IBM VisualAge TeamConnection Enterprise Server

Verifying Installation of TeamConnection

Version 3.0.3

GC34-4742-00

IBM VisualAge TeamConnection Enterprise Server

Verifying Installation of TeamConnection

Version 3.0.3

GC34-4742-00

September 1999

Note
Before using this document, read the general information under “Notices” on page vii.

| This edition applies to fixpack 3.0.3, hotfix 1 of the licensed program IBM VisualAge TeamConnection Enterprise Server and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications by phone or fax. The IBM Software Manufacturing Company takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 284-4721.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

If you have comments about the product, address them to:

IBM Corporation
Attn: Department TH0/Building 062
P.O. Box 12195
Research Triangle Park, NC, USA 27709-2195

You can fax comments to (919) 254-4914.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries:

AIX®	OS/390
C/370™	OS/400
DB2®	PowerPC
IBM®	RISC System/6000
MVS™	RS/6000
MVS/ESA™	SP2
MVS/XA™	TalkLink
OpenEdition®	TeamConnection™
OS/2®	VisualAge®

Lotus and Lotus Notes are registered trademarks and Domino is a trademark of Lotus Development Corporation.

Tivoli, Tivoli Management Environment, and TME 10 are trademarks of Tivoli Systems Inc. in the United States and/or other countries.

The following terms are trademarks of other companies:

HP-UX 9.*, 10.0 and 10.01 for HP 9000 Series 700 and 800 computers are X/Open Company UNIX 93 branded products. HP-UX 10.10 and 10.20 for HP 9000 Series 700 and 800 computers are X/Open Company UNIX 95 branded products.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Intel and Pentium are registered trademarks of Intel Corporation.

Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation.

Java, HotJava, Network File System, NFS, Solaris and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Netscape Navigator is a U.S. trademark of Netscape Communications Corporation.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Reader, and PostScript are trademarks of Adobe Systems Incorporated.

Other company, product, and service names may be trademarks or service marks of others.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM’s valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY, USA 10594.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the Site Counsel, IBM Corporation, P.O. Box 12195, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195, USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

IBM may change this publication, the product described herein, or both. These changes will be incorporated in new editions of the publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

About this book

This book is part of the documentation library supporting the IBM VisualAge TeamConnection Enterprise Server licensed programs. It is written for persons who need to verify that TeamConnection installed properly. Before you use this book, you need to have installed TeamConnection according to the instructions in install.txt.

This book is available in PDF format. Because production time for printed manuals is longer than production time for PDF files, the PDF files may contain more up-to-date information. The PDF files are located in directory path nls\doc\enu (Intel) or softpubs/en_US (UNIX). To view these files, you need a PDF reader such as Acrobat.

Who should read this book

This book assumes familiarity with the operating system on which the TeamConnection server and client are installed. It is intended for network and system administrators. TeamConnection administrators need to be skilled in operating system and database administration. It is also highly recommended that administrators be skilled in DB2 tuning and performance.

Conventions and terminology used in this book

This book uses the following highlighting conventions:

- *Italics* are used to indicate the first occurrence of a word or phrase that is defined in the glossary. They are also used for information that you must replace.
- **Bold** is used to indicate items on the GUI.
- `Monospace` font is used to indicate exactly how you type the information.
- File names follow Intel conventions: **mydir\myfile.txt.** AIX, HP-UX, and Solaris users should render this file name **mydir/myfile.txt.**

Tips or platform specific information is marked in this book as follows:

💡	Shortcut techniques and other tips
OS/2	IBM VisualAge TeamConnection Enterprise Server for OS/2
NT	IBM VisualAge TeamConnection Enterprise Server for Windows/NT
95	IBM VisualAge TeamConnection Enterprise Server for Windows 95
X	IBM VisualAge TeamConnection Enterprise Server for AIX
hp	IBM VisualAge TeamConnection Enterprise Server for HP-UX
◆	IBM VisualAge TeamConnection Enterprise Server for Solaris

Prerequisite and related information

Information on customer service, a glossary, and a bibliography are included at the back of this book.

IBM VisualAge TeamConnection Enterprise Server uses DB2 Universal Database, Enterprise Edition, version 5.2. Refer to the bibliography at the back of this book for a list of publications you can use to install and administer your DB2 database system.

Note: It is not recommended that you make changes to your database by issuing INSERT, UPDATE, or DELETE statements or by changing or deleting database tables or the columns defined in TeamConnection database tables. Changing your database in these ways, through the DB2 administrator tools, the DB2 command line processor, the TeamConnection migration tools, or the tcupdb tool can corrupt your TeamConnection database. Any such changes are made at your own risk. Please contact your IBM representative for information on the terms of IBM customer support.

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other IBM VisualAge TeamConnection Enterprise Server documentation fill out one of the forms at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

Chapter 1. Before you start

This document explains how to verify your installation of TeamConnection servers and clients on all supported platforms. This chapter contains some preliminary information you will need before you start the installation verification process. It includes the following sections:

- Hardware and software requirements for each platform
- Information on setting TCP/IP addresses and ports
- Information on setting up DB2 for use with TeamConnection

To install TeamConnection and create families, you should be a system administrator for your operating system, be able to login as user root (on UNIX operating systems), and be able to create user accounts. You should also be a DB2 administrator with skills in DB2 performance and tuning.

Review hardware and software requirements

For a current list of hardware and software requirements for VisualAge TeamConnection enterprise Server, open your Web browser on the following Web address:

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

Select either **Software Requirements** or **Hardware Requirements**.

DB2 Universal Database

The TeamConnection server requires DB2 Universal Database version 5.2. During TeamConnection server installation, you have the option of installing the appropriate edition of DB2 on your server, if you do not already have it installed.

TeamConnection installed directory structure

After you install the TeamConnection code in its directory structure (represented by $TC_HOME), please keep in mind the following guidelines:

- Do not remove directories, files, or symbolic links from $TC_HOME, unless you have uninstalled the product.
- Do not change the file permissions or the ownership of the directories or files in $TC_HOME.
- Use $TC_HOME only to store the TeamConnection code. Do not use it as the home directory for users and do not use it as the home directory for TeamConnection families.
- If this is your first installation of TeamConnection and you are using the tar images (for UNIX platforms) to install it, accept the default values suggested in the installation instructions and scripts.

Updating TCP/IP files

If you use wizards to install TeamConnection, then your TCP/IP services and hosts files are set up for you and you can skip this section.

Before you install the TeamConnection code, you must have the correct version of TCP/IP installed on your workstation. See “Review hardware and software requirements” on page 1 for the communications software requirements for your operating system.

After TCP/IP is installed, update your TCP/IP services and hosts files.

You can update these files using smit. In many installations, a name server is used instead of /etc/hosts. Also, many installations distribute /etc/services. smit can be used to make the necessary updates.

You can update these files using sam. In many installations, a name server is used instead of /etc/hosts. Also, many installations distribute /etc/services. You can use the sam tool to make the necessary updates.

Do the following steps.

1. Update the services file, which is located in \etc\services in the directory where TCP/IP is installed.

   To determine the directory name, type `echo %etc%` at a prompt.

   If you have a services file, it is located in the system32/drivers/etc subdirectory of the Windows NT installation directory. If the services file does not exist, you must configure it through TCP/IP.

   Include the family name and port address of the TeamConnectionserver. The port address can be any 4-digit number, as long as it does not already exist in your services file. You might want to ask your TCP/IP administrator to assign you a number.

   Type the following entry in your services file. Replace *ffff* with an appropriate port address. Follow the line with a carriage return.

   ```
   #  TeamConnection servers
   testfam    ffff/tcp     # port address for the TeamConnection test family
   ```
2. Update the TCP/IP hosts file, which is located in \etc\hosts.

   To determine the directory name, type `echo %etc%` at a prompt. If the hosts file does not exist, you must configure it through TCP/IP.

   If you have a hosts file, it is located in the system32/drivers/etc subdirectory or the Windows NT installation directory. If the hosts file does not exist, you must configure it through TCP/IP.

Add the following:
- IP address.
- Server name.
- Alias name of the TeamConnection family server, which is your family name. For the initial installation of TeamConnection, the family name is *testfam*.
- Alias name for the *build socket*. For the initial installation of TeamConnection, use *bldsock*.

The following is an example of the entry you would type in your hosts file. Follow the line with a carriage return. You can use the hostname command to get the name of the server.

```
9.12.345.67   teamserv.company.com   testfam    bldsock
```

3. Do the following to verify that the hosts file is specified correctly:
   - Type `host` *`family_name`*, where *family_name* is the name of your TeamConnection family.

The tchostw.exe utility is available from the samples directory on the TeamConnection CD-ROM.

Type `tchostw` *`family_name`*, where *family_name* is the name of your TeamConnection family.

The tchost.hp utility is available from the misc directory on the TeamConnection CD-ROM.

Type `tchost.hp` *`family_name`*, where *family_name* is the name of your TeamConnection family.

The tchost.sol utility is available from the misc directory in the CD-ROM for TeamConnection for UNIX.

Type `tchost.sol` *`family_name`*, where *family_name* is the name of your TeamConnection family.

The information returned should match the number and name specified in your hosts file entry. For example, using the entry given in the previous step, the system response would be as follows:

```
teamserv.company.com = 9.12.345.67
```

   - Type `host` *`ip_address`*, where *ip_address* is the IP address of your machine.

Type `tchostw` *`ip_address`*, where *ip_address* is the IP address of your machine.

Type `tchost.hp` *`ip_address`*, where *ip_address* is the IP address of your machine.

Type `tchost.sol` *`ip_address`*, where *ip_address* is the IP address of your machine.

The information returned should match the number and name specified in your hosts file entry. For example, again using the entry given in the previous step, the system response would be as follows:

```
9.12.345.67 = teamserv.company.com
```

If you do not receive the expected response, contact your TCP/IP administrator to solve the problem.

If the servers are defined in the domain name server, then you can use the UNIX utility ″nslookup″ instead.

4. Do the following to verify that you can connect to your TeamConnection family:
   a. At a prompt, type `ping testfam`.
   b. Press Ctrl+C to end the command.

   a. At a prompt, type `ping -s testfam`.
   b. Press Ctrl+C to end the command.

   If you receive information that is similar to the following, you can successfully connect to your TeamConnection family:

```
PING teamserv.company.com: 56 data bytes
64 bytes from 1.23.457.78: icmp_seg:0. time=0. ms
64 bytes from 1.23.456.78: icmp_seg:1. time=0. ms
64 bytes from 1.23.456.78: icmp_seg:2. time=0. ms
```

   If you receive information that is similar to the following, you can successfully connect to your TeamConnection family:

```
PINGING teamserv.company.com (9.12.345.67): with 32 bytes of data:
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
```

   The PING command will send four requests and then it will stop.

   If you receive the message `unknown host testfam`, you cannot connect to the family. Verify that the data you entered in the hosts and services files is correct, and then try the command again. If you still do not get the correct response, contact your TCP/IP administrator to solve the problem.

5. Do the following to verify that you can connect to your TeamConnection build server:
   a. At a prompt, type `ping bldsock`.
   b. Press Ctrl+C to end the command.

   a. At a prompt, type `ping -s bldsock`.
   b. Press Ctrl+C to end the command.

   If you receive information that is similar to the following, you can successfully connect to your TeamConnection family:

```
PING teamserv.company.com: 56 data bytes
64 bytes from 1.23.457.78: icmp_seg:0. time=0. ms
64 bytes from 1.23.456.78: icmp_seg:1. time=0. ms
64 bytes from 1.23.456.78: icmp_seg:2. time=0. ms
```

NT

If you receive information that is similar to the following, you can successfully connect to your TeamConnection family:

```
PINGING teamserv.company.com (9.12.345.67): with 32 bytes of data:
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
Reply from 9.12.345.67: bytes=32 time=10ms TTL=255
```

The PING command will send four requests and then it will stop.

