

## **Migration to VisualAge TeamConnection Version 3**

# Migration from VisualAge® TeamConnection™ Version 2 to VisualAge TeamConnection Enterprise Server Version 3

Kevin Postreich, Mark Dunn, Lee Perlov, Angel Rivera

TR 29.3209

VisualAge TeamConnection Development Team  
IBM Software Solutions  
Research Triangle Park, North Carolina, USA

© Copyright 1999, IBM Corp.

### **DISCLAIMER:**

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

## **Migration to VisualAge TeamConnection Version 3**

# **Migration to VisualAge TeamConnection Version 3**

## **ABSTRACT**

This technical report provides a procedure and tools that aid in the migration of a VisualAge TeamConnection Version 2 family to VisualAge TeamConnection Enterprise Server Version 3. 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**

- VisualAge
- TeamConnection
- Migration

## **Migration to VisualAge TeamConnection Version 3**

### **ABOUT THE AUTHORS**

#### **Kevin Postreich**

Mr. Postreich is a staff software engineer with the VisualAge TeamConnection development group. He joined IBM® in 1980 as an electronic engineer in Charlotte, North Carolina. He relocated to RTP as an MVS systems programmer.

Mr. Postreich is currently a member of the TeamConnection development/test team, and a member of the first defense team for customer support.

#### **Mark Dunn**

Mr. Dunn is a advisory software engineer with the VisualAge TeamConnection development group. He rejoined IBM in 1996 and is currently in charge of development and support of TeamConnection migration.

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.

#### **Lee R. Perlov**

Mr. Perlov is an advisory software engineer in the VisualAge TeamConnection/CMVC development team. 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. 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.

Mr. Perlov is currently a member of the VisualAge TeamConnection Services team, as well as family, system and web administrator for the services TeamConnection family.

#### **Angel Rivera**

Mr. Rivera is an advisory software engineer with the VisualAge TeamConnection/CMVC development team. He joined IBM in 1989 and since then has worked in the development and support of library systems.

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

Mr. Rivera is currently the team lead for the development of CMVC 2.3 and provides service support for customers of VisualAge TeamConnection.

# Migration to VisualAge TeamConnection Version 3

<b>ABSTRACT</b>	- 3 -
ITIRC KEYWORDS	- 3 -
<b>ABOUT THE AUTHORS</b>	- 4 -
<b>Introduction</b>	- 9 -
Note about migration fine grain data	- 9 -
Note to users of TeamConnection Version 1	- 9 -
Notes about VisualAge TeamConnection Version 3 updates	- 10 -
Disclaimer	- 10 -
Acknowledgments	- 10 -
How to get the most up to date version of this technical report	- 10 -
<b>Overview of the migration process</b>	- 11 -
General suggestions	- 11 -
Notes for practice migrations	- 11 -
<b>Preparing the migration plan</b>	- 12 -
Important assumptions about the migration	- 12 -
Planning considerations	- 12 -
Workarounds for migrating data due to defect when family is started in maintenace mode	- 14 -
Important migration notes about the sequence number for defects and features	- 14 -
Whether to clone or not to clone a Version 2 family	- 15 -
Whether to prune a Version 2 family	- 15 -
Migrate only committed release versions	- 15 -
Time considerations	- 16 -
Day by Day Plan	- 16 -
Day: TC - 1 day (family servers)	- 16 -
Day: TC - 1 day (clients and build servers)	- 17 -
Day: TC - 1 day (users and usability)	- 17 -

## Migration to VisualAge TeamConnection Version 3

Day: TC - 1 day (test migration)	- 17 -
Day: TC (server side)	- 18 -
Day: TC (client side for the test migration)	- 18 -
Day: TC + 1 day (server side)	- 18 -
<b>Gathering all the materials</b>	- 19 -
Finding documentation about the migration process	- 19 -
Migration tools	- 19 -
Checklist of material related to VisualAge TeamConnection	- 20 -
<b>Prepare the source Version 2 family for the migration</b>	- 21 -
Checklist for migrating a TeamConnection Version 2 family to 2.0.9	- 21 -
Checklist for preparing the Version 2 family for migration	- 22 -
What to do to the Version 2 family before migration	- 25 -
<b>Prepare the target Version 3 family</b>	- 28 -
Checklist for creating a new Version 3 family	- 28 -
Check for FixPaks and HotFixes	- 30 -
Important information about installing the code and creating user ids	- 30 -
Create family using tcadmin or dbcreate	- 30 -
Environment setup for running mkmiglst and migration utility (migtc)	- 30 -
<b>Performing the migration</b>	- 32 -
Checklist for migrating the Version 2 family to the Version 3 family	- 32 -
Background on mkmiglst	- 33 -
Compiling mkmiglst	- 33 -
Usage statement of mkmiglst	- 33 -
Help statement of mkmiglst	- 33 -
Description:	- 33 -
Details:	- 34 -
Options:	- 34 -

## Migration to VisualAge TeamConnection Version 3

Notes: .....	- 35 -
<b>Example of using mkmiglst to generate the migration script</b> .....	- 35 -
<b>Example of migration file 1 generated by mkmiglst: myfile1.lst</b> .....	- 36 -
<b>Example of migration file 2 generated by mkmiglst: myfile2.lst (Last committed versions)</b> .....	- 36 -
<b>Example of migration file 2 generated by mkmiglst: myfile2.lst (Migrate all versions of parts)</b> .....	- 37 -
<b>Example of file with migration instructions generated by mkmiglst: myfile.txt</b> .....	- 38 -
<b>Hints and tips for migrating the data from Version 2</b> .....	- 39 -
<b>Take note of the time that it took to perform the test migrations</b> .....	- 39 -
<b>Qualify appropriate migration steps</b> .....	- 39 -
<b>Redirect output during migration</b> .....	- 39 -
<b>Executing the migration scripts</b> .....	- 40 -
<b>Updating the configuration files</b> .....	- 40 -
<b>Updating the configurable fields</b> .....	- 41 -
<b>Updating the user exits</b> .....	- 41 -
<b>Be aware of the difference in user exits between Version 2 and Version 3</b> .....	- 41 -
<b>Verification of the new Version 3 family</b> .....	- 42 -
<b>Checklist for verifying the new Version 3 family</b> .....	- 42 -
<b>Post-migration tasks for the new Version 3 family</b> .....	- 43 -
<b>Checklist for performing post-migration tasks for the new Version 3 family</b> .....	- 43 -
<b>Updating the sequence number for defects and features</b> .....	- 44 -
<b>Our experiences migrating to TeamConnection</b>	
<b>Version 3</b> .....	- 45 -
<b>Migration decisions</b> .....	- 45 -
<b>Do we migrate all of our releases or just the active ones?</b> .....	- 45 -
<b>Do we migrate the change history or just the currently committed version of each file?</b> .....	- 45 -
<b>Do we clean up the family before the migration?</b> .....	- 45 -
<b>Results of migration decisions</b> .....	- 45 -

## Migration to VisualAge TeamConnection Version 3

<b>Migration Shell Scripts</b> .....	- 47 -
Automated verification tool: migVerify.ksh .....	- 47 -
Completing a release in TeamConnection Version 2 before migration: release_completeV2.ksh .....	- 47 -
Assign a File Mode (fmode) to Parts in V209 before migration: processfmode.ksh .....	- 48 -
<b>Warning and Error Messages</b> .....	- 49 -
<b>Obtaining the Tools</b> .....	- 50 -
IBM Intranet .....	- 50 -
Web Home Page for TeamConnection .....	- 50 -
FTP site for TeamConnection .....	- 50 -
Public Internet .....	- 50 -
Web Home Page for TeamConnection .....	- 50 -
FTP site for TeamConnection .....	- 50 -
<b>Obtaining Info-ZIP</b> .....	- 51 -
How to unzip files .....	- 51 -
<b>Copyrights, Trademarks and Service marks</b> .....	- 52 -



# Migration to VisualAge TeamConnection Version 3

## Introduction

This technical report provides a procedure and tools that aid in the migration of a VisualAge TeamConnection Version 2 family to VisualAge TeamConnection Enterprise Server Version 3. The objective is to provide a structured process, to reduce errors and to save time by selecting only appropriate data for the migration.

This technical report is a complement to (rather than a replacement of) the published documentation provided with VisualAge TeamConnection Enterprise Server Version 3.

The current Migration facility (migtc) to migrate a VisualAge TeamConnection Version 2 family to VisualAge TeamConnection Enterprise Server Version 3 is a general purpose tool with considerable capability and flexibility. Unfortunately, it does not enforce a methodology and this may result in mistakes that could be avoided if the user follows a well defined procedure.

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.

We discuss also our experience of migrating from CMVC 2.3 to VisualAge TeamConnection Version 2 and then to VisualAge TeamConnection Version 3.

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

### Note about migration fine grain data

This document does not cover the migration of fine grain data (repository data that cannot be accessed by the VisualAge TeamConnection GUI or line commands, but by a tool that exploits the repository interface of VisualAge TeamConnection). However, the VisualAge TeamConnection customer service and support can provide C++ programs to assist customers with the migration of fine grain data.

### Note to users of TeamConnection Version 1

You cannot directly migrate a TeamConnection Version 1 family to VisualAge TeamConnection Enterprise Server Version 3. You need to migrate the Version 1 family to Version 2, and then migrate the Version 2 family to Version 3. This technical report focuses only on the migration from Version 2.0.9 with latest hotfix to Version 3.

## Migration to VisualAge TeamConnection Version 3

### Notes about VisualAge TeamConnection Version 3 updates

- This technical report refers to VisualAge TeamConnection Version 3, FixPak 3.0.2 which has significant enhancements to the migration tool.
- The FixPak 3.0.3 does not introduce any additional changes related to the migration, and thus, the discussion in this document applies also to FixPak 3.0.3.

### Disclaimer

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

### Acknowledgments

Many of the questions and answers that are compiled in this technical report were obtained from the TEAMC forum in the IBMPC conferencing disk. We want to thank the main participants in these electronic forums for their support!

