnection, DB2 Universal Database, CMVC, AIX	IBM Corp.
Unix	Unix System Laboratories, Inc.
ClearCase, ClearQuest	Rational Software Corp.
Windows, Windows NT	Microsoft Corporation
Intel	Intel Corporation
Info-ZIP	Info-ZIP Group
Solaris, Java	Sun Microsystems, Corp.
HP-UX	Hewlett-Packard Corp.
Informix	Informix Corp.
Sybase	Sybase Corp..
Oracle	Oracle Corp.

***** END OF DOCUMENT *****