If you receive the message `unknown host bldsock`, you cannot connect to the build server. Verify that the data you entered in the hosts and services files is correct, and then try the command again. If you still do not get the correct response, contact your TCP/IP administrator to solve the problem.

Note: Do not install the TeamConnection components until the ping commands successfully complete.

Setting up login IDs

Certain login IDs required for your TeamConnection family need to be coordinated. You need to create a system login, and a TeamConnection initial superuser login that are the same as your family name. In addition, the DB2 user with administrator authority needs to be the same as the TeamConnection superuser ID. If you create a family called testfam, for example, then you also need to create a system login called testfam, an initial TeamConnection superuser ID (created when you create your family) called testfam, and a DB2 user called testfam with administrator authority.

For TeamConnection to function properly, you need to login to your system and start your TeamConnection family using these coordinated login IDs. For example, for a family called testfam on a Windows NT server, you would do the following:

1. Login to Windows NT using the login ID **testfam**.
2. Grant that login ID DB2 administrator authority.
3. Create a TeamConnection family called **testfam** with an initial superuser ID called **testfam**.
4. Start the family while logged into Windows NT as user **testfam**.

On UNIX platforms, the primary group ID for the VisualAge TeamConnection family user ID must be the same as the primary group ID (the default is db2iadm1) for the DB2 instance (the default is testfam) to be used for the family. If this is not properly done, then the family user ID will lack authority to create the database for the family. Use only lowercase names for user accounts.

Setting up DB2

DB2 instances

Each family has its own unique database. If you create more than one family on a single machine, they use the same database manager to access the database. If you want to install families on separate machines, you need a TeamConnection and DB2 server on each machine.

It is recommended that the databases for each TeamConnection family be placed in a separate DB2 instance. By following this recommendation, you can assure that if an instance is stopped, only one TeamConnection family is affected. It also enables you to tune the performance for one instance while affecting only one TeamConnection family.

You need to have at least 100 MB of free disk space in the file system where the family database is to be created.

On Intel platforms, it is recommended that you create only one DB2 instance per (physical) server. Because it is recommended that you have only one family per instance, you should have only one family per (physical) server.

On UNIX platforms, use the sample .profile for each family that you create. This .profile contains DB2 environment variables that you need to customize before you create a family. One of these variables, DB2INSTANCE, defines the DB2 instance in which the family is to be created. DB2INSTANCE must be set in the profile for the new family and should point to the new DB2 instance. The sample profile is located in `$TC_HOME/install/$LANG/profile.family`.

If you already have a family running on an existing DB2 instance, it is necessary to create another DB2 instance for a new family. Refer to *DB2 Quick Beginnings* for information on how to create a new DB2 instance.

Verifying that the database server is active

On Intel platforms, you can start and stop the database manager using the db2start and db2stop commands or using the DB2 Control Center. From the Control Center, select a database manager instance and then select **Start** or **Stop** from the **Selected** menu. Refer to *IBM DB2 Universal Database Administration Getting Started* or the *IBM DB2 Universal Database Administration Guide* for more information.

On UNIX platforms, you can start and stop the database server using the db2start and db2stop commands. Login to the system as the DB2 instance owner. Refer to the *IBM DB2 Universal Database Administration Guide* for information.

Chapter 2. Verifying the installation of TeamConnection

If you used the Server Setup wizard to verify your installation of TeamConnection, you can skip this chapter.

To verify that TeamConnection has installed properly, that the TCP/IP settings are set properly, and that DB2 has been set up properly, you will create a test family called testfam. You can use testfam to explore and learn about TeamConnection. Using testfam with the shipped default information will help you determine how to customize TeamConnection to fit your development needs. When you create your family, you are loading default information shipped by IBM and creating a superuser ID. A superuser ID is required so that at least one person has privileged access to the family to perform special tasks, such as creating other user IDs.

Before running the installation verification procedures in this chapter, verify that the TCP/IP hosts and services files have been set according to the instructions in "Chapter 1. Before you start" on page 1.

Configuring TeamConnection environment variables

This task is not necessary on Intel platforms.

Environment variables in the .profile of the family account need to be configured before you create a TeamConnection family.

It is recommended that that the .profile for the family be based on the sample profile.family. This profile uses all the necessary environment variables to create an use a TeamConnection family.

1. Login as the user account for the TeamConnection family.
2. Use the following commands to copy the sample profile.family as your new .profile, substituting the identifier for the national language version of TeamConnection you are using (such as en_US) for $LANG.
   - For AIX:
     ```
     mv $HOME/.profile $HOME/.profile.original
     cp /usr/teamc/install/$LANG/profile.family $HOME/.profile
     chmod u+w .profile
     ```
   - For HP-UX and Solaris:
     ```
     mv $HOME/.profile $HOME/.profile.original
     cp opt/teamc/install/$LANG/profile.family $HOME/.profile
     chmod u+w .profile
     ```
3. Customize all appropriate entries in the new .profile.

   Edit the new .profile, read the entire file, modify the values as appropriate, and uncomment variables necessary for your operating system.

   Note: If you are using the CDE environment, edit .dtprofile to enable the account to run .profile after .dtprofile. CDE stands for Common Desktop Environment and is the default for AIX 4, HP-UX 10, Solaris, and many other versions of UNIX. CDE is the standard of the Open Group®. For instructions, see the header of the sample profile.

4. For AIX with CDE only, add the following line to the $HOME/.Xdefaults file:

```
Dtterm*loginShell: true
```

5. You will likely want to restrict access to this account. Be sure that your .profile (and .dtprofile in the CDE environment) have no visibility outside of the user account. To restrict access to the account, issue the following command:

```
chmod u=rw,go= .profile .dtprofile
```

6. Logoff and login as the TeamConnection family to activate the changes to the new profile.

Creating testfam

To create a test family to be used for installation verification, follow these steps.

To start the family administrator program, select the TeamConnection Family Administrator from the TeamConnection group.

To start the family administrator program, type **tcadmin** from a command prompt.

1. From the TeamConnection Family Administrator window, select **Family** → **Create Family**.

   A properties notebook opens. You use this window to set up your test family. This notebook contains one page on which you enter required information for your test family.

   After your family is created, you can access other pages in this notebook by selecting the family and then selecting **Family** → **Properties**. Refer to the *Administrator's Guide* for more information on all of the family options you can set using this notebook.

   Note: In these instructions, the values you can enter appear as follows. TeamConnection's installation verification procedure requires that some of these values be set exactly as shown.
   - **Boldface** indicates values you need to enter exactly as shown.
   - *Italics* indicates values that you can substitute to suit your needs or local conventions.

2. Complete the fields in the **Family** section of this page as follows:

   Name **testfam**

   Path Specify *d:\dbpath* for Intel platforms or **/home** for UNIX platforms.

   Specify the fully-qualified path name of the directory where you want testfam stored. Make sure this directory does not already exist. TeamConnection creates testfam in a subdirectory of the path you specify. This subdirectory has the same name as the family. If you specify **c:\proddev** (Intel) or **/home/proddev** (UNIX) as the path name for testfam, for example, TeamConnection places all files related to the family in the directory path c:\proddev\yourDBName (Intel) or /home/proddev/yourDBName (UNIX).

Note: If a testfam directory already exists, you will need to delete it before you proceed. This procedure will fail if a testfam directory already exists.

Port *xxxx*

Specify the TCP/IP port address for testfam, which was set in your TCP/IP services file before you installed the product.

Mailer
mailexit

3. Complete the fields in the **Security for TeamConnection Users** section of this page as follows:

Security Level

Specify the level of security to be used for the family. You can choose from one of the following:

Host only
A valid combination of the system login ID, TeamConnection user ID, and host name must be used to obtain access to the family. This is the default level of security.

Password only
A user must login to and logoff of TeamConnection and supply a password in one of the following ways:

- Select **Login** from the **File** menu of the Tasks window.
- Issue the command `teamc tclogin` from a command prompt.

When the user logs in to the family, the family will send back a token associated with that user from that client. The server will check the attached token and, if valid, will proceed to perform the requested action.

Password or host
The user can use either the Password only function if he or she has a password or the Host only function if he or she has a valid host list entry. This level of security is useful for teams in which particular team members may be remote or mobile and have changing IP addresses. If the user supplies a valid password, then TeamConnection uses the password to admit access to the family. If the user either does not supply a password or supplies an incorrect password, then TeamConnection checks the user's host list entry to admit access.

None Any user can access TeamConnection. Neither a password nor a valid host list entry is required.

Password minimum length
Use this field to set the minimum number of characters to be used for passwords. The minimum is 1, the default is 8, and the maximum value is 32.

Maximum Invalid Attempts
Use this field to set the number of times users can attempt to login before TeamConnection deactivates the user's ID. If this happens, a superuser must reactivate the ID before the user can attempt to login again.

4. Complete the fields in the **SuperUser** section of this page as follows:

Login *userID*

Specify the system login ID for the superuser for the family. For single-user platforms (OS/2), use the value set for the TC_USER environment variable. To see this value, type the following from a command prompt and look for the TC_USER variable:

```
set | more
```

For multiuser platforms (AIX, HP-UX, Solaris, Windows NT) specify the user's system login ID.

Userid

| Specify the TeamConnection user ID for the superuser. If you omit this parameter, it defaults to the value specified in the **Login** field (for multi-user platforms) or to the value of the TC_USER environment variable (for single-user platforms). It is recommended that you create a superuser ID that is the same as both the system login ID from which TeamConnection will be started and the family name.

Host *hostname*

Specify the TCP/IP host name for the family server machine, which was set in your TCP/IP hosts file before you installed the product.

To see this value, type the following from a command prompt:

```
hostname
```

Note: You do not need to use the fully-qualified host name. You can, for example, specify *myServer* instead of *myServer.myCompany.com*.

Password

If you want to use password security, you must specify the password to be used to verify the superuser's access to the TeamConnection server. If you do not specify the password for superuser access, then no one will be able to access the database. To use password security, you need to set the **Security level** field on the **Security for TeamConnection Users** section of this page to **Password only** or **Password or host**.

5. After you have set all the values on the **Required** page of the properties notebook, select the **OK** push button.

   As TeamConnection creates testfam, the **Create Family** window appears showing the tasks TeamConnection performs as it creates a family. This window also contains a progress bar showing you how far along the process is.

After testfam is created, you should be able to see the following:
- A family administrator icon labeled testfam in the **TeamConnection Family Administrator** window.
- A subdirectory called testfam in the directory path that you specified on the family settings notebook.

  This subdirectory, in turn, contains the following:

  authorit.ld
  The default authority groups for the family

  cfgField
  A subdirectory containing default configurable field information

  comproc.ld
  The default component processes for the family

config A subdirectory containing default user exit information for the family and security information

config.ld
The default configurable field types for the family

interest.ld
The default interest groups for the family

queue A subdirectory for the messages to be sent by the notify daemon

relproc.ld
The default release processes for the family

- You should also be able to see your new database in the DB2 Control Center (on OS/2 and Windows platforms) and in the DB2 directory.

Backup your new database using the backup utilities described in the *IBM DB2 Universal Database Administration Guide*. Review the database maintenance utilities and plan your backup and recovery strategy.

Start the family server

NT On Windows NT it is important to start your family while logged in to the same user account under which the family was created. Otherwise you will receive an SQL error -727. The login ID under which the family was created is used as the table schema name when the various tables TeamConnection uses are built. All queries against the family expect to see this login ID.

You can start the TeamConnection server in the following ways:

- Double-click the testfam icon in the TeamConnection Family Administrator window or select the icon and then select **Open Server** from the pop-up menu. In the **testfam - Family Servers** window, select the **Start Both Servers** push button.

  After the family and notification servers are running, you can minimize the **testfam - Family Servers** window, but do not close it. Closing this window stops the servers.
- From a command prompt, change to the directory path containing the testfam database and type the following command. Use of this command assumes that the path statements and environment variables have been set properly.

  ```
  teamcd testfam
  ```

Refer to the *Administrator's Guide* for information about the teamcd command.