We want to thank in particular the following coworkers:

- Gary Greig, CMVC and VisualAge TeamConnection support team, IBM Austin, Texas, USA.
- Keith Purcell, OEM Lab in IBM RTP, North Carolina, USA.
- Jim Austin, AIX Lab in IBM RTP, North Carolina, USA.
- Jon Gregory, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.
- Brent Wunderlich, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.
- Clifford Meyers, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.
- Tim Orłowski, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.
- Alberta Lee, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.
- Gregory Melahn, VisualAge TeamConnection team, IBM RTP, North Carolina, USA.

### 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 VisualAge TeamConnection Enterprise Server related web sites:

- Library home page by selecting the item **Library** at the web address:  
<http://www.software.ibm.com/ad/teamcon>
- FTP site by accessing the URL:  
<ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/README.index.txt>  
<ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/trtc2tc3.pdf>

# Migration to VisualAge TeamConnection Version 3

## Overview of the migration process

### General suggestions

Here is a broad overview of the steps needed to migrate a VisualAge TeamConnection Version 2 family to VisualAge TeamConnection Enterprise Server Version 3. 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 VisualAge TeamConnection Version 3 that requires the participation of your entire team.
  - Train users on VisualAge TeamConnection Version 3 prior to migration. Key differences are identified in the **Appendix “Differences between TeamConnection 2 and TeamConnection 3”**, in the technical report **Comparison between CMVC 2.3.1 and VisualAge TeamConnection Enterprise Server 3**.
2. Gather all materials needed for planning and migration.
3. Prepare the VisualAge TeamConnection Version 2 family for migration (source family).
4. Prepare the new VisualAge TeamConnection Version 3 family (target family).

A new family needs to be created in VisualAge TeamConnection 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. Whether the migration is for practice or a real one, these are the basic steps:

  - Select the releases to be migrated.
  - Construct a migration script using **mkmiglst**.
  - Review the generated migration list and make changes as needed.
  - Migrate the releases using the instructions generated by **mkmiglst**.
6. Verify the new family.
  - Verify the migration, in Unix you can use **migVerify**.
7. Perform post-migration tasks on the new family.

### Notes for practice migrations

- Your team should extensively test your new TeamConnection Version 3 family after migrating from TeamConnection Version 2 for a very limited period of time, such as one full day. Do everything that you do normally in your development and reporting efforts. You can get a feel for a day's activity by looking at the audit log of the source family. This log resides in \$HOME/audit.log (Unix) or %TC\_DBPATH%\audit.log (Intel) of the family user id.
- At the end of testing, have a quick meeting to assess the results and refine the plan for the final migration.
- Once you pass a day of testing your new TeamConnection Version 3 family with no major problems, then discard this testing TeamConnection family.
- Finally, perform the migration again for a final time, and start using your new TeamConnection Version 3 family.

## Migration to VisualAge TeamConnection Version 3

### 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 (column labeled “Est. 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 <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 plan enough time to do backups of the source VisualAge TeamConnection Version 2 family and of the target VisualAge TeamConnection Version 3 family 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).
- Take into account that the migration facility is very flexible and thus, you can expect to decide to change options and try again.
- Most tools described in this document are written for Unix. However, **mkmig1st** is written in C and includes instructions for compiling it on Intel and on Unix.

#### Planning considerations

Moving your data from a VisualAge TeamConnection Version 2 family to Version 3 will require careful planning. The answers to the following questions are critical because they can help you determine whether you will need to:

- Clone any VisualAge TeamConnection Version 2 families from your current host to another host that has more resources and/or you want to continue working with some releases in Version 2 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.

## Migration to VisualAge TeamConnection Version 3

These are the questions you need to ask yourself:

- Will all of the releases from my current Version 2 family be migrated, or only some of them?
- If not all of the releases will be migrated, are there any releases that need to be kept in the Version 2 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 Version 2 family (which means that this family needs to be active in order for users to find information about these file changes) as a reference family and migrate only the latest file change to Visual TeamConnection Version 3? If only the last file change will be migrated, then migrating only the latest file change (the “tip version”) will save time during the migration and disk space with the new TeamConnection Version 3 family.  
We recommend that only the tip versions should be migrated.
- Is there enough disk space to keep both Version 2 and Version 3 families at the same time in the same host?  
In Version 2, the size of the tcd database file (such as \$TC\_DBPATH/familyName.tcd) can give you an indication of how much disk space is being used.  
In Version 3, you can find out how much disk space is being used by issuing:
  - If using \$DB2\_DBPATH: `du -k -s $DB2_DBPATH`
  - If not using \$DB2\_DBPATH: `du -k -s $DB2INSTANCE_HOME`
- Is there enough CPU and system resources to run Version 2 and Version 3 families at the same time in the same host? You need to consider the resources that are needed to run ObjectStore for the Version 2 family and DB2 Universal Database® for the Version 3 family. You can use the command “vmstat” to find out the basic resource utilization in a system.
- Decide how you are going to stage your migration from Version 2 to Version 3: 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 Version 2 and Version 3 in parallel while all the families are being migrated.
- If the Version 2 production family needs to be active for normal work after the migration, then it might be necessary to make a clone of this family and migrate to Version 3 from this cloned family. To clone a family is to make an identical copy of the family into another host (same name, same setup, same database contents). In this case, we recommend that the Version 2 production family should be taken off-line during the cloning process. 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.

## Migration to VisualAge TeamConnection Version 3

### Workarounds for migrating data due to defect when family is started in maintenance mode

Because of a defect with the Version 2 **teamcd** command, a Version 2 family cannot be placed under **maintenance mode** (that is, read-only). That is, even if you start the family with the -m flag, the family is still in read-write mode. Because the migration utility will request data from the Version 2 family, this family should be up and running; however, it is not desirable to allow users to modify the Version 2 family while its data is being migrated. Some alternatives to this situation are:

- Start the family in a different port number that will be known only to the family administrator.  
For example, if the normal port is 2345, then the administrator could use 3456 as the special port to be used during migration, starting with 2 daemons:  
`teamcd familyName@3456 2`
- Clone the family (see preceding item, above)
- Use user exits in the update actions. 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 vatc@tcaix05."
    print -u2 "                for all v3 work"
fi
exit ${RC}
```

This generic user exit should be defined in `$TC_HOME/config/userExit` for all update actions using user exit 0 as follows (this is just one example):

```
WorkAreaUndo    0        generic
```

### Important migration notes about the sequence number for defects and features

If you do not migrate all the defects and features from the Version 2 family (including all completed and canceled ones), then there might be gaps in the numbering sequence for the defects and features that will be opened in the Version 3 family after the migration.

In order to avoid these gaps being filled in by VisualAge TeamConnection Version 3, you will need to find out what is the latest sequence number for a defect in the source Version 2 family and then update with this number the sequence number for defects in the target Version 3 family. In that way you will avoid having one defect in the target Version 3 family that has the same number as a closed defect in the source Version 2 family but that refer to different problems.

## Migration to VisualAge TeamConnection Version 3

For details on how to modify the sequence number in the target Version 3 family, see section *Updating the sequence number for defects and features*.

### Whether to clone or not to clone a Version 2 family

If you are going to keep some releases in a Version 2 family around for reference and if you have other releases that are going to continue to stay in production in Version 2, then you should consider cloning your source Version 2 family.

By duplicating the contents of a Version 2 family in another family, you can make one read-only family for migration and continue work in the other family.

### Cloning a Version 2 family:

- Make a copy of the database using the ObjectStore 'oscp' command  
Syntax: `oscp family.tcd clone.tcd`  
Where `family.tcd` is the production family and `clone.tcd` is the new cloned family.
- Copy the cloned family to its target location using `ftp` or the `copy/cp` commands.
- Setup the directory structure for your new family, and then `ftp` or `copy` all of the files from the source family directory structure.

Directory structure:

```
TC_DBPATH/  
    clone.tcd  
    *.ld  
    schema.adb  
    config/  
        security  
        userExit  
    cfgField/  
        *.tbl  
        *.fmt
```

- Setup your TC environment variables. Refer to the Version 2 Administrator's Guide for details.
- Start the new family.

### Whether to prune a Version 2 family

Pruning a Version 2 family is a good way to save space after migrating releases to Version 3 or cloning a family. Pruning a Version 2 family eliminates confusion about which release in which family is the one to be updated.

### Migrate only committed release versions

Another option is to only migrate committed release versions of the version 2 family. This can be done by setting on the target machine the environment variable:

