

**VisualAge TeamConnection V3
Frequently Asked Questions
GUI and Line Command Clients for
Unix, OS/2, and Windows 32-bit**

Document Number TR 29.3099

Angel Rivera, Janet Leskovec, Lee Perlov, Carl Anderson, Keith Wells

VisualAge TeamConnection Development
IBM Software Solutions
Research Triangle Park, North Carolina
Copyright (C) 1998 IBM Corp.
All rights reserved.

ABSTRACT

This technical report provides answers to frequently asked questions made by users of VisualAge TeamConnection Enterprise Server Version 3 with respect to the client (GUI and line commands) for Unix, OS/2 and Windows 32-bit. It is intended for both people who use TeamConnection, and also for those who help TeamConnection users.

ITIRC KEYWORDS

The following keywords are used in ITIRC to try to find this technical report. For more information on ITIRC (InfoGate) see: <http://itirc.ibm.com/>

- VisualAge TeamConnection
- client
- gui
- line commands
- Unix
- OS/2
- Windows 32-bit
- Windows 95
- Windows NT

ABOUT THE AUTHORS

ANGEL RIVERA

Mr. Rivera is an Advisory Software Engineer and team lead for the development of CMVC 2.3 and VisualAge TeamConnection. He joined IBM in 1989 and since then has worked in the development and support of library systems.

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

JANET LESKOVEC

Ms. Leskovec is a Software Engineer in the VisualAge TeamConnection development group. She began working for IBM as a Coop in 1984, and spent a large portion of her career in the Computer Aided Engineering group in Endicott, NY before transferring to Raleigh.

Ms. Leskovec earned a B.S. in Mechanical Engineering from the University of Minnesota at Minneapolis in 1985.

LEE R. PERLOV

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

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

CARL T. ANDERSON

Mr. Anderson is a Staff Software Engineer in the TeamConnection/CMVC development group. He started working for IBM in 1990 in Rochester, MN, in the Internal Tools group, and has worked for IBM's Education division and Retail & Distribution Solutions division prior to joining the TeamConnection group in 1997.

Mr. Anderson received a B.S. degree in Computer Science from Brigham Young University in 1990. He also completed three years of graduate work in the Department of Computer Science at Brigham Young University.

KEITH WELLS

Mr. Wells is an Advisory Software Engineer and team lead for the development of the GUI of VisualAge TeamConnection.

He joined IBM in 1985, working with the Federal Systems Division for contract work with the Mission Control Center with NASA.

Mr. Wells completed graduate school in Computer Science at Clemson University for the Master Degree Program.

CONTENTS

ABSTRACT	iii
ITIRC KEYWORDS	iii
ABOUT THE AUTHORS	v
Angel Rivera	v
Janet Leskovec	v
Lee R. Perlov	v
Carl T. Anderson	v
Keith Wells	vi
Figures	ix
Introduction	1
Acknowledgements	1
Notes	2
Getting this Technical Report	2
VisualAge TeamConnection Interfaces	2
Line Commands	3
Graphical User Interface (GUI)	3
Other TeamConnection Interfaces	3
Web Client	3
Lotus Notes	4
Source Code Control	4
Object Repository	4
Line Commands and GUI Overview	5
TeamConnection GUI and Line Commands Model	5
Additional Details	6
Client-Server Communication	7
Licensing Mechanisms	8
Processing TeamConnection Line Commands	8
Difference between TeamConnection user and system login	10
Printing Operations	12
Invoking tcmerge to show file differences and to merge files	12
What do I do if tcmerge will not come up?	12
Handling Error Messages from the Family Server	13
Configurable Fields	13
The teamc.log file	14

Information about the password authentication level	14
Generic Frequently Asked Questions	17
Maximum Length of the Base and Path Names	17
Testing the NLS settings for the client and the server	17
Testing the Name Resolution Using ping, host, nslookup	19
Additional examples of using ping and host	20
HP-UX: using the ping -o command to provide trace	21
Using the nslookup command	21
Looking at the TCP/IP Configuration Notebook in OS/2	22
Using ipconfig in Windows NT	22
Using winipcfg in Windows 95	22
Capturing the help output from a TeamConnection line command	23
Using environment variables with the client	23
Controlling Backup Copies Created During Extract and Checkout	24
teamc Part changes the file permissions to read-only after a checkin	25
Creating a Task to Display a Specific Tree View	25
Entering Multi-line Remarks for the Line Commands	26
Unix	26
Specifying the Initial TeamConnection Window	28
Bypass the Hosts/Services Files for a Family Name	28
Location of the Host and Services Files	29
Use of Forward or Backward Slashes in the Path Name	29
Why are backward slashes ignored in queries for path names	30
Specifying Long File Names with Imbedded Spaces	30
Searching for % and _ When Using the "like" SQL Operand	31
Filter dialogs are aimed to novice users who use basic SQL commands	32
Difference Between "Choices" and "Import" Buttons	33
Missing Files for the Choices Button	33
Can I create a user exit in my TeamConnection client?	34
Strange Error Messages Caused by Wrong Message Catalog	35
Using the TC_CASE Environment Variable	36
Can I have multiple INI files for different families?	37
In some Parts queries, it seems that the wrong workarea is used	37
CR (carriage return) and LF (line feed)	38
Using the tool fixCrLf to fix CR-LF conversion problems	38
How to enable/disable the time stamp in the GUI teamcv3.log	39
Unix Frequently Asked Questions	41
Sample Profile for a TeamConnection Unix User	41
CDE: my LIBPATH statement in .profile is not exported, why?	41
Files to Backup for the TeamConnection GUI	42
Difference Between INI and Teamcgui Resource Files	43