Ensure that the TeamConnection client is accessible

Because the verification step uses the TeamConnection client, you need to ensure that it is accessible:

1. Type the following TeamConnection command, which does not require connection to a working TeamConnection family.

   ```
   teamc report -testClient
   ```
2. If the teamc command is not found, then ensure that the TC_HOME and PATH variables are properly set.
3. If the teamc command is found, then verify that the top portion of the output is something like this, assuming that the LANG variable is en_US:

```
Product Version:                      3.0.3
Message catalog language:             English
Environment variables:
Variable:             Setting:
--------------------- ----------------------------------
TC_BECOME             <your TeamConnection user id>

TC_FAMILY             <your TeamConnection family name>
```

4. If you see that the entry for ″Message catalog language″ has the value `English (Internal)`, then the LANG or NLSPATH variables are not properly set and the TeamConnection client is using its internal messages. It is strongly recommended that the external message catalog be used.

Run the installation verification procedure

The following installation verification procedure runs a command file that performs a series of TeamConnection actions on testfam.

OS/2	Executing this command file requires that you have a full build environment set up, including VisualAge C++ and the TeamConnection build server. If you have not installed the build function or VisualAge C++, run the installation verification procedure with the -nobuild option.
NT	Windows/NT requires Microsoft Visual C++ to run the installation verification procedure. If you have not installed the build function or VisualAge C++, run the installation verification procedure with the -nobuild option.
X hp	On UNIX platforms, the installation verification procedure does not include build and packaging functions. It executes TeamConnection user, host, part, workarea, release, defect, feature, and other commands that do not require the build and packaging functions.

The installation verification procedure does the following:
- Ensures that all TeamConnection daemons are stopped (using tcstop)
- Starts the family daemon and a build server (using teamcd and temporary TCP/IP socket numbers)
- Runs a suite of TeamConnection actions

To verify that TeamConnection installed correctly, do the following:
1. Type the following at a prompt in the directory where TeamConnection is installed, and then press **Enter**. On OS/2 and Windows, you can use the -nobuild option to run the procedure without verifying the build function.

OS/2	`univos2 <-nobuild>`
NT	`univwin <-nobuild>`

```
univunix
```

For more information on using this command, you can enter it with the parameter `-help`.

2. Type **yes** when you are asked if you want to start the build server. When you are asked to enter the build server socket name, type **bldsock**. This is the build socket name that is specified in the hosts file.

If you did not install the build function or you do not want to test it, use Ctrl-Break to stop the installation verification procedure.

On UNIX platforms, the installation verification procedure does not include the build function.

When this command completes, it creates a log file named as follows:

univos2.log

univwin.log

univunix.log

3. Review the log file for a completion code of zero on the commands it executes. If all commands execute successfully, then TeamConnection was installed and configured correctly. A non-zero return code may indicate problems with the installation or configuration of TeamConnection.

Starting the TeamConnection client

To start the TeamConnection client, select the **TeamConnection Client** icon from the TeamConnection Group. The Tasks window appears.

Before you start the client GUI for the first time, do the following:

1. Copy the sample initial tasks list for the main window. You will need to do this only once:

   ```
   cp $TC_HOME/nls/cfg/$LANG/teamcv3x.ini $HOME/.
   chmod u+w $HOME/teamcv3x.ini
   ```

   Where $TC_HOME is the location where the TeamConnection code was installed.
2. If you want to change the font type and size used by the TeamConnection GUI, copy the sample teamcgui local resource file and then edit and customize it. You will need to do this only once. To copy the file, issue the following commands:

   ```
   cp $TC_HOME/nls/cfg/$LANG/Teamcgui.user $HOME/Teamcgui
   chmod u+w $HOME/Teamcgui
   ```
3. Start the TeamConnection GUI by issuing the following command:

   ```
   teamcgui &
   ```

You can issue TeamConnection line commands from the family account. To see a list of all users, for example, issue the following command:

```
teamc report -view users -where 1=1
```

What do you do now?

You now have your initial family installed and running. As mentioned earlier in this chapter, you or someone else in your organization can use this family to verify that TeamConnection is working properly. You can also use this family to explore and learn about TeamConnection, and test configuration changes, user exits, automated backup utilities, and automated build and packaging events.

To learn more about planning for and creating your TeamConnection production family, refer to the *Administrator's Guide*.

Chapter 3. Migrating to TeamConnection version 3

TeamConnection provides tools for migrating from TeamConnection version 2 to version 3 (called migtc) and from CMVC to TeamConnection (called migcmvc). This chapter explains how to use these tools.

Note: If you are currently using TeamConnection version 1 and you want to migrate to version 3, you must first migrate to TeamConnection version 2. Refer to the version 2 documentation for instructions.

This chapter contains the following sections:
- Migrating from TeamConnection version 2 to TeamConnection version 3
- Migrating from CMVC to TeamConnection
- Using the migration tools

If you are currently using TeamConnection version 2 or CMVC, and you want to begin using TeamConnection version 3, then you must migrate your database to TeamConnection version 3.
- For TeamConnection version 2 customers, because TeamConnection version 3 uses DB2 Universal Database instead of ObjectStore, you cannot simply install version 3 over your version 2 installation. Instead, you need to install TeamConnection version 3 on a different server from your version 2 server and migrate your version 2 database to the version 3 server using the migtc tool.
- For CMVC customers, although the TeamConnection product was developed as the next generation of CMVC, much of the underlying functionality has changed. For this reason, you need to migrate your CMVC database to TeamConnection version 3. To migrate your database from CMVC to TeamConnection, you need to install TeamConnection version 3 on a different server from your CMVC server and migrate your CMVC database to the version 3 server using the migcmvc tool.

Refer to “Using the migration tools” on page 43 to learn more about the migration commands before beginning the migration process.

The following list of terms used in this chapter will help you understand the migration process:

original database
The original TeamConnection version 2 or CMVC database before migration.

migrated database
The new TeamConnection version 3 database after migration.

source server
The original family server (TeamConnection version 2 or CMVC) from which you are migrating your database.

target server
The new TeamConnection version 3 family server to which you are migrating your database.

new source server
A relocated source server. This term is used to refer to a source server that has been moved from its original production machine.

It is recommended that you make a backup copy of your family database and file structure before you begin migration. After you have backed up your family database, restart it in maintenance mode so that no updates can be made to it from client machines while the migration is being performed.

Migrating from TeamConnection version 2 to version 3

TeamConnection version 3 cannot be installed over any previous version of TeamConnection. Instead you need to migrate your TeamConnection version 2 database to a new TeamConnection version 3 server. Existing TeamConnection customers must migrate their data if they want to be able to use it on TeamConnection version 3.

Note: The following configuration is not necessary on UNIX platforms, but suggested.

To migrate to TeamConnection version 3, you need two separate server machines:

- A TeamConnection version 2.0.9 (or later) family server with your original database. This server must also have a superuser ID and a host list for the superuser so that the target server can access the source server.

  You must have TeamConnection version 2.0.9 (or later) installed before you begin migration.
- A separate TeamConnection family server with TeamConnection version 3 client and server components installed, to which you will migrate your database. This server must be installed and configured as described in the installation chapters of this book.

You can migrate your TeamConnection version 2 database to any platform supported by TeamConnection version 3.

General guidelines

The user ID of the person responsible for the migration must have superuser authority on the source server to guarantee access to all data for migration. This authority will be migrated from the source database to the target database. The superuser must have a host list entry to the target server.

It is recommended that the migration be done in stages with frequent backups of the database. You can use where clauses to capture and migrate a limited or expanded set of data. With frequent backups, if a problem occurs, the database can be reset and the problem area rerun once fixes have been made.

If you need to migrate fine-grained data to version 3.0, there some additional considerations. Fine-grained data includes information stored in TeamConnection that is not a file, such as DataDictionary objects or VisualAge Generator objects.

- The user performing the migration must have write access to the current working directory of the target server.

- There must be enough free space available on the target server for the .cdf files containing the fine grained data for the releases and for a small script file (*releaseName*FG.ksh or *releaseName*FG.cmd). During fine-grained data migration, one script file is generated for each release containing fine-grained data. These script files contain commands that export fine-grained data from the original database into .cdf files and import it from the .cdf files into the migrated database.

Preparing the source server for migration

There are several steps you must take before you begin migrating to help the migration process execute more smoothly:

- If you need to migrate fine-grained data to version 3.0, set the environment variable TC_MIGRATE_EXPORT=1 before you start your version 2.0 database. This environment variable causes only fine-grained data to be exported to .cdf files during fine-grained migration.
- All drivers and workareas should be integrated and all drivers committed before beginning migration. All test records and verify records should be completed.
- CollisionViews are not migrated. In the 2.0 database, if a part is checked out in two different workareas, you need to resolve all collision records before migrating the part. Refer to the *User's Guide* for instructions on part reconciliation.
- Create a user ID with superuser access to your original and migrated databases and a host list for the user ID. This ID will be used to access the database from the target server.
- Ensure your version 2.0 server is running with the latest fixpaks and the correct versions of the executables:
  1. Stop the TeamConnection version 2.0 family.
  2. Download hotfix3 from the TeamConnection web site <http://www.software.ibm.com/ad/teamcon/downloads/>. From this web page select **previous fixpaks** and then select the following directories and files:

     For UNIX:
     aix/fixpak209/hotfixes/98–06–24.tar

     For Intel:
     intel/hotfixes/ihotfix3.zip
  3. From your version 2.0 database, set the TC_VIEWS environment variable to point to the file allv2.idl in your family database directory (defined by the TC_DBPATH environment variable)

     For UNIX:
     export TC_VIEWS=$DB_PATH/allv2.idl

     For Intel
     set TC_VIEWS=*tc_dbpath*\allv2.idl
  4. Edit the version 2.0 security file (located in the config subdirectory of your TC_DBPATH) and set the security level to NONE. The following is an example:

     ```
     authentication level     = NONE
     ```
  5. Copy allv2.idl from the TeamConnection version 3.0.1 CD to your TeamConnection version 2.0 TC_DBPATH.

For UNIX:
allv2.idl is located in the /misc directory of the CD

For Intel:
allv2.idl is located in the \samples directory of the CD

6. Replace teamcd.exe in the bin subdirectory of your version 2.0 installation path with the version of teamcd.exe from hotfix3.
7. Restart the TeamConnection version 2.0 family in debug mode using the following command:
   ```
   teamcd -d familyName 2
   ```
8. Verify that you are logged on with the superuser user ID.
9. Verify that the new view (allv2.idl) is loaded properly by issuing the following command:
   ```
   tcselect describe allversions2
   ```
10. On UNIX platforms, if there are parts in the 2.0 database that do not have an fmode, they will be extracted during migration with the "sticky" bit set. Migration will then fail while trying to run. To prevent this error, do the following for each release to be migrated
    a. Run the following report on the 2.0 database and replace *yourRelease* with the name of the release:
    ```
    teamc report -view partview -release  yourRelease -where "fmode is null"
    ```
    b. For each part returned by this query, issue the following part -modify command and replace *partName* with the name of each part and *yourRelease* with the release it is located in:
    ```
    teamc part -modify partName -release yourRelease  -fmode 0777
    ```
11. On UNIX platforms, use the **chmod** command to set file permissions for the executables to 777.
12. Use the **which teamcd** or **what teamcd** command to verify that you are running with the correct versions of the executable (that it has the same date/time stamp as the one from hotfix3).

- When migrating a single, committed version of each file from TeamConnection version 2, you must first do a `release -extract` or `part -extract` to get the files you want to migrate out of the source database. You must then move the files to the TeamConnection version 3 server machine to migrate them into the target database. You can move the files by using a copy command, ftp, or any other available method.
- You may want to run queries like the following and save the output to a file. These commands capture the data that will be migrated. You can experiment with where clauses to limit the objects to be migrated.