```
TC_PARTFULLVIEW_REL_ONLY=1.
```

## Migration to VisualAge TeamConnection Version 3

This variable tells the TeamConnection migration utility to only migrate parts from release versions and skip parts in workareas and drivers. This migration method allows for a much quicker migration, while migrating key history of and file changes from all driver and/or workarea commit commands. Parts in workareas and drivers will not be migrated. However, all release versions will be migrated. A release version is created each time a driver or a workarea is committed to the release.

### 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 Version 2 family will get your target Version 3 family up and available much faster.

In order to save time and effort, once you have migrated the active releases, put the source Version 2 family in read-only mode (with the appropriate user exits as mentioned before) and remove it from your backup list. There is no reason to backup a family that does not change.

### 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 **TC day**. Based on that target date there are several activities that need to be done prior, during and after that date.

### Day: TC - 1 day (family servers)

Although we assume that the Version 3 family is installed on a different host than the Version 2 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 (Version 2):

- Ensure that all users of the new target Version 3 family have the proper host lists, passwords and access in the current source Version 2 family. In that way, the new family will be ready to handle the authentication of users after the migration from Version 2 to Version 3.
- Ensure that the TeamConnection user conducting the migration is defined as a TeamConnection SuperUser in the Version 2 family.
- Get the migration tools mentioned in this document. See *Obtaining the Tools* at the end of this document.
- Setup the proper environment variables for the migration process.

### Target system (Version 3):

- If you have a pilot project with beta code from VisualAge TeamConnection Version 3, you need to delete the beta code from the server host.
- Remove any old versions of TeamConnection so that the server points to the current code.



## Migration to VisualAge TeamConnection Version 3

- Install VisualAge TeamConnection Version 3 and DB2 UDB with the latest FixPak. Follow the directions in the README.TXT and INSTALL.TXT. See the section *Gathering all the materials*.
- Plan the backup and restore strategy for your new family, because of the change from Object-Store (in Version 2) to DB2 UDB (in Version 3); their commands are different.
- Get the migration tools mentioned in this document. See *Obtaining the Tools* at the end of this document.
- Setup the proper environment variables for the migration process.

### Day: TC - 1 day (clients and build servers)

- Ensure that everybody in the team has installed the VisualAge TeamConnection Version 3 client with the latest FixPak and HotFix.
- Remove any old versions of TeamConnection, such as the beta versions used in a pilot project.
- Prepare a list of team members that will test the new target Version 3 family from unique combinations of client platforms, such as Windows NT and AIX®. The idea is to ensure that all relevant client platforms can work with Version 3 in your network configuration.

### Day: TC - 1 day (users and usability)

- Schedule and deliver training sessions for the users of the source Version 2 family to provide them with the essentials of how to use the new target Version 3 family. A starting point is to present the **Appendix “Differences between TeamConnection 2 and TeamConnection 3”**, of the technical report **Comparison between CMVC 2.3.1 and VisualAge TeamConnection Enterprise Server 3**.
- Review with key users how you all are using the current source Version 2 family (such as component structure, naming convention) and decide what changes, if any, will be done with the new Version 3 family.
- Do not take for granted that the current setup in the source Version 2 family is perfect and keep in mind that it might be possible to do some fine tuning with the setup in the new target Version 3 family just after the migration but before overall production use.

### Day: TC - 1 day (test migration)

- Migrate the latest snapshot of the source Version 2 family into the new test target Version 3 family.
- The new target Version 3 family 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 TC day.
- The new target Version 3 family should be up for a few hours to allow the users to test their access to the family.
- Rehearse the backup and restore strategy for the new test family.
- All team members ensure that they are ready for TC day: they should be able to connect to the new target Version 3 family. This can be checked by issuing the command **ping \$TC\_FAMILY** (where \$TC\_FAMILY is the name of the family) and if this is successful, then issue the command **teamc report -testServer**.

## Migration to VisualAge TeamConnection Version 3

- They should have the proper host list and access list. This can be checked by issuing the command **teamc user -view userName**, where userName is the name of each user.
- The user passwords from the Version 2 family are NOT migrated (due to security reasons). They will have to be re-created in the Version 3 family after the migration is complete.

### Day: TC (server side)

- The new target Version 3 family is started with the desired number of family servers, which is **roughly** the cubit root of the total number of users. For example, if you have 25 users, the closest upper cubic root is 3 ( $3*3*3 = 27$ ): **teamcd \$TC\_FAMILY 3**
- If using passwords instead of host lists, the system administrator should setup the users with new passwords, as the existing passwords are not migrated.
- The target Version 3 family administrator and selected members of the team should monitor the new family, paying particular attention to not running out of disk space. The TeamConnection **monitor** tool can be used to watch the activity in the TeamConnection family. Also, the family **\$TC\_DBPATH/audit.log** file can be monitored by using: **tail -f audit.log**
- The family administrator may decide to stop the family for few minutes during the middle of the day and restart it again, just to see if there are any problems with this action.
- At the end of the day, the target Version 3 family will be shut down.
- Perform a backup of the family and then do a restore, to test your backup/restore strategy.

### Day: TC (client side for the test migration)

- Every member of the team will spend time interacting with the test target Version 3 family. First with the specific assignments and then later on in an exploratory mode.
- Once there are several workareas and drivers, you should try to do a "normal" build. That is, if you are not using the build function of TeamConnection, extract the release and then start builds in all the platforms from the extracted code. If you are using the build function, then select the top build object and initiate a build event.
- 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 migration results of the test Version 3 family, proceed to perform the real migration.

### Day: TC + 1 day (server side)

- The test Version 3 family should be deleted in order to prepare the next test (if needed) or the final migration.
- Perform the final migration for the family by repeating the tasks described in **Day: TC**. Keep in mind that this time the migration is not a test anymore; it is the final migration.
- Start the family servers for the new family.
- Perform the verification of the family.

# Migration to VisualAge TeamConnection Version 3

## Gathering all the materials

### Finding documentation about the migration process

The single largest challenge when performing a migration to VisualAge TeamConnection Enterprise Server Version 3 is sifting through the available documentation in order to perform all of the necessary actions. Here is a brief overview:

1. Read the **README.TXT** and **RELNOTES.PDF** files. This file points to all other installation documents. For English, it is located in the FullPak CD-ROM for 3.0.1 as follows:  
**Intel:** readme.txt and *nationalLanguage/relnotes.pdf*  
**Unix:** readme.txt and *nationalLanguage/relnotes.pdf*  
See in the readme.txt for v301, "Migration Notes for Version 3.0.1".
2. Read the migration directions in **Chapter 8, Migrating to TeamConnection Version 3** in the **VisualAge TeamConnection Installation Guide** (choose the desired format: **INSTALL.PDF**, **INSTALL.HTM** or **INSTALL.TXT**) from the FullPak CD-ROM 3.0.1:  
**Intel:** nls/doc/*nationalLanguage/install.pdf*  
**Unix:** softpubs/*nationalLanguage/install.pdf*
3. Use this technical report to plan for the actual migration.

### Notes:

\* The installation and migration sections of the **Administrator's Guide Version 3.0.0** are out-of-date, therefore, do not use them. The version 3.0.1 or later of this manual does not have the migration instructions, because starting with FixPak 3.0.1, they are now contained in the Installation Guide.

\* You need Acrobat Reader to view the documentation in PDF format; you can download it from the VisualAge TeamConnection web site:

<http://www.software.ibm.com/ad/teamcon/downloads>

### Migration tools

The migration tools mentioned in this technical report are packaged in a zip file (**migrate.zip**). Because we may update the migrate.zip, we are showing below only some of the tools that are included:

- Unix: mkmiglst.c, migVerify.ksh, release\_completeV2.ksh, processfmode.ksh, allv2.idl
- OS/2®: mkmiglst.c, find.exe, rm.exe, grep.exe, tail.exe, tee.exe, which.exe, allv2.idl
- Windows NT: mkmiglst.c, find.exe, grep.exe, rm.exe, tail.exe, tee.exe, which.exe, allv2.idl

### Notes:

- The shell scripts (suffix .ksh) are described in the section **Migration Shell Scripts** in this document.
- The **mkmiglst** program is described in the section **Performing the Migration** chapter of this document.

## Migration to VisualAge TeamConnection Version 3

### Checklist of material related to VisualAge TeamConnection

	Document/Media Name	Current Version
	VisualAge TeamConnection Version 2 CD-ROM	2.0.5
	VisualAge TeamConnection Version 2 FixPak 2.0.9.	2.0.9
	VisualAge TeamConnection Version 2 Administrator's Guide	2
	VisualAge TeamConnection Enterprise Server V3 CD-ROM	3.0.1
	VisualAge TeamConnection Enterprise Server V3 FixPak	3.0.2
	VisualAge TeamConnection Enterprise Server V3 HotFix(s)	contact IBM
	VisualAge TeamConnection Enterprise Server V3: Administrator's Guide and Installation Guide (softcopy)	3.0.1
	Technical Report 29.3076: Configuration and Administration of DB2 Universal Database V5 by users of VisualAge TeamConnection Enterprise Server V3	TeamConnection ftp site. File <b>trtc3db2.pdf</b>
	Technical Report 29.3094: VisualAge TeamConnection Version 3: how to do routine operating system tasks	TeamConnection ftp site. File <b>trtc3oem.pdf</b>
	Technical Report 29.3096: Comparison between CMVC 2.3.1 and VisualAge TeamConnection Enterprise Server 3.	TeamConnection ftp site. File <b>trtc3-cm.pdf</b>
	Technical Report 29.3032: Making the most of VisualAge TeamConnection user exits.	TeamConnection ftp site. File <b>trtcusrx.pdf</b>
	Migration tools, packaged in the file <b>migrate.zip</b> .	TeamConnection ftp site. File <b>migrate.zip</b>

#### Notes:

1. To get the latest VisualAge TeamConnection code, you can access the following sites:
  - Public Internet URL: <http://www.software.ibm.com/ad/teamcon/>
2. To get the latest VisualAge TeamConnection tools, you can access the following sites:
  - Public Internet URL: <ftp://ftp.software.ibm.com/ps/products/teamconnection/tools/>
  - IBM intranet URL: <http://wwwteamc.raleigh.ibm.com/tools.html>
3. To get the latest VisualAge TeamConnection technical reports, you can access the following sites:
  - Public Internet URL: <ftp://ftp.software.ibm.com/ps/products/teamconnection/papers/>
  - IBM intranet URL: <http://wwwteamc.raleigh.ibm.com/papers.html>

## Migration to VisualAge TeamConnection Version 3

### Prepare the source Version 2 family for the migration

#### Checklist for migrating a TeamConnection Version 2 family to 2.0.9

The following steps are needed to upgrade a Version 2 family to the latest FixPak for 2.0.9, which is needed for the migration.

**Source system:** These activities are done in the host system where the Version 2 family is located. This system will be called the “source system”.

	Description of task	Est. Time	Comments
	Make a complete backup of the original Version 2 family.	1 hr	See <b>VisualAge TeamConnection Administrator’s Guide Version 2.</b>
	Install the FixPak 2.0.9 code.	1 hr	See the installation instructions provided with the FixPak code.
	Apply the latest HotFix for v209 from the ftp site. HotFix 3 required	1 hr	Includes a new teamcd for migrating build relationships stored in TeamConnection.
	Make a backup of your upgraded 2.0.9 family.	1 hr	See <b>VisualAge TeamConnection Administrator’s Guide Version 2.</b>

## Migration to VisualAge TeamConnection Version 3

### Checklist for preparing the Version 2 family for migration

**Note:** You must have a VisualAge TeamConnection Version 2 family running on FixPak 2.0.9 in order to migrate to VisualAge TeamConnection Version 3. If you use TeamConnection version 2 to do builds and want to migrate the build information to Version 3, you will also have to install the latest Version 209 HotFix available on the VisualAge TeamConnection ftp site.

**Source system:** These activities are done in the host system where the Version 2 family is located. This system will be called the **source system**.

	Description of task	Est. Time	Comments
	If the family administration id (superuser) of the new target family is different than the one in the source family, then create this user id in the source family and ensure that this user id is a superuser.	5 min	If the superuser for the target family is not a superuser in the source family, then the authorization mechanism in Version 2 may prevent the migration of all the data and there will be no warnings.
	As a family superuser, add host list entries for the new host where the new target Version 3 family will reside: <b>teamc Host -create login@hostName</b> Where <i>login</i> is the system login and <i>hostName</i> is the new host where the target Version 3 family will reside.	5 min	If you do not add these host list entries, you will not be able to access the new family using the default authentication which uses host list entries (HOST_ONLY).
	Find out the value of the latest defect number from the Sequence table. See section <b>What to do to the Version 2 family before migration</b> .	5 min	After the migration, this number should be given to the defect number in the Sequence table in the Version 3 family.
	Backup the version 2 family database.	1 hr	See <b>VisualAge TeamConnection Administrator's Guide Version 2</b> .
	Copy <b>allv2.idl</b> from the TeamConnection samples directory to your TC_DBPATH directory.	5 min	This is a new view required to migrate renamed parts.
	Set the environment variable: <b>TC_VIEWS=STC_DBPATH/allv2.idl</b>	5 min	To dynamically load a new view.
	Restart the Version 2 family by running: <b>teamcd familyName N</b> Where <i>N</i> is the number of daemons (at least 2).	5 min	The family should be active while migrating. We recommend you migrate from a cloned Version 2 database.

## Migration to VisualAge TeamConnection Version 3

	<b>AIX:</b> If the family fails to start, contact IBM TeamConnection Level 3 Support for the latest version of the teamcd executable.		
	Verify the new view is setup properly by running: <b>tcselect describe allver2</b>	1 min	There should be 20 rows of data as a result of the command.
	Disable the automated shell scripts that might affect the availability of the source Version 2 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.
	Ensure all files in the family have a valid fmode by running: <b>teamc report -raw -view PartFullView -where "fmode is null"</b>	10 min	For each part that is listed, you must add a valid fmode. <i>See next step.</i>
	Modify all files that do not have a valid fmode by running: <b>teamc part -modify partName -release releaseName -workarea workAreaName -fmode 777 -verb</b>	30 min	For Unix, a sample shell script is included in the migtc.zip file in the tools directory on the ftp site. The file name is <b>processfmode.ksh</b> .
	Remove anything from the Version 2 family you do not need anymore, such as old host list entries or obsolete entries in the authority list.	1 hr	There is no need to migrate obsolete data.
	If necessary, trim the number of superusers.	30 min	Some customers have a long list of superusers. It is not a good idea to have too many superusers because they can bypass in many cases the formal processes in TeamConnection.
	Reorganize the component hierarchy (if needed) or make changes to the naming conventions.	30 min	Take advantage of the transition to the new family to improve the usage of TeamConnection.
	Ensure that all parts are unlocked. One way to find out is by executing the following command: <b>teamc report -view PartsOutView -where "userLogin like '%'"</b>	30 min	The parts must be unlocked for migration.
	It is required to complete all workareas and drivers of the releases to be migrated.	1 - 2 hrs	<b>Unix:</b> The sample script release_completeV2.ksh may be used.

### Migration to VisualAge TeamConnection Version 3

		<b>Intel:</b> Commit and complete all workar-eas and drivers.
Execute: <b>teamc report -view defectView</b> . If it fails, then fix it by running: <b>chfield -object Defect -source \$HOME</b> <b>chfield -object Feature -source \$HOME</b>	½ hr	The migration utility uses the <b>teamc Report</b> command
Unset the <b>LIBPATH</b> environment variable.	5 min	Paging space problems could occur if it is set.



## Migration to VisualAge TeamConnection Version 3

### What to do to the Version 2 family before migration

The VisualAge TeamConnection migration utility, **migt**, will use VisualAge TeamConnection client commands. The only requirements from VisualAge TeamConnection are:

- The VisualAge TeamConnection client Version 3 must be installed on the target system where the TeamConnection server Version 3 is installed.
- If the Version 2 family is not using the latest FixPak 2.0.9, then it must be upgraded to 2.0.9.
- Ensure that you apply HotFix 3 for FixPak 2.0.9.

There are many preparation tasks that can be performed in the Version 2 family prior to migrating it to Version 3.

- Remove anything from the Version 2 family you do not need anymore, such as old host list entries or obsolete entries in the authority list.
- You can take advantage of this transition to reorganize the component hierarchy (if needed) or to make changes to the naming conventions, such as changing release names from very long names with upper and lower case letters, to no more than eight lower-case letters (for example).
- 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).

- **Find out the defect sequence number.**

The sequence number that the source Version 2 family is using for naming defects will be used later on after the migration process to setup the corresponding sequence number in the target Version 3 family in order to allow a continuous numbering scheme. If the sequence number is not properly updated in the new target family then there might be gaps in the numbering of the defects in the new family because it may be possible that the canceled defects from the source Version 2 family (for example, defect 30) were not migrated and thus, the new target Version 3 family might use their numbers when creating new defects (such as a new defect 30). Do the following to find out the sequence number in the Version 2 family:

- Find out the current number: **tcselect "\*" from Sequence"**

The output will look like this. The sequence number for defects in this example is 590.

This number will be used later on in the migration process.

NAME	LASTSERIAL
defect	590
source	435
General	34

## Migration to VisualAge TeamConnection Version 3

- **Ensure all parts in TeamConnection Version 2 contain a valid 'fmode'**

We have discovered failures during the migration of parts from the Version 2 database that do not have an fmode, which tells TeamConnection the file mode characteristics such as "read-only". The migration utility requires this information in order to determine how to create the part in the Version 3 family.

If the fmode of a Version 2 part is 0, then during the migration process you may encounter the following error message (even though the part can be manually extracted and the information looks good).