  Note: These queries may result in very large output files.
  ```
  teamc report -view workareaView >workare1.vew


  teamc report -view releaseView >release1.vew


  teamc report -view compView >compon1.vew


  teamc report -view defectView >defect1.vew


  teamc report -view featureView >feature1.vew
  ```

- Stop the family server and use the `xcopy` command with the `/s` and `/e` parameters (in Intel), `cp -r` in UNIX environments, or an equivalent command, to create a backup copy of your TC_DBPATH directory structure and all files in it (including all .ld files, all .fmt and .tbl files). Make sure these files do not get copied to your target server.
- Restart the family server in maintenance mode (using the -m option) so that no updates can be made to the server while migration is in progress. Client users can view information, but not change it.

TeamConnection version 2 databases are not portable. It is recommended that you keep the TeamConnection version 2 server installed and operational on its original machine. If you do move the version 2 database to a different machine prior to migration, please do the following:

- Use xcopy, cp, or other environment-appropriate command to copy the files in the source server's TC_DBPATH directory to the new source server. Use the exact same TC_DBPATH. The drive and directory for the database must exactly match the original.
- On the new source server, on the first line of the TCP/IP hosts file, place the IP address of the server followed by the hostname and address of the source server, a space, and then the alias, as follows:

  ```
  1.11.111.111  testfam.company.com  testfam
  ```

If you plan to use your source server as your TeamConnection version 3 server, then you need a new IP address and hostname for this machine, since its original hostname has been transferred to the new source server.

Setting up the target server

You run the migration tools from the TeamConnection client component of the target server, which communicates with your source server to migrate the database.

Install TeamConnection version 3 on your target server according to the instructions in the installation chapters of this book.

If TeamConnection version 2 is installed on the machine you intend to use as your target server, you must uninstall it before installing TeamConnection version 3 and then reboot. If you remove a TeamConnection version 2 installation, you may have to manually delete some files from the TeamConnection version 2 installation directory.

- Install the TeamConnection version 3 client and server components and perform the installation verification procedure to create the testfam database. Make sure you complete the installation process, including updating your TCP/IP hosts and services files with IP addresses for your target family.
- Set or modify the following environment variables in the Environment page of System Properties (Windows NT) config.sys (OS/2) or .profile (UNIX) on the target server:

  TC_FAMILY=*migratedDatabaseName*

  Set this variable to the name of your migrated family database on the target server.

TC_FAMILY_CLIENT=*originalDatabaseName@hostname@port*
: Set this variable to the family name, host name, and port ID of your original database on the source server.

TC_DBPATH=*home directory*
: Set this variable to the home directory of your target database.

TC_USER=*superuserID*
: Set this variable to a user ID with superuser authority.

TC_BECOME= *superuserID*
: Set this variable to a user ID with superuser authority.

TC_PFV_NUPATH=1

The following are examples of these environment variables on Intel, using v2dbase as the original family name and v3dbase as the migrated family name:

```
SET TC_FAMILY=v3dbase


SET TC_FAMILY_CLIENT=v2dbase@v2server@2222


SET DB_PATH=d:\v3dbase


SET TC_USER=migrator


SET TC_BECOME=migrator

SET TC_PFV_NUPATH=1
```

- OS/2 NT Reboot the machine.

- Determine the directory or path where the target database will be located. This directory path will be your TC_DBPATH. This path is specified when your database administrator uses the TeamConnection Administration GUI to create the new database instance.

  Note: The TeamConnection administrator is also responsible for migrating the .ld files after migration. See “Tables for authority, interest, configurable fields, and processes” on page 28.
- Verify that your version 2 server and ObjectStore are running on the source machine.
- Verify that the DB2 database instance is running on your target server.
- Verify that the parts extracted from the source server have been copied to your target server.

Performing the migration

Once you have prepared your source and target servers, you are ready to perform the migration. The TeamConnection migration utility, migtc, is a command-line utility that you run from the TeamConnection client and server installed on your target server. It uses the information you provided in “Setting up the target server” on page 19 to locate and access the source database.

TeamConnection provides a file called migtc.lst containing sample migration commands. This file is located in the /bin subdirectory of your TeamConnection version 3 installation path. It may be helpful for you to start with this command file, modify it to suit your needs, and then execute the migration using this command file.

Migrating the most current committed version of objects

The following are sample contents of migtc.lst. The commands in this file are written for the following migration functions:

- All objects are migrated, for example, all users, all parts, all defects, and so on.
- Only the most current committed versions of parts (files) are migrated.

If you want to select specific objects to be migrated, you will need to modify this command file by adding `where` clauses to the commands. For example, if you have deleted parts from a release, you may choose not to migrate those parts by using a `where dropDate is null` clause. See “Using the migration tools” on page 43 for information on specific migration commands.

Note: Depending on the size and complexity of the database you are migrating, it may be advisable to set maxerrors to a value larger than 100.

```
set batchsize 100

set decache   1

set maxerrors 100

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'

migrate   ReleaseView
migrate   EnvView

migrate   AccessView

migrate   NotifyView

migrate   DefectView

migrate   FeatureView

migrate   BuilderView

migrate   ParserView

# migrate   DriverView where state in ('complete', 'commit')

# migrate   WorkAreaView where state in ('commit','complete')

set  top  x:\sourceCodeTreeStructureforReleaseName1

migrate   PartView -release yourReleaseName1

set  top  x:\sourceCodeTreeStructureforReleaseName2

migrate   PartView -release yourReleaseName2

# migrate   FixView

# migrate   ApproverView

# migrate   ApprovalView

# migrate   SizeView

# migrate   TestView

# migrate   DriverMemberView

migrate   VerifyView

migrate   NoteView

# migrate   CoreqView
# migrate FineGrained -release releaseName

quit
```

To use this command file for your migration, you need to make the following minimal modifications. For a complete list of migration commands, see "Using the migration tools" on page 43.

- The `migrate` commands that are preceded by the # character are optional. To activate the commands, uncomment them (delete the # character) as needed.
- It is wise to divide this command file into several smaller files, such as migtc1.lst, migtc2.lst, and so on, and to back up your database after executing each chunk. Chunking this command file will enable you to backup your database at regular intervals during the migration.
- The `set top` command is used to point to the source code tree structure of parts when you migrate only the current version of parts. If you want to migrate only the current version of parts, do the following:
  1. Create the source code tree structure on the target server or on a LAN drive that the target server has read/write access to.
  2. Change the `set top` variable in migtc.lst to point to the directory structure.
- The following commands migrate TeamConnection parts. Modify these commands to include the release names defined in your original database. If you have only one release, delete one of these commands. If you have more than two, create additional `migrate PartView` commands.
  ```
  migrate   PartView -release yourReleaseName1

  migrate   PartView -release yourReleaseName2
  ```
- If your original database contains fine-grained data, you need to include one `migrate FineGrained` command for each release that contains fine-grained data. Specify the release names using the `-release` *`releaseName`* attribute.

Migrating previous versions

It is not recommended that you migrate all of your past versions, as this can be very time consuming, and you will most likely not use all of this information. Instead, keep your version 2 database as an archive and retrieve previous versions of parts when you need them. If you want to migrate multiple versions, however, you must have TeamConnection Version 2.0.9 or later installed.

To migrate previous versions, you need to create two migration command files containing the following commands. Make the minimum required changes to these command files as described previously. The `set top` variable must remain null, as shown in this example.

Note: Depending on the size and complexity of the database you are migrating, it may be advisable to set maxerrors to a value larger than 100.

```
set top

set maxErrors 100

set decache 1
```

```
set batchsize 100

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'

migrate releaseView

migrate envView

migrate accessView

migrate notifyView

migrate defectView

migrate featureView

migrate builderview

migrate parserview

migrate driverview

migrate workareaview
```

Execute the commands in this command file and then quit and save your database. See "Executing migtc.lst" on page 26 for instructions. Then execute the second command file containing the following commands:

```
migrate versionview where 1=1 order by adddate


migrate partfullview -release  oneOfYourReleaseNames
migrate partfullview -release  anotherRelease

migrate partfullview -release  anotherRelease

migrate partview -release oneOfYourReleaseNames

migrate partview -release anotherRelease

migrate partview -release anotherRelease

migrate workareaview

migrate driverview

migrate fixView

migrate approvalView

migrate approverView

migrate PartsOutView where userLogin is not null

migrate changeview

migrate SizeView

migrate TestView

migrate DriverMemberView

migrate VerifyView

migrate CommonParts

migrate NoteView

migrate CoreqView
```

WorkareaView and DriverView are migrated twice because the branchpoint needs to be migrated following migration of the VersionView data.

Note: See the product readme.txt file for any additional migration commands and utilities not included in the command files provided in this document.

Executing migtc.lst

To migrate your database follow these steps:

1. Make sure your source and target servers are set up properly and the source server is running in maintenance mode.
2. Start the DB2 database instance on the target server.
3. From the target server, run TeamConnection report commands against the source server to make sure you are connected:

   ```
   teamc report -view users -family sourceFamilyName
   ```
4. Ensure that migtc.lst and migtc.exe (migtc in UNIX environments) are in the bin subdirectory of your TeamConnection version 3 installation path. Modify migtc.lst to suit your needs.
5. From a command prompt on your target server, type the following command to start the migration tool:

   ```
   migtc
   ```

   The migtc command displays a header and the migtc command prompt. The header contains information about your source and target servers. The migtc command prompt consists of the family name and > symbol.

   ```
   testfam>
   ```

   You can capture output to a file by starting the migration tool using the following command:

   ```
   migtc 1>mig.out 2>&1
   ```

   In a UNIX environment, you can send the output to a file and to your screen by issuing the following command:

   ```
   migtc 2>&1 | tee mig.out
   ```
6. At the migtc command prompt, enter the following command to execute the migration commands in migtc.lst:

   ```
   exec migtc.lst
   ```

   As you migrate the database, you may notice the following messages and behavior:

   - When users are migrated you will see one fhccomp record migrated. This is the root component for the family.
   - When VersionView is migrated, you may see a message that releaseName:1 and releaseName are already loaded in the target database. This is not an error.
   - When BuilderView is migrated, you will see a message for null builders indicating that there is no source to extract.
   - Migrating NoteView combines the actions from the history section into a single note. The number of notes migrated is likely to be less than the number of records for NoteView in the source database.
   - If you are migrating all version of parts, there may be more part extract actions than there are records in the report from the source database. This occurs for the following reasons:
     – There are additional part extracts for each part that has been changed in a workarea.
     – There are additional part extracts for each part that has been changed in a driver before the driver was refreshed from the release.

- If you are migrating all versions and your database contains parts with no contents, you will see five attempts to extract the part. The part will be migrated to the target after five attempts and the part type will be `empty`.

7. Backup your database between execution of each command file using the following command, or an equivalent:

   ```
   db2 backup database family_name to backup_directory
   ```

   Substitute your family name for *family_name* and a directory path for your backed up database for *backup_directory*. The DB2 backup utility will place a compressed version of the database in the backup directory path. Be sure to set file permissions for the backup directory such that the compressed backup file is accessible.

   Note: For a details on backing up DB2 databases, see the *DB2 Universal Database Administration Guide*.
8. After migration is complete, exit the migtc command interface.
9. Perform runstats and table reorganization to optimize database performance. Although general guidelines are provided in the *Administrator's Guide*, it is advisable to see the VisualAge TeamConnection readme.txt file for references to specific recommendations.
10. Start the migrated family server on the target server.
11. The defect serial number must be updated to determine the initial defect number in the migrated database; otherwise, TeamConnection will use the first unused number (starting from 1) when a defect or feature is opened.
    a. Create a defect to initialize the fhcsequence table.
    b. Update the fhcsequence table.

       An Administrator with the correct DB2 authority needs to connect to the family database and update the fhcsequence table.

       The command syntax is as follows:

       ```
       db2 connect to tc_family
       db2  select * from fhcsequence
       db2 update fhcsequence set lastSerial=lastDefectNumber where name='defect'
       ```

       Where `tc_family` is the name of the family and `lastDefectNumber` is the last number used.
12. Run queries like the following to verify that you have migrated all necessary data from the source database. Compare the output from these queries to the output from the queries you ran from the source server in "Preparing the source server for migration" on page 17.

    ```
    teamc report -view workareaView >workare2.vew


    teamc report -view releaseView >release2.vew


    teamc report -view compView >compon2.vew


    teamc report -view defectView >defect2.vew


    teamc report -view featureView >feature2.vew
    ```

In addition, for each release migrated, perform extracts from the source and target databases and validate that release contents and part bulk contents are identical.

For more information on the migration commands, see “Using the migration tools” on page 43.

Preparing to administer your new database

You cannot use your existing TeamConnection version 2 configurable field or user exit files in their present state with your version 3 database. The following sections provide guidelines for adding or updating entries from your existing files to your version 3 database.

Tables for authority, interest, configurable fields, and processes

Your config.ld, authorit.ld, interest.ld, relproc.ld, and comproc.ld files have been modified since the version 2 release, and the version 2 files are not valid for a version 3 database. Generate new .ld files from the database by redirecting the raw output of a report.

The following commands show how to create these files on your version 3 database:

```
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 -sel "configtype, name, dflt, value1,
value2, kind, driverid, driverseq, choiceorder, description,
concat('\"', concat(helptext,'\"')) " -fam $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.

The command will extract the configurable information from your migrated (target) database and update your version 3 .ld files accordingly.

You should only access these .ld files if you want to add, delete, or change any of the values. Refer to the *Administrator's Guide* for instructions.

If you have modified the .ld files, you will need to reload the tables according to the instructions in the *Administrator's Guide*.

User exit file

User exits from version 2 cannot be use with version 3. Use the userExit file shipped with TeamConnection version 3. Move the userExit file to the \config subdirectory of your TC_DBPATH directory.

Any user exits written for version 2 will have to be rewritten since the infrastructure from version 2 to version 3 has changed. Use the sample viewexit.cmd located in the samples subdirectory of the TeamConnection installation path to see the new order of user exit parameters. You need to concern yourself mainly with any parameters being searched and returned. the viewexit.cmd sample will help you determine the new order of parameters.

Configurable field definitions

You cannot use your version 2 .tbl files with a version 3 database. To preserve the version 2 configurable field information, you must migrate your version 2 .tbl files.

After creating a TeamConnection Version 3 family, a subdirectory cfgField is created in the family's home directory. This directory contains a .tbl file for each object that has default configurable fields. The format of this file (for Defect.tbl, for example) is as follows:

```
Active|DBColumnName|CMDAttribute|FieldLabel|Title|Create|Required|type|owner|origin
```

The steps for migrating TeamConnection version 2 .tbl files are as follows:

1. After creating a TeamConnection version 3 family, but before running migtc, copy the .tbl files from the TeamConnection version 2 cfgField directory to the TeamConnection version 3 cfgField directory. For example:
   ```
   copy /V2family_home/cfgField/Defect.tbl /V3family_home/cfgField/Defect.tbl
   ```
2. Edit the .tbl files to append `|yes|yes` to the end of each row. For example, if the old format is as follows:
   ```
   yes|symptom|symptom|Symptom||yes|yes|symptom
   ```
   the new format is as follows:
   ```
   yes|symptom|symptom|Symptom||yes|yes|symptom|yes|yes
   ```
3. Run fhcfupdv for each new tbl file. For example:
   ```
   fhcfupdv Defect.tbl DefectView
   ```
   Note: Refer to the *Administrator's Guide* for additional details on the fhcfupdv utility.

Mail exits

If you are migrating from TeamConnection version 2, you can use your version 2 mailexit file unless you are also migrating to another platform. If you are migrating to another platform, use the TeamConnection version 3 mailexit file for your target platform.

Initialization file

If you want to preserve queries you've written in your task list, you can use the teamcv3.ini (for Intel) or teamcv3x (for UNIX) file, as follows:

```
copy teamc20.ini teamcv3.ini
```

or

```
cp teamc20.ini teamcv3x.ini
```

Importing fine-grained data into your migrated database

During execution of migtc, the migration tool created a script file for each release for which the `migrate FineGrained` command was issued. The following is an example of such a script file for a UNIX platform:

```
# First, point to TC_FAMILY_CLIENT database for export
export TC_FAMILY=v2dbase@v2server@2222
teamc Release -export releaseName -file releaseNameFG.cdf -verbose
# Now, point to TC_FAMILY database for import
export TC_FAMILY=v3dbase
teamc workarea -create releaseNameWA -release releaseName -family v3dbase -verbose
# Now, import  the CDF file into the new workarea
teamc workarea -import releaseNameWA -file releaseNameFG.cdf -family v3dbase -verbose
# Integrate the workarea
teamc workarea -integrate releaseNameWA -release releaseName -family v3dbase -force -verbose
# All Finished
```

These script files perform the following commands:

1. Sets the TC_FAMILY environment variable to the original database.
2. Exports fine-grained data from the release specified into a file called *releaseName*FG.cdf.
3. Sets the TC_FAMILY environment variable to the migrated database.
4. Creates a workarea for the *releaseName*FG.cdf file.
5. Imports the .cdf file into the workarea.
6. Integrates the workarea.

To import fine-grained data into your migrated database, you need to do the following for each script file:

1. Edit each one to ensure that the source and target databases are properly specified in the TC_FAMILY environment variables; that release names, workarea names, and .cdf file names are appropriate; and that the family names specified in the -family attributes are correct.
2. If the target release uses any track processes, then you must modify the script by adding commands appropriate to your release process (such as commands for opening and approving defects, creating fix records and approval records, and so on). You must also include any required configurable field information in the script.
3. Execute the script file.

Only parts at the release tips will be migrated: parts in workareas will not be migrated.

Migrating from CMVC to TeamConnection version 3

Existing CMVC customers must migrate their data if they want to be able to use it on TeamConnection version 3. To migrate to TeamConnection version 3, you need two separate server machines:

- A CMVC 2.3.1.2 family server with your original database. This server must also have a superuser ID and a host list for the superuser so that the target server can access the source server.
- A separate TeamConnection family server with TeamConnection version 3 installed, to which you will migrate your database. This server must be installed and configured as described in the *Installation Guide*. No database should exist in

the TC_DBPATH location of the target server. This server must also contain a CMVC 2.3.1.2 client with a superuser ID for accessing the source database.

| For more information about migrating from CMVC, refer to the technical report
| ″Migrating CMVC 2.3.1 to VisualAge TeamConnection Enterprise Serve V3.″ You
| can obtain a copy of this technical report from Web address
| <http://www.software.ibm.com/ad/teamcon/library>.

General guidelines

The user ID of the person responsible for the migration must have superuser authority in the source system to guarantee access to all data for migration. This authority will be migrated from the source database to the target database.

It is recommended that the migration be done in stages with frequent backups of the database. You can use WHERE clauses to capture and migrate a limited or expanded set of data. With frequent backups, you can reset the database to the steps prior to this change and then repeat the migration with the modification.

Preparing the source server for migration

There are several steps you can take before you begin migrating to help the migration process execute more smoothly:

- Complete all levels and tracks before beginning migration to ensure accurate results.
- Tracks (in CMVC) in the integrate state are migrated to the fix state (In TeamConnection) because there is no way of knowing which files have been checked out in relation to the track when moving to TeamConnection version 3. Fix records in the active state are ignored during migration. Ensure that tracks (in CMVC) are in the complete state before beginning the migration. For any tracks not in the complete state, the owner will have to manually check out the files, replace them with the appropriate files containing the modifications, check the files into the workarea, and integrate the workarea.
- Files from CMVC in the checkout or locked state are not migrated to the target database because the migration tool does not know the workarea or defect that the locked file is associated with. Before migrating, check in or unlock all CMVC files.
- Level members in CMVC are migrated to a driver in the target database. Because only committed and complete levels are migrated to TeamConnection version 3, then level members for levels in the integrate state are not migrated. Ensure that all levels in the integrate state are committed or complete before migrating.
- Create a user ID with superuser access to your original database and a host list for the user ID. This ID will be used to access the DB2 database instance from the target server.
- Convert SCCS keywords in text files to be compatible with TeamConnection keyword support. TeamConnection provides several scripts that aid this process. These scripts can be found in the samples subdirectory.
  – FileExtract2.migcmvc changes the keywords and generates a list of text files for all releases to be migrated. This script is added to the `file -extract` step of the migration process as a user exit to the `file -extract` command to be executed during migcmvc with change history (full) migration.

  – keywordConversion.migcmvc is a script to be executed after performing a release extract to prepare for migrating files with keywords.

  See “Converting SCCS keywords” for complete instructions for converting keywords.

- You may want to run queries like the following and save the output to a file. These commands capture the data that will be migrated. You can experiment with WHERE clauses to limit the objects to be migrated.

```
cmvc report -view trackView >track1.vew

cmvc report -view releaseView >release1.vew

cmvc report -view compView >compon1.vew

cmvc report -view defectView >defect1.vew

cmvc report -view featureView >feature1.vew
```

- Restart the family server in maintenance mode (using the -m option) so that no updates can be made to the server while migration is in progress. Client users can view information, but not change it.

The following table shows how certain objects in a CMVC database are migrated into TeamConnection. These tables indicate which states can be migrated and how the states of objects might change from CMVC to TeamConnection.

Table 1. Migrating CMVC objects to TeamConnection

Object	**State in CMVC**	**State in TeamConnection**
Tracks	approve	approve
	cancel	cancel
	fix	fix
	integrate	fix
	commit	commit
	complete	complete
Fix	active	(not migrated)
Levels	working	not migrated
	integrate	not migrated
	commit	commit
	complete	complete

Defects in the working state prior to migration will be in the working state following migration. Any file changes that were checked into the database prior to migration but did not have completed fix records are not migrated. Following migration, the owner of the defect must create a workarea, check out the necessary files, make changes, check the changes in, and integrate the workarea.

Converting SCCS keywords

Keywords that have been inserted into files can be expanded during file extracts so that you can determine their version after the files have been delivered. This operation is accomplished by requesting the ″Expand keywords″ option during a file extract. Expanding keywords is essential in large systems where updates do not always replace all files in your product. CMVC and TeamConnection use different formats for keywords, so during a database migration, the CMVC keywords need to be converted to TeamConnection format. TeamConnection provides the following script that helps you convert CMVC keywords to TeamConnection format:

- keywordConversion.migcmvc prepares files with keywords for migration. You use this script only if you extract files before migrating.

- FileExtract2.migcmvc changes the keywords. This script is added to the `file -extract` step of the migration process as a user exit to the `file -extract` command to be executed during migcmvc with change history (full) migration.

These scripts use the following tools:
- sed to make changes to the files
- grep to make sure files need to be changed
- CMVC to find all files that have a file type other than "binary" (e.g. text, long, longSp, special) and have not been deleted

Do the following before you run these scripts:
- Set the CMVC_FAMILY environment variable to your CMVC family name
- Set the CMVC_TOP environment variable to the directory path to which you are extracting

The conversion will change only those CMVC keywords that have TeamConnection equivalents. Keywords that do not have TeamConnection equivalents will be left as is in the files.

There are two ways to use these scripts:
- If you extract levels or releases before you begin the migration process, use these scripts as follows:
  1. Extract the files to directory structures using the level or release extract command. Do not use the file extract command.
  2. Run keywordConversion.migcmvc to prepare files before the migration. Start this script as follows:
     ```
     keywordConversion.migcmvc CMVC_FAMILY CMVC_TOP
     ```
     Only .c and .ksh text files will be converted. You will have to modify the script to convert other file types. Binary files will not be converted.
  3. Run migcmvc.
- If you extract files as part of a change history migration, use these scripts as follows. When using this scenario to convert keywords, the you must start the CMVC family with at least 2 daemons. This is due to the fact that the user exit will issue report commands that otherwise will deadlock the family.
  1. Add user exit FileExtract2.migcmvc to `file -extract` user exit ID 2. (See the *Administrator's Guide* for instructions on setting up a user exit.
  2. Run migcmvc.

The file /tmp/checkreps.releaseName is a structured file containing the release and path name for each text file in all releases to be migrated. The script appends to this file for each release. Be sure to delete /tmp/checkreps.releaseName each time you want to generate a new list of files in it.

Keywords are converted to TeamConnection format as follows:
- The concept of current date/extract date is not relevant to TeamConnection. The last updated date is used during extract. As a result, the extract dates are commented out and the update dates are used:
  ```
  %D% -> /* Use $ChkD, get date NA */