```
MIGPART, no bulk data so file X has file type none
```

To avoid this error, perform the following:

- **Step 1.** Determine which files need to be modified in TeamConnection Version 2.

```
teamc report -raw -view PartFullView -where "fmode is null"
```

- **Step 2.** Issue the following part modify command to add a valid fmode to all the parts listed in step 1 (in one single line).

```
teamc part -modify partName -release releaseName  
-workarea workAreaName -fmode 777 -verbose
```

- **Complete some releases as required.**

It is important to complete **certain** releases in the Version 2 family because uncommitted workareas that contain multiple versions of a part are not guaranteed to migrate successfully. Although this case is rare, you must take precautions to avoid future problems.

- **Step 1.** To determine which releases need to be completed prior to migration, use the following command. The resulting records from the query represent files that have more than a single version in a workarea. Record the list of releases that are affected and complete as stated (in a single line):

```
teamc report -raw -view partFullView  
-where "workAreaName is not null order by releaseName, nuPathName"  
| awk -F\| '{ print $12, $2, $17 }' | uniq -d
```

- **Step 2.** You may use the shell script **release\_completeV2.ksh** in Unix to assist with this process.

**NOTE: The sample script release\_completeV2.ksh may need to be modified prior to its execution.**

- If you are cloning the Version 2 family, then as the family administrator you may need to update the database configuration for defects and features:
  - `chfield -object Defect -source $HOME`
  - `chfield -object Feature -source $HOME`

## Migration to VisualAge TeamConnection Version 3

- **Ensure that all environment variables are properly set.**

Ensure that all of the environment variables required for migration are properly set for your configuration, for example:

```
TC_FAMILY=TargetFamily
TC_BECOME=SuperUserId
TC_VIEWS=$TC_DBPATH/allv2.idl
```

- Ensure that LIBPATH is **not** set:

**Unix:** unset LIBPATH

**Intel:** set LIBPATH=

We assume that all other TeamConnection 2.0.9 environment variables have been set properly as a matter of operation of the production family.

## Migration to VisualAge TeamConnection Version 3

### Prepare the target Version 3 family

#### Checklist for creating a new Version 3 family

The following steps are needed to install VisualAge TeamConnection Enterprise Server Version 3 and to create a new family which will be used as the place to store the data migrated from the source Version 2 family.

**Target system:** These activities are done in the host system where the new VisualAge TeamConnection Version 3 family will be located. This system will be called the **target system**.

Description of task	Est. Time	Comment
If using VisualAge TeamConnection Version 1 or Version 2, then you may want to remove it, in order to free up system resources for the Version 3 family.	1 hr	For performance reasons, it is not recommended to have VA TC Version 3 and a previous version in the same system.
Install the VisualAge TeamConnection Enterprise Server Version 3.0.1 or later.	1 hr	See <b>Verifying the Installation of TeamConnection</b> .
Install DB2 Universal Database (DB2 UDB) Version 5; then create the DB2 instance.	1 hr	See the technical report <b>Configuration and Administration of DB2 Universal Database V5 by users of VisualAge TeamConnection Enterprise Server V3</b>
Install the DB2 UDB FixPak and any HotFixes available from the TeamConection FullPak 301 CD; update the DB2 instance.	½ hr	From Unix CD-ROM, see: <b>platform/db2/fixes/readme.db2</b> From OS/2 CD-ROM, see: <b>dbfiles0/fixes/readmeo.db2</b> From Windows CD-ROM, see: <b>dbfilesw/fixes/readmew.db2</b>
Install the FixPak 3.0.2 of VisualAge TeamConnection. This FixPak does not include DB2 UDB.	1 hr	See <b>README.INSTALL</b> to install the FixPak. Then see <b>FIXPAK301.README</b> to update the family.
Install the latest HotFix for FixPak 3.0.2 of VisualAge TeamConnection. HotFix 3 required.	1 hr	See the appropriate readme file for the HotFix.
Create the family in TeamConnection. Use the sample profile and update it. You may need to add the following new environment variables that are not part of the sample profile:	½ hr	See in this chapter, the section <b>Environment setup for running mkmiglst and migration utility (migtc)</b> .

## Migration to VisualAge TeamConnection Version 3

<pre>export TC_PFV_NUPATH=1 export TC_OLD_CLIENT=1</pre>		
<p>It is recommended that during the migration process, the authentication level of the target family should be the default <code>HOST_ONLY</code>. The required use of a password (such as with <code>PASSWORD_ONLY</code>) may interfere with the migration.</p>	5 min	<p>This is setup in the Properties notebook in TCADMIN or in the file <code>\$HOME/config/security</code></p>
<p>Logout and login again in order to have a fresh environment.</p>	5 min	<p>You should avoid simply re-executing the profile because there are some cumulative variables that may cause problems if they are not properly set.</p>
<p>Start the TeamConnection family:  <b>teamcd <i>yourFamily</i> <i>N</i></b>          Where <i>N</i> is the number of daemons.          For AIX, it may be required to be started with the -d option:  <b>teamcd -d <i>yourFamily</i></b></p>	5 min	<p>The TeamConnection Version 3 family must be operational in order to migrate the data from the source Version 2 family.</p>
<p>Ensure that the notification daemon, <b>notifyd</b>, should not be active. If it is active, then terminate it:</p> <ul style="list-style-type: none"> <li>• find out its process id (pid):  <b>ps -ef   grep notifyd</b></li> <li>• terminate it:  <b>kill -15 <i>pid</i></b></li> </ul>	5 min	<p>During migration some notification messages might be generated; because these messages are not needed, the notification daemon should not be running.</p>
<p>Ensure that the source Version 2 family can be reached. Execute in one single line:  <b>teamc report -testServer -family <i>sourceFamily</i></b>          Where <i>sourceFamily</i> is the v2.0.9 family.          You should see a message saying that the family server is active.</p>	5 min	<p>If this command fails, then migtc will fail too. It is better to fix the problem now.</p>
<p>Make a backup of the new target family. It is faster to restore a newly created family from a backup than to recreate it from scratch.</p>	½ hr	<p>Use the sample <b>tcbakup</b>, which first backs up the DB2 database and then the complete home directory of the family. See the comments of the header of that utility.</p>
<p>Setup the environment variables needed for the TeamConnection migration utilities.</p>	10 min	<p>Refer to the following text under the heading:</p>

## Migration to VisualAge TeamConnection Version 3

		Environment setup for running <b>mkmiglst</b> and migration utility ( <b>migtc</b> ).
--	--	---

### Check for FixPaks and HotFixes

Determine what version of VisualAge TeamConnection you are using, then go to the IBM software FTP site to see if any FixPaks or HotFixes are available that could make the migration go easier.

To find out the version, run the following command in the **\$STC\_HOME/bin** directory:

```
what teamcd | grep teamc
```

**Note:** Starting with HotFix 3 for FixPak 302 and continuing with FixPak 303, the new command `teamc report -view PartsOutView -where "userLogin like '%'"` can be used to show the information about the version of the VisualAge TeamConnection code.

### Important information about installing the code and creating user ids

After you install the VisualAge TeamConnection Enterprise Server Version 3 code, adhere to the following guidelines:

- Use different directories for the DB2 code, for the VisualAge TeamConnection code, for the DB2 instance and the TeamConnection Version 3 family.
- Do not alter the contents of **\$STC\_HOME** unless you are going to deinstall the product.
- Do not use **\$STC\_HOME** as the home directory for any families or users.

It is very important that the directories and the user ids must be kept separate. This will avoid potential loss during maintenance activities.

### Create family using **tcadmin** or **dbcreate**

You must use the **VisualAge TeamConnection Family Administrator (tcadmin)** tool or the **dbcreate** shell script to create a new VisualAge TeamConnection Version 3 family before migrating. These utilities will delete any existing family database prior to creating a new one, saving the effort of manually dropping the database.

### Environment setup for running **mkmiglst** and migration utility (**migtc**)

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

1. Login to the VisualAge TeamConnection Version 3 family that will be the target for the migration. In this example, it is the Unix family "tcpu3mig".

## Migration to VisualAge TeamConnection Version 3

### 2. Update the environment:

- `export TC_FAMILY_CLIENT=sourceFamilyName`
- `export TC_FAMILY=targetFamilyName`
- `export TC_BECOME=superUserId`
- `export TC_PFV_NUPATH=1`

#### **If migrating to TeamConnection v3.0.2 or higher:**

- `export TC_OLD_CLIENT=1`

The last environment variable listed above, `TC_PFV_NUPATH`, has recently been added to the migration code. It should be set to 1 if you want to successfully migrate parts that have been renamed or deleted. An example using renamed parts is as follows: the user creates part1 in workarea1, then commits workarea1. The user then renames part1 to part2 in workarea2 and commits workarea2. If the user who is doing the migration does not set `TC_PFV_NUPATH=1` before executing the migration utility, then the migrate of partfullview will create a part named part2 in workarea1 (as well as in workarea2). It only migrates the latest name for the part, rather than using the old part name where appropriate.

If you decide to use this environment variable, you must start set the environment variable `TC_VIEWS` before starting the v209 server. `TC_VIEWS` should be set to the full path name for allver2.idl file. For more details see the section **Ensure that all environment variables are properly set** in the previous chapter.

## Migration to VisualAge TeamConnection Version 3

### Performing the migration

#### Checklist for migrating the Version 2 family to the Version 3 family

**Target system:** These activities are done in the host system where the new VisualAge TeamConnection Version 3 family 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 Version 3 family or the DB2 instance. 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.
Compile <code>mkmiglst</code> from the C source file <code>mkmiglst.c</code>	5 min	See the compilation instructions in the header of <code>mkmiglst.c</code>
Run <code>mkmiglst</code> to prepare the migration script.	1/2 hr	If needed, obtain sign-off from management or technical leaders for the resulting migration scripts.
Run <code>migt</code> with the migration scripts by following the migration instructions.	1 - 12 hrs	These files are generated by <code>mkmiglst</code> .
Generate and load the new (*.ld) files for configurable items: <ul style="list-style-type: none"> <li>• authority groups: <code>authorit.ld</code></li> <li>• notification groups: <code>interest.ld</code></li> <li>• component processes: <code>comproc.ld</code></li> <li>• releases processes: <code>relproc.ld</code></li> <li>• configurable types: <code>config.ld</code></li> </ul>	½ hr	See section <b>Preparing to administer your new database</b> in <b>Chapter 8. Migrating to TeamConnection version 3</b> of the <b>Installation Guide</b> , which is located in the CD-ROM 3.0.1 in: <b>Intel:</b> nls/doc/enu/install.pdf <b>Unix:</b> softpubs/en_US/install.pdf
Setup the configurable fields (*.tbl , *.fmt) in <code>\$STC_DBPATH/cfgField</code> These configurable fields files are not migrated. Use TCADMIN to add the configurable field data to TeamConnection.	½ hr	See above comment for instructions.
Setup the user exits in <code>\$STC_DBPATH/config/userExit</code> Place the executables in a directory that is in the PATH, such as \$HOME/bin	½ hr	See above comment for instructions.



## Migration to VisualAge TeamConnection Version 3

### Background on mkmiglst

The TeamConnection utility **migt** is used to migrate a Version 2 family to Version 3. It is a front end to a generalized SQL query and update mechanism. This tool should be used only during migration.

The **migt** utility provides a command line environment from which you can issue the migration commands. Because the order of the commands for migration is very important, it is recommended to use an input file that contains migrate commands that can be piped to the **migt** utility to perform a generic migration. Each of the migrate commands can be customized using SQL syntax to fine tune the selection of data to be migrated.

In order to make migration easier, this technical report describes a tool **mkmiglst** that generates a customized input file for the **migt** utility. This input file is much more useful than the generic **migt.lst** file shipped with TeamConnection Version 3.

### Compiling mkmiglst

The tool **mkmiglst** is provided in a C source code file. You need to compile it for your platform:

- Unix:
  - `unset LIBPATH`
  - `cc -D__UNIX__ mkmiglst.c -o mkmiglst`
- Intel:
  - `icc -D__INTEL__ mkmiglst.c`

**Note:** On OS/2 and Windows NT, **mkmiglst** uses **rm.exe** (to delete files and directories) and **tail.exe** (to show the last lines in a text file) that are provided in **migrate.zip**.

### Usage statement of mkmiglst

If you invoke **mkmiglst** without arguments, then the usage statement is displayed:

```
mkmiglst usage:
mkmiglst -T|-C|-h|-? -f|-c [-o <outfile>] [-a|-r <release>...]
mkmiglst                - This help message
mkmiglst [-h|-?]        - Extended help
Notes: Parameters must be used in order above,
      Run command from family account home directory.
```

### Help statement of mkmiglst

If you invoke **mkmiglst** with the **-h** or **-?** arguments, then the help text is displayed into different sections:

#### Description:

```
This utility creates a migration script for use by the
VisualAge TeamConnection Version 3 migration utilities:
- migt:   Migrate from TeamConnection Version 2
- migcmvc: Migrate from CMVC Version 2.3.1
```

## Migration to VisualAge TeamConnection Version 3

The generated migration script is input to these utilities.  
Also generated is a file containing instructions for performing the migration.

Example: Migrate all data in a TeamConnection V2 family to a TeamConnection V3 family (generate script, list instructions, migrate):

```
$ mkmiglst -T -c -o migtc -a
$ more migtc.txt
$ migtc < migtc1.lst > migtc.out 2>&1
$ migtc < migtc2.lst >> migtc.out 2>&1
```

Example: CMVC 2.3.1.3 to TeamConnection V3 migration, migrating snapshots of selected releases from CMVC 2.3.1.3 family to a TeamConnection V3 family (generate script, list instructions, migrate):