  %H% -> /* Use $ChkD, get date NA */

  %T% -> /* Use $ChkD, get date NA */
  ```

```
%E% -> $ChkD;

%G% -> $ChkD;

%U% -> $ChkD;
```

- Some of the keywords do not have a direct translations in SCCS (they were derived from PVCS):
  ```
  $Own;
  ```
- All references to internal information exposed by SCCS are not converted to TeamConnection external values. As a result several values convert to a single FileName:
  ```
  %F% -> $FN;

  %M% -> $FN;

  %P% -> $FN;
  ```
- Versioning information has been consolidated. As a result, the following keyword changes occur:
  ```
  %I% -> $Ver;

  %W% -> $Ver;

  %R% -> /* Use $Ver */

  %B% -> /* Use $Ver */

  %S% -> /* Use $Ver */
  ```
- Ideally, the sed script should locate the last keyword and insert the end-replace keyword $EKW after it. However, that is a personal preference and left for users to modify as they see fit.

If the conversion scripts detect keywords not supported, they will generate comments stating this fact. The following is a brief description of some of the SCCS keywords. If a keyword does not have a TeamConnection equivalent, it will remain as is in the file.

$Own; The owner of the component that manages the part.

$KW=@(#);
Begin keyword expansion after this keyword.

$EKW;
End keyword expansion until the next $KW=@(#); keyword. %Z% SCCS keyword converts to $KW=@(#); in TeamConnection.

%A% Not applicable in TeamConnection

%Y% Not applicable.

%C% Not available in TeamConnection

Migrating renamed parts

In order for renamed parts to be migrated correctly, the changeview in the CMVC family must be dropped and recreated. To do this, issue the following CMVC commands in the order shown:

```
 stopCMVC familyname

drop view changeview
```

```
create view ChangeView
        (pathId, trackId, fileId, versionId, levelId, type,
         defectPrefix, defectName, defectReference, defectAbstract, defectType,
         releaseName, pathName, versionSID, userId, levelName, newPathName,
         userName, fileType) as
 select
        c.pathId, c.trackId, c.fileId, c.versionId, c.levelId, c.type,
        d.prefix, d.name, d.reference, d.abstract, d.type,
        n.name, p.name, v.SID, c.userId, b.name, f.basename, u.login, f.type
 from
        Changes c, Tracks t, Defects d, Releases n, Path p, Versions v, Levels b,
        Files f, Users u
 where
        c.trackId = t.id and
        t.defectId = d.id and
        t.releaseId = n.id and
        c.versionId = v.id and
        c.pathId = p.id and
        c.levelId = b.id and
        f.id =  c.fileid and
        u.id =  c.userid and
        p.id = f.pathid

commit

cmvcd familyname number of daemons
```

Setting up the target server

You run the migration tools from the target server, which communicates with your source server to migrate the database. Although the version 3 installation program cannot detect if a CMVC server is installed, you need to make sure that you install the version 3 server on a different machine from your CMVC server.

Follow these guidelines to ensure that your target server is set up properly for migration.

Install TeamConnection version 3 on your target server according to the instructions in the *Installation Guide*:

- Install TeamConnection version 3 and perform the installation verification procedure to create the testfam database. Make sure you complete the installation process, including updating your TCP/IP hosts and services files with IP addresses for your target family.
- Install the CMVC 2.3.1.2 client on the target server. Make sure your config.sys or .profile contain the following settings:
  - Ensure that CMVC is in the PATH statement before TeamConnection, for example,

    `SET PATH=C:\CMVC\EXE;E:\teamcpath;`
  - Make sure CMVC.CAT is in the NLSPATH statement before TEAMCV30.CAT, for example,

    `SET NLSPATH=c:\cmvc23\exe\nls\%N;c:\teamcpath\nls\msg\enu\%N;`

  Several executable programs that are common to both the CMVC and target client are used in the migration. The PATH environment must find the appropriate version of these programs:

File.exe
: To migrate files

Report.exe
: To migrate all views

Defect.exe
: To migrate notes

Feature.exe
: To migrate notes

The CMVC file path should precede the TeamConnection version 3 file path.

- Determine the directory or path where the target database will be located. This directory path will be your TC_DBPATH. Do not create a family database in this location. Instead just create a subdirectory from your TeamConnection version 3 installation directory to contain the migrated database. You could, for example, use the directory path `d:\teamc\v3dbase`.
- Set the following environment variables in config.sys or .profile on the target server:

SET TC_FAMILY=*migratedDatabaseName@hostname@port*
: Set this variable to the name of your migrated family database on the target server.

SET CMVC_FAMILY=*originalDatabaseName@hostname@port*
: Set this variable to the family name, host name, and port ID of your original database on the source server.

SET TC_USER=*superuserID*
: Set this variable to a user ID with superuser authority.

SET CMVC_USER=*superuserID*
: Set this variable to a user ID with superuser authority.

SET CMVC_BECOME= *superuserID*
: Set this variable to a user ID with superuser authority.

SET TC_BECOME= *superuserID*
: Set this variable to a user ID with superuser authority.

SET TC_DBPATH=*migratedDatabasePath*
: Set this variable to the directory path you created for your migrated database on the target server.

SET CMVC_TYPE=
: Set this variable only if you are migrating from a CMVC95 server. If you are migrating from a CMVC 2.3.1.2 server, this environment variable is not necessary.

SET CMVC_KEYS=yes
: Set this variable to yes to convert SCCS keywords; omit it to prevent keywords from being converted.

SET CMVC_ALLCOMMON=yes
: CMVC_ALLCOMMON is used in conjunction with ″migrate commonparts.″ Set CMVC_ALLCOMMON=yes to migrate all common parts in a CMVC family. If CMVC_ALLCOMMON is not set only common files in binding releases will be migrated. Migrating only the common files in binding releases will greatly reduce migration times.

The following are examples of these environment variables, using cmvcdbase as the original family name and v3dbase as the migrated family name:

```
SET TC_FAMILY=v3dbase@v3server@1111

SET CMVC_FAMILY=cmvcdbase@v1server@9999

SET CMVC_USER=migrator

SET CMVC_BECOME=migrator

SET TC_USER=migrator

SET TC_BECOME=migrator

SET TC_DBPATH=d:\teamc\v3dbase
```

- Reboot the machine.
- Set up a directory structure from your new subdirectory for your migrated database similar to the testfam sample structure described in the installation verification procedure. The following directory structure and files must exist off the TC_DBPATH. For example, if TC_DBPATH=d:\teamc\v3dbase, then the directory structure should appear as follows. You can copy these files from the \testfam subdirectory created when you ran the installation verification procedure for your version 3 installation.

```
teamc
     v3dbase
     cfgfield
         Defect.fmt
         Defect.tbl
         Feature.fmt
         Feature.tbl
         Part.fmt
         Part.tbl
         PartView.fmt
         Release.fmt
         RelView.tbl
         User.fmt
         User.tbl
         WAView.fmt
         Workarea.tbl
    config
         userExit
         security (new file; did not exist in CMVC)
    queue
    tctmp
```

Note: The TeamConnection administrator is responsible for migrating the .ld files after migration. See the *Administrator's Guide*.

- When migrating a single, committed version of each file from CMVC, you must first do a `release -extract`, `file -extract`, or `part -extract`, to get the files you want to migrate out of the source database. You must then move the files to the TeamConnection version 3 server machine to migrate them into the target database. You can move the files by using a copy command, ftp, or any other available method.
- Start the DB2 database instance.

Performing the migration

Once you have prepared your source database and set up your source and target servers, you are ready to perform the migration. The TeamConnection migration utility, migcmvc, is a command-line utility that you run from your target server. It uses the information you provided in "Setting up the target server" on page 35 to locate the source database.

TeamConnection provides a file called migcmvc.lst containing sample migration commands. This file is located in the /bin subdirectory of your TeamConnection version 3 installation path. It may be helpful for you to start with this command file, modify it to suit your needs, and then execute the migration using this command file.

Migrating the current version of files from CMVC

The following are sample contents of migcmvc.lst as shipped. The commands in this file are written for the following migration functions:

- Only the current versions of parts (files) are migrated.
- All objects are migrated, for example, all users, all parts, all defects, and so on.

If you want to select specific objects to be migrated, you will need to modify this command file by adding `where` clauses to the commands. See "Using the migration tools" on page 43 for information on specific migration commands.

```
set batchsize 100

set decache   1

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'

migrate   ReleaseView

migrate   EnvView

migrate   AccessView

migrate   NotifyView

migrate   DefectView

migrate   FeatureView

migrate   LevelView where 1=1 order by commitDate

migrate   TrackView

migrate   FixView

migrate   ApproverView

migrate   ApprovalView
migrate   LevelMemberView

set  top  x:\location of files in release if migration only the current version

migrate fileview where releasename='9604' and dropdate is null

migrate   SizeView

migrate   NoteView


migrate CommonParts
```

The migration tool supports migration of committed versions of CMVC files directly into the release. The bulk contents are obtained from a local drive. To migrate CMVC files, you need to do the following:

- If you have a very large database, it may be wise to divide this command file into several smaller files, such as migcmvc1.lst, migcmvc2.lst, and so on. Chunking this command file will enable you to backup your database in intervals during the migration.
- Set the CMVC_TOP environment variable or use the migrate command to set the top directory to the directory containing the CMVC files.
  1. Create the source code tree structure on the target server or on a LAN drive that the target server has read/write access to.
  2. Change the `set top` variable in migcmvc.lst to point to the directory structure.
- Use the set batchSize command to specify the number of files to process in one transaction.
- Use the migrate FileView command to specify the release to be migrated from the CMVC database to TeamConnection. Files are read from the local drive. They are not extracted from CMVC when migrating a single version. Always include **dropdate is null** with your migrate command to ensure that it does not attempt to select files that have been deleted. You do not need to migrate or create a workarea before migrating CMVC files. Modify this command to include the release names defined in your original database. If you have more than one release, create additional `migrate fileview` commands.

To use this command file for your migration, you need to make the following minimal modifications. For a complete list of migration commands, see “Using the migration tools” on page 43

Migrating previous versions

It is not recommended that you migrate all of your past versions, as this can be very time consuming, and you will most likely not use all of this information. Instead, keep your CMVC database as an archive and retrieve previous versions of parts when you need them.

To migrate previous versions, the `set top` variable must remain null, as shown in this example.

```
migrate   LevelView

migrate   TrackView

migrate   ChangeView
```

```
set top

migrate   FileView -release releaseName
```

Executing migcmvc.lst

To migrate your database follow these steps:

1. Make sure your source and target servers are set up properly, the source server is running in maintenance mode, and the client on the target server is running.
2. From the CMVC client installed in the source server, run CMVC report commands to make sure you are connected:
   ```
   report -view users -family sourceFamilyName

   report -view hostView -family sourceFamilyName
   ```
3. Ensure migcmvc.lst is in the \bin subdirectory of your TeamConnection version 3 installation path. Modify migcmvc.lst to suit your needs.
4. From a command prompt in the directory where you copied migcmvc.lst, type the following command to start the migration tool:
   ```
   migcmvc
   ```

   The migcmvc command displays a header and the migcmvc command prompt. The header contains information about your source and target servers. The migcmvc prompt consists of the family name and > symbol.
   ```
   testfam>
   ```

   You can capture output to a file by starting the migration tool using the following command:
   ```
   migcmvc 1>mig.out 2>&1
   ```
5. At the migcmvc command prompt, enter the following command to execute the migration commands in migcmvc.lst:
   ```
   exec migcmvc.lst
   ```

   As you migrate the database, you may notice the following messages and behavior:
   - When users are migrated you will see one fhccomp record migrated. This is the root component for the family.
   - Migrating NoteView combines the actions from the history section into a single note. The number of notes migrated is likely to be less than the number of records for NoteView in the source database.
6. Backup your database between execution of each command file using the following command, or an equivalent:
   ```
   db2 backup database family_name to backup_directory
   ```

   Substitute your family name for *family_name* and a directory path for your backed up database for *backup_directory*. The DB2 backup utility will place a compressed version of the database in the backup directory path. Be sure to set file permissions for the backup directory such that the compressed backup file is accessible.

Note: For a details on backing up DB2 databases, see the *DB2 Universal Database Administration Guide*.

7. After migration is complete, exit the migcmvc command interface.
8. Perform runstats and table reorganization to optimize database performance. Although general guidelines are provided in the *Administrator's Guide*, it is advisable to see the VisualAge TeamConnection readme.txt file for references to specific recommendations.
9. If you have partitioned this command file into several smaller files, backup your database between execution of each command file using the DB2 utilities provided for backup:
10. Start the migrated family server on the target server.

    Note: You are using the shipped version of configurable field files. If you have configurable fields, they may not display until the administrator has modified these files.
11. Grant users access to the data in the migrated database. In version 3, users cannot access data as they did in CMVC if they are not:
    - A superuser
    - A component owner
    - A part owner

    You will need to grant users access to each component with the appropriate authority using a command like the following:

    ```
    teamc access -create -login userName -authority authorityGroup
    -component componentName
    ```
12. Run queries like the following to verify that the correct number of objects were migrated. Compare the output from these queries to the output from the queries you ran from the source server in "Preparing the source server for migration" on page 17.

    ```
    teamc report -view workareaView >workare2.vew

    teamc report -view releaseView >release2.vew

    teamc report -view compView >compon2.vew

    teamc report -view defectView >defect2.vew

    teamc report -view featureView >feature2.vew
    ```

    In addition, for each release migrated, perform extracts from the source and target databases and validate that release contents and part bulk contents are identical.

For more information on the migration commands, see "Using the migration tools" on page 43.

Preparing to administer your new database

After migrating your database, you need to set up a directory structure for your family similar to the testfam sample structure described in the installation verification procedure. The following directory structure and files must exist off the TC_DBPATH. Type **SET TC_DBPATH** to get the name of the family directory. For example, if TC_DBPATH=d:\teamc\testfam, then the directory structure should appear as follows:

```
teamc
     testfam
         (this directory contains .ld files, which the
          administrator is responsible for migrating)
    cfgfield
         Defect.fmt
         Defect.tbl
         Feature.fmt
         Feature.tbl
         Part.fmt
         Part.tbl
         PartView.fmt
         Release.fmt
         RelView.tbl
         User.fmt
         User.tbl
         WAView.fmt
         Workarea.tbl
    config
         userExit
         security (new file; did not exist in CMVC)
    queue
    tctmp
```

In most cases you cannot use your existing CMVC files with your version 3 database. The following sections provide guidelines for migrating your existing files to your version 3 database or using the version 3 files.

Tables for authority, interest, configurable fields, and processes

You cannot use the configurable files as they existed in CMVC. 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 -sel "configtype, name, dflt, value1,
value2, kind, driverid, driverseq, choiceorder, description,
concat('\"', concat(helptext,'\"')) " -fam $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.

The command will extract the configurable information from your migrated (target) database and update your version 3 .ld files accordingly.

You should only access these .ld files if you want to add, delete, or change any of the values. Refer to the *Administrator's Guide* for instructions.

If you have modified the .ld files, you will need to reload the tables according to the instructions in the *Administrator's Guide*.

User exit file

If you are converting from CMVC, use the TeamConnection version 3 config\userExit file and add to it any modifications that you made for your CMVC installation. See the *Administrator's Guide* for instructions on working with user exits.

Configurable field definitions

The CMVC configurable field files (defectConfigFormat, featureConfigFormat, fileConfigFormat) are replaced by a collection of .fmt files (Defect.fmt, Feature.fmt, Part.fmt, Release.fmt, User.fmt, and Wrokarea.fmt). An additional configurable field file, called PartView.fmt, is available in TeamConnection version 3. If you are migrating from CMVC, use the TeamConnection version 3 .fmt files and add to them any modifications that you made for your CMVC installation. See the *Administrator's Guide* for instructions on working with these configurable field files.

The configurable field tables from CMVC (ConfigTable files) can be used with TeamConnection version 3. Refer to the *CMVC Administrator's Guide* for the name and location of the files. The corresponding files should then be copied and renamed then placed in the cfgfield directory: defect.tbl, feature.tbl, user.tbl, part.tbl, release.tbl, and workarea.tbl

Mail exits

Use the TeamConnection version 3 mailexit file for your target platform.

Using the migration tools

Note: It is not recommended that you make changes to your database by issuing INSERT, UPDATE, or DELETE statements or by changing or deleting database tables or the columns defined in TeamConnection database tables. Changing your database in these ways, through the DB2 administrator tools, the DB2 command line processor, the TeamConnection migration tools, or the tcupdb tool can corrupt your TeamConnection database. Any such changes are made at your own risk. Please contact your IBM representative for information on the terms of IBM customer support.

The CMVC and TeamConnection migration tools provide a command line environment from which you can issue the migration commands. You start the migration tool using the version 3 client, which communicates with the source server to migrate data to the version 3 server. To start the migration tools, type one of the following from your operating system command line:

migcmvc
To start the migration tool for CMVC-to-TeamConnection

migtc To start the migration tool for TeamConnection-to-TeamConnection

The command presents a prompt consisting of the family name and > symbol. Type the migration commands at this prompt. To see a list of the migration commands, type **?** at this prompt, as in the following example:

```
testfam> ?


     Valid commands are:
```

```
    migrate view[,view...] [[where] where-clause]

    migrate bcompview  where name='top-component or root'

    set argument [value]

    exec file

    !system-command

    quit

  For more information, enter the command followed by a "?"
```

To see additional help information for a single command, type the command followed by ?.

Migration tool variables

Certain environment variables for the migration tool can be set to change how the tool operates. These variables use default values, but you can change them to suit your needs. The following is a list of the default values. For information on these variables and instructions for setting them, see "Set command" on page 49

```
batchSize set to 1000.

maxErrors set to 0.

decache set to 0.

top set to "D:\".

reportStyle set to "stanza".

loadMessage set to "none".
```

Sample migration files

Sample files, called migtc.lst and migcmvc.lst, are provided with the install package. You can edit the commands in these files and use them to perform a migration. Use the exec command to execute the commands in these files:

```
exec migtc.lst
```

```
exec migcmvc.lst
```

To add comments to migration command files, use a # symbol. Any line preceded by a # symbol is ignored by the migration tool.

Migrate command

The migration command executes a report command against the source client and parses this data while creating the objects in the TeamConnection target database. You can choose to migrate all or part of the source data. You select the data to be migrated by specifying `where` clauses on the migrate command to generate reports on the data to be migrated. Use caution when writing these where clauses, because there is potential for error if all objects, such as components or releases, are not migrated but files relating to these objects are migrated. It is your responsibility to generate the correct set of data for migration.

To migrate data from CMVC or TeamConnection version 2 to TeamConnection version 3, type the following command at the migration command prompt:

```
migrate view[,view...] [[where] whereClause]
```

Where:
- *view* is one of the views shown in the following list.
- *whereClause* is a whereClause constructed using column (or field) names as shown in the *Commands Reference* Refer to the *Commands Reference* for more information on writing queries on TeamConnection views.

The following are some examples of the migrate command:
- The following command migrates all users:
  ```
  migrate users
  ```
- The following command migrates all components whose dropDate is not null.
  ```
  migrate compview where dropDate is not null
  ```
- The following command migrates all defects in open and working state:
  ```
  migrate defectview where state in ('open', 'working')
  ```

You can migrate the following views using the migrate command:

AccessView
: Access list data. Used for migration from TeamConnection version 2 and CMVC.

ApprovalView
: Approval records. Used for migration from TeamConnection version 2 and CMVC.

ApproverView
: Approver records. Used for migration from TeamConnection version 2 and CMVC.

Authority
: Authority list. Used for migration from TeamConnection version 2 and CMVC. During the process of migrating Authority, the actions defined in the authority table are changed to the TeamConnection action name as follows:
  - Track actions are changed to workarea actions
  - Level actions are changed to driver actions
  - File actions are changed to part actions

  New actions that exist in TeamConnection but not in the source database are added and paired with an appropriate group. After migration, you can issue a TeamConnection report command to show the actions and authority groups that exist. These can be compared to a similar report from your source database.

Following migration, if the authority table needs modifications, you can produce an authorit.ld file as described in the *Administrator's Guide*.

bCompView
Component members which make up the component hierarchy. Used for migration from TeamConnection version 2 and CMVC. When migrating components (CompView) followed by component members (bCompView), it is required that the top level component be specified on the migrate command. For example, if the default top level component exists in the source database and is named root then, you need to migrate the component members using the following command:

```
migrate bcompview  where name='root'
```

To see which component members will be migrated for the top-level component, issue the following report command:

```
report -view bcompview -where "name='yourRootComponent'" -raw
```

where *yourRootComponent* is the name of the root component on the source database. If you are not sure of the name of the top level component, the following query on the source client will provide it:

```
report -view compview -where "id not in (select


childid from compmemberview where parentid<>childid) and id in (select


parentid from compmemberview where parentid<>childid)" -raw
```

If the name of the top component is not root and root does not exist as a component, then you should modify the component name to make it root.

bPartView
Migrating build trees. Used for migration from TeamConnection version 2.

BuilderView
Builders are migrated and the corresponding file is extracted and created in the target database. Used for migration from TeamConnection version 2.

Cfgcomproc
Configurable component processes. Used for migration from TeamConnection version 2 and CMVC.

Cfgrelproc
Configurable release processes. Used for migration from TeamConnection version 2 and CMVC. When migrating configurable processes using Cfgrelproc, the name of the process will remain exactly as it was in the source database. For example, track_level process will be named track_level, but the process now includes ″driver″ instead of ″level.″ If you wish to change a process name, it is recommended that you make a duplicate of the current process and name the new one track_driver. To do this, first generate a relproc.ld file as described in the *Administrator's Guide*. In this file, change track_level to track_driver and then execute the following command to reload the configurable release processes table:

```
fhclproc relproc.ld databaseName r
```

where *databaseName* is the path name of the target database. See the *Administrator's Guide* for more information.

ChangeView
All versions of files. Used for migration from CMVC.

CommonParts
Common parts. Used for migration from TeamConnection version 2 and CMVC.

CompView
Components. Used for migration from TeamConnection version 2 and CMVC. See further instructions on bCompView.

Config
Configurable field values. Used for migration from TeamConnection version 2 and CMVC.

CoreqView
Corequisites. Used for migration from TeamConnection version 2.

DefectView
Defects. Used for migration from TeamConnection version 2 and CMVC.

DriverMemberView
Driver members. Used for migration from TeamConnection version 2.

DriverView
Drivers. Used for migration from TeamConnection version 2.

EnvView
Environments. Used for migration from TeamConnection version 2 and CMVC.

FeatureView
Features. Used for migration from TeamConnection version 2 and CMVC.

FileView
Files. Used for migration from CMVC.

FineGrained -release *releaseName*
Fine-grained data. Used for migration from TeamConnection version 2. Generates a script file called *releaseName*FG.ksh (UNIX) or *releaseName*FG.cmd (Intel) for the release specified. See "Importing fine-grained data into your migrated database" on page 30 for instructions on editing and executing these script files to migrate fine-grained data.

FixView
Fix records. Used for migration from TeamConnection version 2 and CMVC.

HostView
Host lists. Used for migration from TeamConnection version 2 and CMVC.

Interest
Interest table. Used for migration from TeamConnection version 2 and CMVC. During the process of migrating the Interest, the actions defined in the interest table are changed to TeamConnection action names as follows:

- Track actions are changed to workarea actions
- Level actions are changed to driver actions
- File actions are changed to part actions

New actions that exist in TeamConnection but not in the source database are added and paired with an appropriate group. After migration, you can issue a TeamConnection report to show the actions that exist. These can be compared to a similar report from your source database.

Following migration, if the interest table needs modifications, you can produce an interest.ld file as described in the *Administrator's Guide*.

LevelMemberView
Level members. Used for migration from CMVC.

LevelView
Levels. Used for migration from CMVC.

NoteView
Used for migration from TeamConnection version 2 and CMVC. The migrate NoteView command iterates over the collection of defects and features that have been migrated and issues a **defect -long -view defectName** or **feature -long -view FeatureName** command against the source database. It then searches for the word ″history:″ and takes all the following lines of data and creates a single note in the TeamConnection database. The user ID for the note is the originator of the defect or feature and the addDate of the defect or feature.

If you do not want to migrate all the notes, then migrate defects and features in stages. For example, if you want only notes for defects and features in the ′open′ and ′working′ state, then migrate defects and features ″where state in (′open′,′working′)″ and then migrate NoteView. Then proceed to migrated defects and features ″where state not in (′open′, ′working′)″.

NotifyView
Notification records. Used for migration from TeamConnection version 2 and CMVC.

ParserView
Parsers. Used for migration from TeamConnection version 2. When migrate of ParserView is complete, the parser files must be moved manually from the source server to the new TeamConnection server and located in the appropriate directory.

PartFullView
All versions of parts. Used for migration from TeamConnection version 2. Specify the release whose parts you want to migrate by including a `-release` attribute with the command as follows:

```
migrate   PartFullView -release yourReleaseName
```

PartView
Parts. Used for migration from TeamConnection version 2. Issue one `migrate PartView` command for each release in your original database. Specify the release whose parts you want to migrate by including a `-release` attribute with the command as follows:

```
migrate   PartView -release yourReleaseName
```

If you have fine-grained parts defined in your database, then you need to exclude them from the migration by specifying this command as follows:

```
migrate   PartView -release yourReleaseName where partType='TCPart'
```

See "Preparing the source server for migration" on page 17 for more information on preparing a database containing fine-grained data for migration.

ReleaseView
Releases. Used for migration from TeamConnection version 2 and CMVC.

SizeView
Sizing records. Used for migration from TeamConnection version 2 and CMVC.

TestView
Test records. Used for migration from TeamConnection version 2 and CMVC.

TrackView
Tracks from CMVC migrate to workareas. Used for migration from CMVC.

Users Used for migration from TeamConnection version 2 and CMVC. Do not use a `where` clause when migrating users. The **migrate users** command is a required step because it migrates the superuser authority and creates the special user named InheritedAccess. After you issue this command a component called root is created after all users are migrated. The root component is the top-level component and is used as the parent of the first component. The top level component on your target database must be named root, the default name that is created when the database is first initialized. This should not be modified.

Since users are the owners, originators, and creators of most objects in the database, they must also exist in the target database in order for the related objects to be migrated. For this reason it is recommended that all users be migrated and that you avoid using the where clause.

VerifyView
Verification records. Used for migration from TeamConnection version 2 and CMVC.

WorkAreaView
workareas. Used for migration from TeamConnection version 2.

Set command

To set options for the migrate and report commands, use the set command as follows:

```
set option value
```

Where:

- *option* is one of the options listed below.
- *value* is a valid value for the option.

You need to issue one set command for each option that you want to change. The following list shows the options you can set and the possible values for each. To display current values for each option, type **set ?**.

batchSize
This option specifies the number of objects to migrate in one transaction. The default is 1000.

maxErrors
The number of errors found before a migrate view is halted. For example, if migrating defects and the originator's user login is not found, then this counts as 1 error. The default is 0.

reportStyle
Specifies how reports generated by the report command are to be formatted. The default is stanza.

- **terse**
- **stanza**

loadMessage
Specifies how messages generated during a migration are to be formatted. The default is none.

- **terse**
- **stanza**
- **none**

deadlockRetry
Specifies how many times to retry a transaction if a deadlock occurs. This variable can be set to any value.

top Specifies the location of the source code, if needed.

Exec command

Use the exec command to issue migration commands from a file.

```
exec filepath
```

Where:

filepath is the full path name of the command file to execute. If you do not specify a full path name, the migration tool looks for the file in the current directory.

The following example executes migration commands defined in a file called migtc.lst.

```
exec migtc.lst
```

The following example executes migration commands defined in a file called d:\teamc\bin\migtc.lst.

```
exec d:\teamc\bin\migtc.lst
```

The file migtc.lst is a sample of the migration views that can be used to migrate your database.

! command

Use the ! command to issue operating system commands from the migration command prompt.

```
!systemCommand
```

Where: *systemCommand* is an operating system command.

The following example issues a dir command from the migration command prompt:

```
!dir
```

Quit command

Use the quit command to exit the migration command prompt and return to the operating system command prompt.

```
quit
```

# command

Use the # command to add a comment to a migration command file.

Services and Support

VisualAge TeamConnection Services and Support

Services

IBM consultants are available to help you, from planning to production and everything in between. For information about these services, please visit the following web site:

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

If you are interested in VisualAge TeamConnection Services, contact IBM Software Development Services via e-mail at:

`websphere_consulting@us.ibm.com`

Support

If you have a question or problem regarding VisualAge TeamConnection, you can find support information and our telephone numbers at the following web site:

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

Newsgroup

You can access VisualAge TeamConnection technical information, exchange messages with other VisualAge TeamConnection users, and receive information regarding the availability of FixPaks by visiting our newsgroup at:

`news://news.software.ibm.com/ibm.software.teamcon`

Bibliography

IBM VisualAge TeamConnection Enterprise Server library

The following is a list of the TeamConnection publications. For a list of other publications about TeamConnection, including white papers, technical reports, a product fact sheet, and the product announcement letter, refer to the IBM VisualAge TeamConnection Enterprise Server Library home page. To access this home page, select **Library** from the IBM VisualAge TeamConnection Enterprise Server home page at Web address <http://www.software.ibm.com/ad/teamcon>.

- **License Information:**

  Contains license, service, and warranty information.
- **Verifying Installation of TeamConnection:**

  Explains how to verify that TeamConnection has been installed correctly. Guides you through the process of creating an initial test family.
- **Administrator's Guide:**

  Provides instructions for configuring the TeamConnection family server and administering a TeamConnection family.
- **User's Guide:**

  A comprehensive guide for TeamConnection administrators and client users that helps them install and use TeamConnection.
- **Commands Reference:**

  Describes the TeamConnection commands, their syntax, and the authority required to issue each command. This book also provides examples of how to use the various commands.

TeamConnection technical reports

The following is a list of technical reports available for TeamConnection. Refer to the IBM VisualAge TeamConnection Enterprise Server Library home page for the most up-to-date list of technical reports. To access this home page, select **Library** from the IBM VisualAge TeamConnection Enterprise Server home page at Web address <http://www.software.ibm.com/ad/teamcon>.

29.2147	SCLM Guide to TeamConnection Terminology
29.2196	Using REXX Command Files with TeamConnection MVS Build Scripts
29.2231	TeamConnection Interoperability with MVS and SCLM
29.2235	Using REXX Command Files with TeamConnection MVS Build Scripts for PL/I Programs
29.2266	TeamConnection frequently asked questions: National Language Support (NLS) and Double-Byte Character Sets (DBCS)
29.2307	Data Driven TeamConnection User Exits
29.2333	Evolution of a New TeamConnection Family, Common Dos and Don'ts
29.2357	Evolution of a New VisualAge TeamConnection Family: Taking Advantage of Automation
29.3076	Configuration and Administration of DB2 Universal Database V5 by Users of VisualAge TeamConnection Enterprise Server V3
29.3088	Moving a VisualAge TeamConnection Version 3 Family
29.3090	Evolution of a VisualAge TeamConnection family: Using the Web and Shadowing to Build and to Distribute

29.3094	VisualAge TeamConnection 3: How to Do Routine Operating System Tasks
29.3096	Comparison Between CMVC 2.3.1 and VisualAge TeamConnection Enterprise Server 3
29.3098	VisualAge TeamConnection Version 3: Simple Build Function in UNIX
29.3099	VisualAge TeamConnection V3 Frequently Asked Questions: GUI and Line Command Clients for UNIX, OS/2, and Windows 32-bit
29.3113	Migrating CMVC 2.3.1 to VisualAge TeamConnection V3

DB2

The following publications are part of the IBM DB2 Universal Database library of documents for DB2 administration. DB2 publications are available in HTML format from the DB2 Product and Service Technical Library at the following Web address:

`<http://www.software.ibm.com/data/db2/library/>`

- *Administration Getting Started* (S10J-8154–00)

  An introductory guide to basic administration tasks and the DB2 administration tools.
- *SQL Getting Started* (S10J-8156–00)

  Discusses basic concepts of DB2 SQL.
- *Administration Guide* (S10J-8157–00)

  A complete guide to administration tasks and the DB2 administration tools.
- *SQL Reference* (S10J-8165–00)

  A reference to DB2 SQL for programmers and database administrators.
- *Troubleshooting Guide* (S10J-8169–00)

  A guide to identifying and solving problems with DB2 servers and clients and to using the DB2 diagnostic tools.
- *Messages Reference* (S10J-8168–00)

  Provides detailed information about DB2 messages.
- *Command Reference* (S10J-8166–00)

  Provides information about DB2 system commands and the command line processor.
- *Replication Guide* (S10J-0999–00)

  Describes how to plan, configure, administer, and operate IBM replication tools available with DB2.
- *System Monitor Guide and Reference* (S10J-8164–00)

  Describes how to monitor DB2 database activity and analyze system performance.
- *Glossary*

  A comprehensive glossary of DB2 terms.

Related publications

- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - *TCP/IP 2.0 for OS/2: Installation and Administration* (SC31-6075)
  - *TCP/IP for MVS Planning and Customization* (SC31-6085)
- MVS
  - *MVS/XA JCL User's Guide* (GC28-1351)
  - *MVS/XA JCL Reference* (GC28-1352)
  - *MVS/ESA JCL User's Guide* (GC28-1830)

- *MVS/ESA JCL Reference* (GC28-1829)

- NLS and DBCS
  - *AIX 4, General Programming Concepts: Writing and Debugging Programs.* (SC23-2533-02). See chapter 16 ″National Language Support″ for an updated contents of the AIX 3 material (see below).
  - *AIX 4, System Management Guide: Operating System and Devices* (SC23-2525-03). See chapter 10, ″National Language Support″ for system tasks.
  - *AIX Version 3.2 for RISC System/6000, National Language Support* (GG24-3850).
  - *Internationalization of AIX Software, A Programmer's Guide* (SC23-2431).
  - *National Language Design Guide Volume 1* (SE09-8001-02). This manual contains very good information on how to enable an application for NLS.
  - *National Language Design Guide Volume 2* (SE09-8002-02). This manual provides information on the IBM language codes (consult the ″Language codes″ chapter).

Readers' Comments — We'd Like to Hear from You

IBM VisualAge TeamConnection Enterprise Server
Verifying Installation of TeamConnection

Publication No. GC34-4742-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name	Address
Company or Organization	
Phone No.	

Readers' Comments — We'd Like to Hear from You
GC34-4742-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department G7IA / Bldg 062
P.O. Box 12195
Research Triangle Park, NC
27709-2195

Fold and Tape **Please do not staple** Fold and Tape

GC34-4742-00

Cut or Fold
Along Line

Program Number: 5622-717

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC34-4742-00