```
$ mkmiglst -C -f -o migcmvc -r v209 -r v210
$ more migcmvc.txt
$ migcmvc < migcmvc1.lst 2>&1 | tee migcmvc1.out
$ migcmvc < migcmvc2.lst 2>&1 | tee migcmvc2.out
```

Example: CMVC 2.3.1.3 to TeamConnection V3 migration, migrating snapshots of selected releases inputted from a file:

```
$ mkmiglst -C -f -o migcmvc < release.list
$ more migcmvc.txt
$ migcmvc < migcmvc1.lst 2>&1 | tee migcmvc1.out
$ migcmvc < migcmvc2.lst 2>&1 | tee migcmvc2.out
```

### Details:

Executing mkmiglst generates the following outputs (assume "-o XXX" used):

- XXX.txt: Instructions for performing migration, customized based on the parameters used in mkmiglst.
- XXX1.lst: First file containing actual migration commands, customized based on the parameters used in mkmiglst. Commands in this file migrate all non-part data.
- XXX2.lst: Second file containing actual migration commands, customized based on the parameters used in mkmiglst. Commands in this file migrate all files/parts (snapshot of release or all change history).

These scripts do the following:

- Create an extract directory.
- Create a subdirectory for each release.
- Provide a commented entry that will allow for delete of directory after the release is migrated.

### Options:

Migrate from CMVC, TeamConnection, or provide help:

- C: To migrate from CMVC Version 2.3.1.3 (upgrade to CMVC 2.3.1.3 required)
- T: To migrate from VisualAge TeamConnection Version 2
- h or -?: For this detailed help

Migrate snapshot of files or complete change history:

- f: Migrate FileView (snapshot of committed release)
- c: Migrate ChangeView (complete history of release)

Output file option:

- o: Specify filename for migration listfile. Default=migrate.lst.

Migrate selected release or all releases:

- r ReleaseName: The release to be migrated. Use -r for each release.
- a: To migrate the entire family (all releases)

Note: If no releases are specified, and -a is not used, then the user is prompted to enter each release.

## Migration to VisualAge TeamConnection Version 3

### Notes:

1. This tool and associated technical report is a supplement to the VisualAge TeamConnection Version 3 Administrator's Guide. Read the chapter on Migration before using this tool.
2. The output of this tool is determined by the experiences of the VisualAge TeamConnection development, test and services teams. Editing of the output may be desired, but should not be necessary.
3. Required environment variables  
For CMVC 2.3.1.3:  
To migrate from CMVC 2.3.1:  
CMVC\_FAMILY, CMVC\_USER, CMVC\_BECOME, CMVC\_KEYS, CMVC\_ALLCOMMON  
TC\_FAMILY, TC\_USER, TC\_BECOME, TC\_DBPATH, HOSTNAME  
To migrate from TeamConnection Version 2:  
TC\_FAMILY, TC\_FAMILY\_CLIENT, TC\_USER, TC\_BECOME, HOSTNAME

### Example of using mkmiglst to generate the migration script

In this example, the mkmiglst tool was used to provide the migration scripts for the following example in which we wish to migrate the releases **Release1** and **Release2** from the Version 2 family.

```
mkmiglst -T -f -o myfile -r Release1 -r Release2
```

Where:

- -T indicates that the migration will be from TeamConnection.
- -f indicates to migrate the tip of the versions (not all the history of file changes).
- -o indicates the name to be used for the resulting migration scripts.
- -r indicates one release to be migrated.

The following files will be generated by mkmiglst:

- **myfile1.lst**: The first of 2 migration steps.
- **myfile2.lst**: The second of 2 migration steps.
- **myfile.txt**: The migration instructions.

When mkmiglst is executed with the options shown above, these are the runtime messages:

```
Migrating from TeamConnection Version 2
Migrating snapshot of committed version of release(s)
Source family: tcpc3os@cordoba@3403
Target family: tcpc3mig
Migration file 1: myfile1.lst
Migration file 2: myfile2.lst
Instruction file: myfile.txt
Migrating list of releases
Migrating release: Release1
Migrating release: Release2
Total number of releases in list: 2
Writing to root directory: /home/tcpc3mig
Writing first set of instructions for VATC2 to VATC3
Writing second set of instructions for VATC2 to VATC3
Writing user instructions
Processing complete: Output files generated.
```

## Migration to VisualAge TeamConnection Version 3

### Example of migration file 1 generated by mkmiglst: myfile1.lst

The following file is the migration file 1 (myfile1.lst) generated by mkmiglst.

```
# Writing first set of instructions for VATC2 to VATC3
# If you use release.delete.ksh set maxerrors to 500
# (allowing for duplicate records)
set maxerrors 5000
# This is the recommended batchsize (a smaller number reduces the
# rollback on failure)
set batchsize 100
# Migrating individual tables
migrate Users
migrate HostView
migrate Authority
migrate Interest
migrate Cfgcomproc
migrate Cfgrelproc
migrate Config
migrate CompView where 1=1 order by addDate
migrate bCompView where name = 'root'
# Recommend migrating all Release records, due to references
# from other objects
migrate ReleaseView
migrate EnvView
migrate AccessView
migrate NotifyView
# Recommend migrating all defects and features, due to references
# from other objects
migrate DefectView
migrate FeatureView
migrate BuilderView
migrate ParserView
migrate DriverView where state in ('commit','complete')
migrate WorkAreaView where state in ('commit','complete')
migrate DriverMemberView order by WorkAreaName asc, committedVersion desc
migrate FixView
migrate ApproverView
migrate ApprovalView
migrate SizeView
migrate TestView
migrate VerifyView
migrate CoreqView
migrate NoteView
quit
```

### Example of migration file 2 generated by mkmiglst: myfile2.lst (Last committed versions)

The following file is the migration file 2 (myfile2.lst) generated by mkmiglst.

Notice that because some lines are larger than 80 characters, we needed to show them in multiple lines in this document; however, in reality they take only one line in the actual script.

```
# Writing second set of instructions for VATC2 to VATC3
# Few warnings should be reported migrating part data
set maxerrors 100
# You can increase decache to increase performance if you do not approach
# your paging space limit during tests
set decache 1
# This is the recommended batchsize (a smaller number reduces
# the rollback on failure)
set batchsize 100
```

## Migration to VisualAge TeamConnection Version 3

```
# Migrating Part data
set top /home/tcpc3mig/extract/Release1
! mkdir -p /home/tcpc3mig/extract/Release1
# Extract without expanding keywords and convert text files to Unix format
! teamc release -extract Release1 -nokeys -crlf -root \
  /home/tcpc3mig/extract/Release1 -family sourceFamily
migrate PartView -release Release1 where dropDate is null
# rm -rf /home/tcpc3mig/extract/Release1
set top /home/tcpc3mig/extract/Release2
! mkdir -p /home/tcpc3mig/extract/Release2
# Extract without expanding keywords and convert text files to Unix format
! teamc release -extract Release2 -nokeys -crlf -root \
  /home/tcpc3mig/extract/Release2 -family sourceFamily
migrate PartView -release Release2 where dropDate is null
# rm -rf /home/tcpc3mig/extract/Release2
quit
```

### Example of migration file 2 generated by mkmiglst: myfile2.lst (Migrate all versions of parts)

The following file is the migration file 2 (myfile2.lst) generated by mkmiglst.

Notice that because some lines are larger than 80 characters, we needed to show them in multiple lines in this document; however, in reality they take only one line in the actual script.

NOTE: Migrate branchpoints is new to FixPak 302 and is required to be executed following the version view commands, and before partview commands.

```
# Writing second set of instructions for VATC2 to VATC3
# Few warnings should be reported migrating part data
set maxerrors 100
# You can increase decache to increase performance if you do not approach
# your paging space limit during tests
set decache 1
# This is the recommended batchsize (a smaller number reduces
# the rollback on failure)
set batchsize 100
set top
migrate VersionView where releasename='Release1' order by addDate
migrate VersionView where releasename='Release2' order by addDate

migrate branchpoints

migrate PartFullView -release Release1
migrate PartView -release Release1 where dropDate is null
migrate ChangeView where releasename='Release1' order by versionID

migrate PartFullView -release Release2
migrate PartView -release Release2 where dropDate is null
migrate ChangeView where releasename='Release1' order by versionID

#migrate CommonParts

#migrate bPartView if you use the build function of Team Connection v209.
#migrate bPartView -release Release1
#migrate bPartView -release Release2
quit
```

## Migration to VisualAge TeamConnection Version 3

### Example of file with migration instructions generated by mkmiglst: myfile.txt

The following file is the file that contains the migration instructions (myfile.txt) generated by mkmiglst.

#### PRE-MIGRATION RECOMMENDATIONS

1. Upgrade TeamConnection V2 family to FixPak 209, then add HotFix 3.  
Technical reports addressing CMVC and TeamConnection migration issues are available at <http://www.software.ibm.com/ad/teamcon> (select Library)
2. Read the README.TXT file provided with the installation media for any notes on migration.
3. Create a superuser login and a host list entry for the TeamConnection V3 family account in the TeamConnection V2 family to be migrated.
4. Backup the VisualAge TeamConnection V2 family.
5. Follow directions for migration in the VisualAge TeamConnection Enterprise Server Version 3 Installation. Use the on-line copy delivered on CD-ROM, as the directions are more current.

#### MIGRATION COMMANDS

Use the following commands, where the migration instructions call for executing migtc:

1. migtc > myfile1.out 2>&l < myfile1.lst
2. For status: tail -f myfile1.out  
\* tail.exe for OS/2 and Windows NT provided in migrate.zip available from the IBM VisualAge TeamConnection ftp site:  
<ftp://ftp.boulder.ibm.com/ps/products/teamconnection/tools>
3. Repeat for myfile2.lst
4. Return to migration instructions for the following:
  - Update starting Defect/Feature number in fhcsequence table
  - Performance Tuning
  - Backup of Database

#### POST-MIGRATION RECOMMENDATIONS

1. Remove authorities from VATC2 family to block any further work on the migrated VATC2 release.

#### VERIFICATION ACTIONS FOR TEAMCONNECTION V2 MIGRATION

1. Change to directory with myfile2.lst
2. migVerify.ksh -T myfile2.lst (**available only for Unix**)  
migVerify.ksh will temporarily create \$MIGEXPORT\_TOP/TCv2/extract and \$MIGEXPORT\_TOP/TCv300/extract directory structures.  
where MIGEXPORT\_TOP is an exported directory on your system.  
export MIGEXPORT\_TOP=/exportedDirectory
3. Review output file: summary.file.

end of instructions

## Migration to VisualAge TeamConnection Version 3

### Hints and tips for migrating the data from Version 2

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

#### Notes:

- The migration script generated by `mkmiglst` includes the invocation for the command **teamc release -extract**. Currently this command does not run when the source Version 2 family is in maintenance mode; for more details see the section **Workarounds for migrating data due to defect when family is started in maintenance mode**. The **teamc Release -extract** command is included mostly for documentation. We suggest performing the extract before running `migt`. However, if you are migrating from a cloned version 2 family, the family is not required to be in maintenance mode (read-only), therefore the extract commands are permitted inside the migration script.
- If the source Version 2 family is not in maintenance mode and users have access to the family, then you risk migrating inconsistent data.

#### 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 Version 2 family will need to be in maintenance mode.

#### Qualify appropriate migration steps

All of the migration steps support SQL for limiting the amount of data that can be migrated. However, when data is referenced by other tables and is not migrated, some warning messages will be displayed and they may be numerous and disconcerting. The Version 2 objects that are not qualified by `mkmiglst` should be allowed to migrate all data.

If it is critical that the data be limited (that is, cannot migrate all data), we recommend contacting IBM VisualAge TeamConnection Services to assist with the migration. For more information, see the following URL:

<http://www.software.ibm.com/ad/teamcon/support/>

We suggest to not qualify the migration of users and migrate all of them (active and deleted). One of the side effects of trying to migrate only the active users and not the deleted ones is that there might be errors during the migration of other objects, such as hosts lists.

#### Redirect output during migration

When running the migration step, we recommend redirecting your output to a file, in that way you can review the output at leisure. The example below redirects both normal output and error messages to the same file, in the correct order:

## Migration to VisualAge TeamConnection Version 3

```
migtc < myfile1.lst 2>&1 | tee myfile1.out
```

If this is a practice run, we also recommend timing your test:

```
timex migtc < myfile1.lst 2>&1 | tee myfile1.out
```

### Note for OS/2:

In OS/2 we have seen cases where the migration does not start properly with the commands above; repeating the message “Restarting...”. In this case we recommend the alternative syntax:

```
c:\> migtc 2>&1 | tee myfile1.out
```

...

Press <Enter> once

```
migtc> exec myfile1.lst
```

In a separate window run:

```
c:\> tail -f myfile1.out
```

## Executing the migration scripts

The actual migration is covered thoroughly in the VisualAge TeamConnection Installation Guide. However, because this document adds steps, it is helpful to tie the two processes together.

To actually migrate the Version 2 family to Version 3, follow the instructions provided with the output file generated from **mkmiglst** which has a file extension of **.txt** (see previous sections).

## Updating the configuration files

The configuration files (\*.ld) should be manually regenerated on the file system following the migration. The migration utility will migrate the configuration data as stored in the database, however the actual files will not be replaced on the file system. See section **Preparing to administer your new database** in **Chapter 8. Migrating to TeamConnection version 3** of the **Installation Guide**, which is located in the CD-ROM FullPak 3.0.1 in:

- **Intel:** nls/doc/enu/install.pdf
- **Unix:** softpubs/en\_US/install.pdf

You cannot use the configurable files as they existed in the Version 2 family. You need to create and modify these files for TeamConnection. You can create the .ld files from the database by redirecting the raw output of a report. The following commands show how to create these files:

```
teamc report -raw -view authority > authorit.ld
teamc report -raw -view interest > interest.ld
teamc report -raw -view cfgrelproc > relproc.ld
teamc report -raw -view cfgcomproc > comproc.ld
teamc report -general config -select "configtype, name, dflt, value1,
```



## Migration to VisualAge TeamConnection Version 3

```
value2, kind, driverid, driverseq, choiceorder, description,  
concat('\ ', concat(helpext, '\ ')) " -family $TC_FAMILY > config.ld
```

**Note:** Replace \$TC\_FAMILY with the name of your TeamConnection family. Otherwise, the command should be typed as shown, aside from line breaks.

### Updating the configurable fields

After creating a TeamConnection Version 3 family, a subdirectory called cfgField is created in the Family \$HOME directory, which usually is also \$TC\_DBPATH (Intel: %TC\_DBPATH%). This directory contains \*.tbl files for each object that has configurable fields, such as Defect.tbl.

The configurable fields files in \$HOME/cfgField (\*.tbl, \*.fmt) must be manually updated after the migration. The migration utility DOES NOT migrate these configurable fields files:

- Use TCADMIN to modify them. TCADMIN is completely documented in the TeamConnection Administrator's Guide.
- Refer to the TeamConnection Installation Guide in the section **Preparing your new Database** for complete details on updating the configurable fields files.
- By default, there should be two \*.tbl files for Defect and Feature. You will need to create any additional \*.tbl files that your current source Version 2 family may contain, such as for Parts and Users.

### Updating the user exits

User Exits from the source Version 2 family are not migrated to the target Version 3. Refer to the TeamConnection Administrator's Guide for details on creating user exits. You should use TCADMIN to generate the required User Exits.

### Be aware of the difference in user exits between Version 2 and Version 3

There are subtle differences in user exit behavior in VisualAge TeamConnection Version 3 with respect to Version 2. The main differences are:

- The order of parameters have changed.
- The names of some exits have changed.
- There is a new feature that allows for easier handling of parameters using the names of the parameter.

There is a technical report covering TeamConnection user exits and it is available from the TeamConnection home page or ftp site (file **trtcusrx.pdf**):

**TR 29.3032 Making the most of VisualAge TeamConnection User Exits**

## Migration to VisualAge TeamConnection Version 3

### Verification of the new Version 3 family

#### Checklist for verifying the new Version 3 family

**Target system:** These activities are done in the host system where the new VisualAge TeamConnection Version 3 family will be located. This system will be called the **target system**.

	Description of task	Est. Time	Comment
	Review output of migtc. There might be warnings or errors that you need to be aware of.	½ hr	See section <b>Migration Warnings and Messages</b> at the end of this document.
	Verify the migrated data by running the <b>teamc Report</b> command on various objects in the target Version 3 family. Perform a comparison with the corresponding <b>teamc Report</b> command on the source Version 2 family.	1 hr	Ensure that the number of reported items by the target Version 3 family matches the ones reported by the source Version 2 family.
	Run the <b>migVerify.ksh</b> tool. Available only for Unix.	1-4 hrs	Usage described in the section <b>Migration shell scripts</b> of this document.
	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. You will need to review the output file from migtc, which has a suffix of **.out**, to see if there are any errors or warnings reported by the migration utility.  
If there are errors, see section **Migration Warnings and Messages** at the end of this document.
2. Run the command **teamc Report** on both families to ensure that the number of records match for different objects, such as users, components. You will need to use the **-where** clause to limit the number of records found by the Version 2 family to match the ones found by the Version 3 family (as defined by the \*.lst file generated by the tool).  
You may create your own shell scripts that have pairs of commands, the first item in the pair for the Version 2 family and the second for the Version 3 family.
3. Manually explore some objects, such as view a defect, show the host lists for a user, see the history of a file (-view -long) if the history was migrated, etc.
4. Execute the **migVerify.ksh** script to verify the following:
  - verify that the correct number of parts were migrated.
  - verify that the correct number of change history versions were migrated.
  - verify that the part content is correct for all parts in the releases migrated.

## Migration to VisualAge TeamConnection Version 3

### Post-migration tasks for the new Version 3 family

#### Checklist for performing post-migration tasks for the new Version 3 family

**Target system:** These activities are done in the host system where the new Version 3 family will be located. This system will be called the “target system”.

	Description of task	Est. Time	Comment
	<p>Update the sequence number for defects and features.</p> <p>If only some of the defects are migrated, then it might be possible that the “defect” record in the Sequence table was not created. In this case it will be necessary to create a dummy defect to force the creation of this row.</p>	5 min	Allow new defects/features to be numbered appropriately
	Reset the password for all users because they are not migrated.	30 min	
	Update the scripts that used data from the Version 2 family and replace them with the appropriate data from the Version 3 family.	1 hr	This is important if you made changes in the organization of the family.
	<p>For both the source Version 2 family and the target Version 3 family, update the administration system scripts or cron jobs to include the new family.</p> <p>You may need to enable any cron jobs that were disabled before the migration to avoid an interruption of the migration process.</p>	1 hr	For example, to include the new Version 3 family with the daily backup scripts.
	<p>Remove files for unnecessary messages:</p> <pre>cd \$TC_HOME/queue rm *</pre>	5 min	Some messages might have been generated and placed in the queue during migration. These messages should not be sent because they might confuse users.
	Make a complete backup of the new Version 3 family	1 hr	
	<p>Update the statistics that will be used by the query optimizer from DB2.</p> <p>Unix: <b>analyzeReorg</b></p> <p>Windows NT: <b>rexx analyzer</b></p>	1 hr	This will improve performance of the new target Version 3 family. Consider scheduling this along with off shift backups. It needs to

## Migration to VisualAge TeamConnection Version 3

OS/2: analyzer		be done periodically, but not daily.
Start the notification daemon to send new messages to users: <b>notifyd familyName mailexit</b>	5 min	This ensures that the users will get notification from the TeamConnection family.
Notify the users that the new Version 3 family is ready to be used.	5 min	

### Updating the sequence number for defects and features

The sequence number that the source Version 2 family uses for naming defects will now be used to setup the corresponding sequence number in the target Version 3 family to allow a continuous numbering scheme. If the sequence number is not properly updated in the new Version 3 family then there might be gaps in the numbering of the defects because may be the canceled defects from the Version 2 family were not migrated and thus, the new Version 3 family might use those numbers. Perform the following to find out the sequence number in the new Version 3 family:

- Connect to the database:  
db2 connect to \$TC\_FAMILY
- Find out the current number:  
db2 "select \* from Sequence"

The output will look like this. The sequence number for defects in this example is 253.

```
NAME                LASTSERIAL
-----
defect                253
source                435
General              34
```

In our example, we found out that the Version 2 family showed a sequence number for defects of 590. Thus, we need to update this sequence number (from 253 to 590) in the new Version 3 family by performing the following:

- Update the value for the defect counter:  
db2 "update Sequence set lastSerial=590 where name='defect' "
- Commit the change:  
db2 commit
- Verify it:  
db2 "select \* from Sequence"

The output should look now like this:

```
NAME                LASTSERIAL
-----
defect                590
source                435
General              34
```

- Terminate the DB2 session:  
db2 terminate
- The next defect that is opened will use the sequence number 591.

## Migration to VisualAge TeamConnection Version 3

### Our experiences migrating to TeamConnection Version 3

#### Migration decisions

When we prepared the plan to migrate our original development CMVC family to VisualAge TeamConnection Version 2, and then later on to migrate to VisualAge TeamConnection Enterprise Server Version 3, we had to address the following questions:

#### Do we migrate all of our releases or just the active ones?

We decided to migrate only the active releases. There was too much data to migrate over a weekend (the period when there would be no development activity) if we wanted to migrate all the releases. We wanted to leave the releases related to VisualAge TeamConnection Version 2 in a Version 2 family. Likewise, we still maintain the CMVC code in a CMVC family.

#### Do we migrate the change history or just the currently committed version of each file?

When we migrated from CMVC to TeamConnection Version 2, we choose to keep our CMVC family around running in maintenance mode, and only migrate the most recently committed files. This minimized the amount of time we were down for migration and preserved in CMVC the historical information.

We did the same thing when we migrated from TeamConnection Version 2 to Version 3: we choose to keep the Version 2 family around in maintenance mode and only migrate the most recently committed files.

#### Do we clean up the family before the migration?

We decided to do the following for our appropriate family (running on AIX):

- Run **release\_completeV2.ksh** on the releases to be migrated
- Review output of **release\_completeV2.ksh** carefully, then verified that all workareas and drivers were in complete state.
- Prune workareas and drivers not used to save space so that we could keep the old family around for a while, just in case we needed to migrate something else.

For the migration from Version 2 to Version 3, we did not run **reassign\_work.ksh**, since we had migrated our Version 2 family from CMVC only 18 months prior to this new migration. Also, we did not use **release\_delete.ksh**, since it is not applicable to TeamConnection.

#### Results of migration decisions

As a result of the above decisions, here is how our migration from Version 2 to Version 3 went:

- 7000+ files (mostly text source files)
- 2 releases (most recent changes, with no change history)
- Migrated from RS/6000™ C20

## **Migration to VisualAge TeamConnection Version 3**

- Migrated to RS/6000 f30
- It took 23 hours to migrate
- 1 hour to migrate base objects
- 4 hours to migrate tip versions
- 18 hours to migrate notes

We did many practice runs, wrote the tools referenced in this document, and tuned what we were migrating along the way, timing the migration to make sure it would fit in a weekend.

## Migration to VisualAge TeamConnection Version 3

### Migration Shell Scripts

All the tools used here are written in Korn shell and are available via the public Internet or through IBM's intranet. The tools are updated periodically, so we are not providing the source in this document. The tools are zipped into a single file called **migrate.zip**. Because we may update the contents of the file migrate.zip, we are showing below only some of the tools that are included:

- Unix: mkmiglst.c, migVerify.ksh, release\_completeV2.ksh, reassign\_work.ksh, release\_delete.ksh, file\_import.ksh, processLevels.ksh, processDefects.ksh
- OS/2: mkmiglst.c, find.exe, rm.exe, grep.exe, tail.exe, tee.exe, which.exe
- Windows NT: mkmiglst.c, find.exe, grep.exe, rm.exe, tail.exe, tee.exe, which.exe

The shells scripts are listed in alphabetical order.

### Automated verification tool: migVerify.ksh

Help contents:

```
USAGE: migVerify.ksh -C|-T MIGRATION_SCRIPT
WHERE: -C   Migration from CMVC to TC
        -T   Migration from TC - TC
        MIGRATION_SCRIPT is the name of the migration script
        that was used to issue the "migrate FileView" or the
        "migrate PartView" commands.
NOTE : script must be run by a SUPERUSER
```

This shell script helps you to verify that the correct number of parts were migrated, that the correct number of change history versions were migrated and that the part content is correct for all parts in the releases migrated.

### Completing a release in TeamConnection Version 2 before migration: release\_completeV2.ksh

Help contents:

```
USAGE: release_completeV2.ksh -f <FAMILY> -r <RELEASE>
WHERE: FAMILY      is the version 2 family to process
        RELEASE    is the active CMVC release
NOTE : script must be run by a SUPERUSER
```

This utility is used to commit and complete all workareas and drivers in the specified release prior to migration. The utility does the following:

- integrate workareas in the fix state
- abstains the required approval records

NOTE: This is a sample utility and may need to be modified before it is executed.

## Migration to VisualAge TeamConnection Version 3

### Assign a File Mode (fmode) to Parts in V209 before migration: processfmode.ksh

Help contents:

```
USAGE: processfmode.ksh RELEASE
WHERE:  RELEASE      is the is the release to be processed
NOTE : script must be run by a SUPERUSER
```

This utility is used to add an fmode to all parts in the specified release in v209 prior to migration.

The utility does the following:

- creates a shell script that the administrator should execute to modify all parts with no valid fmode.

We have discovered that while extracting parts from V209 with no fmode will work. However, it makes it impossible for the part to be moved into the Version 3 database because of the file attributes that are set during the extract.



## Migration to VisualAge TeamConnection Version 3

### Warning and Error Messages

These are some warning and error messages from the migtc:

- **migrate users:** You may get the following messages because the users are already in the database. They were added when the database was created.

0010-098 The login name InheritedAccess already exists. Specify a different login name.

0010-098 The login name migrate already exists. Specify a different login name.

- **migrate HostView:** The host list entry for the initial superuser was created when the database was created.

0010-055 migrate is already a host list member for user  
zorin.raleigh.ibm.com.

- **migrate Authority:** You will get many of the following messages because the authority tables were loaded when the database was created

6026-009 \*\*\* authority general, "CompView", already loaded.

6026-009 \*\*\* authority general, "DefectView", already loaded.

- **migrate Interest:** You will get many of the following messages because the Interest tables were loaded when the database was created.

6026-009 \*\*\* interest high, "VerifyAssign", already loaded.

6026-009 \*\*\* interest high, "VerifyReject", already loaded.

- **migrate Cfgcomproc:** You will get many of the following messages because the tables were loaded when the database was created.

6026-009 \*\*\* comProc preship, "dsrDefect", already loaded.

6026-009 \*\*\* comProc default, "dsrFeature", already loaded.

- **migrate Cfgrelproc:** You will get many of the following messages because the tables were loaded when the database was created.

6026-009 \*\*\* relProc preship, "approval", already loaded.

6026-009 \*\*\* relProc track\_approval, "approval", already loaded.

- **migrate PartFullView:** You may get the following messages because the particular version of the file has been deleted.

0010-048 The action cannot be completed. Part xxxx has been deleted.

# Migration to VisualAge TeamConnection Version 3

## Obtaining the Tools

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

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

### Notes:

- The tools referenced in this document are available from ftp sites instead of being included in this document so that we can update the tools as necessary.
- VisualAge TeamConnection Enterprise Server Version 3 is available to IBM users through our internal web site.
- The fixpaks for VisualAge TeamConnection are available from the external home page:  
<http://www.software.ibm.com/ad/teamcon/downloads>

### IBM Intranet

#### Web Home Page for TeamConnection

You can access the VisualAge TeamConnection Home Page at:

<http://tc-cmvc.raleigh.ibm.com/>

From the index at the top of the page, select **Tools to migrate from CMVC or VisualAge TeamConnection Version 2 to VisualAge TeamConnection Version 3**.

#### FTP site for TeamConnection

You can download the code from our internal FTP site (an OS/2 host) for TeamConnection, by doing:

1. ftp tc-cmvc.raleigh.ibm.com
2. login as **anonymous** and for password give your email address.
3. cd papers
4. binary
5. get migrate.zip (You can get also PDF files)
6. quit

### Public Internet

#### Web Home Page for TeamConnection

You can access the VisualAge TeamConnection Home Page at:

<http://www.software.ibm.com/ad/teamcon>

#### FTP site for TeamConnection

You can download the code from our external FTP site for 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 migrate.zip (You can get also PDF files)
6. quit

## Migration to VisualAge TeamConnection Version 3

### Obtaining Info-ZIP

The VisualAge TeamConnection team uses the Info-Zip **zip** and **unzip** tools to package compressed files (in which the files to be packaged are compressed first). The migration tools mentioned in this technical report are packaged into a single zip file.

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/`

### How to unzip files

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

## Migration to VisualAge TeamConnection Version 3

### 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
AIX, IBM, CMVC, VisualAge, TeamConnection, RS/6000, DB2 Universal Database, OS/2	IBM Corporation
Unix	Unix System Laboratories, Inc.
Acrobat	Adobe Systems Incorporated
Windows, Windows NT	Microsoft Corporation
Intel	Intel Corporation
Info-ZIP	Info-ZIP Group
ObjectStore	Object Design Inc.

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