Setting the Fonts for the TeamConnection GUI	44
Customizing the Colors of the GUI	45
The Search Path for the Motif Resources	46
Common Usage Errors with Motif Resources	47
Hints for Isolating Motif Resource Problems	48
Which Resource is Used: doubleClickTime or multiClickTime?	49
Connecting to the X server.	49
Setting the DISPLAY to Get the Best X-Windows Performance	50
Using Telnet to Access the TeamConnection GUI	51
The GUIs Do Not Appear (there are no errors)	51
Use of .kshrc May Cause Weird Behaviors	53
Window Managers Used with the TeamConnection GUI	53
Line Command Arguments for the Standalone TeamConnection GUI	54
Customization hints for the CDE environment	54
How to replace Editor icon with Terminal icon in CDE console.	54
How to have all window icons inside a single Icon box.	55
How to change the title name of a dtterm window	55
How to customize the context menu from the desktop.	56
Customizing the Motif Environment for Left-handed Users	56
Intel Frequently Asked Questions	59
Files to backup from the TeamConnection GUI	59
Important Warnings with Windows 32-bit Logins	59
Restriction on file extracts into OS/2 FAT drives	60
Changing the Fonts in the TeamConnection GUI	60
Problems with APM Drivers When Named Pipes are Suspended in OS/2	60
Enabling the Netscape Navigator for Java on OS/2	61
Appendix A. Copyrights, Trademarks and Service marks	65

FIGURES

1. Model: TeamConnection GUI - line command - server	5
2. teamcgui line command arguments.	54

INTRODUCTION

This technical report provides answers to frequently asked questions made by users of VisualAge TeamConnection Enterprise Server Version 3 with respect to the client (GUI and line commands) for Unix, OS/2 and Windows 32-bit.

It is intended for both people who use TeamConnection, and also for those who help TeamConnection users.

This technical report is organized as follows:

- An overview of the TeamConnection Line Commands and GUI. See “Line Commands and GUI Overview” on page 5.
- Information that is not platform specific. See “Generic Frequently Asked Questions” on page 17.
- Answers to questions that are specific to running TeamConnection on Unix platforms. See “Unix Frequently Asked Questions” on page 41.
- Answers to questions that are specific to running TeamConnection on Intel platforms. See “Intel Frequently Asked Questions” on page 59.

The focus in this document is VisualAge TeamConnection Enterprise Server Version 3. Whenever is necessary, the differences with respect to Version 2 will be discussed.

ACKNOWLEDGEMENTS

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

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

- Tim Orlowski, who develops code for TeamConnection.
- Naomitsu Ishii, IBM Japan, Translation tester.

We want to thank Dodde Stark for editing this technical report.

NOTES

This document reflects the questions of our users. Because our Unix users tend to do more customization of their environments through tools, such as shell scripts, many of the answers are primarily for Unix.

We recognize that X-Servers exist for non-Unix platforms. However, we only tested using the X-Servers that come with each Unix operating system. You may be able to apply our suggestions to X-Servers that run on Intel platforms, such as IBM's PMX for OS/2 or Hummingbird Exceed for Windows 32-bit.

In this document, the term "Unix" is used to describe in generic terms the following operating systems: AIX, HP-UX and Solaris.

Similarly, the terms "Windows" and "Windows 32-bit" are used to describe in generic terms the following operating systems: Windows 95 and Windows NT.

Finally, the term "Intel" is used to describe in generic terms: OS/2, Windows NT, and Windows 95.

GETTING THIS TECHNICAL REPORT

The most up to date version of this technical report can be obtained from the IBM VisualAge TeamConnection Enterprise Server:

- Library home page by selecting the item Library at URL:
<http://www.software.ibm.com/ad/teamcon>
 - FTP site by accessing the URL:
<ftp://ftp.software.ibm.com/ps/products/teamconnection/papers>
- See the file README.index.txt for details.

VISUALAGE TEAMCONNECTION INTERFACES

The TeamConnection client has several interfaces. Only the interface provided by the Line Commands and the Graphical User Interface, will be discussed in detail in this document.

Line Commands

The line commands can be entered from a command prompt (DOS prompt or Unix shell). For example, “teamc Part” gives information about the TeamConnection parts. There are no differences between the Intel and Unix line commands.

Graphical User Interface (GUI)

The Graphical User Interface (GUI) is invoked by the command “teamcgui”. In this document, this interface will be referred to as the TeamConnection GUI, TC GUI or simply the GUI.

What are the differences between Intel GUI and Unix GUI?

The significant difference between the Intel GUI and the Unix GUI is the functionality of the “Split Bar” in the Details view:

- In the Intel GUI, windows that provide a list of objects, such as defects, a vertical bar, or “Split Bar”, allows the user to “anchor” certain fields at the left side of the screen. When the user scrolls to the right to see other fields, the fields to the left of the bar do not move. The other fields will move to the left, and eventually disappear. The user can then scroll all the way to the right, and still know the defect number (by default, an “anchored” field) associated with a particular line.
- The Split Bar does not work in the Unix GUI.

OTHER TEAMCONNECTION INTERFACES

The following other interfaces for the TeamConnection client will be briefly described here, but will not be discussed in the rest of this document.

Web Client

The Web Client is a TeamConnection Client that provides a subset of the functions available in the Intel or Unix client. The name Web Client is a slightly misleading name, because it really is a specialized Web Server built into the TeamConnection family server.

The Web Client provides access to a TeamConnection family from a frame-enabled web browser (such as Netscape Navigator or recent versions of Microsoft Internet Explorer), without having to install any TeamConnection components on the client system.

The function missing from the Web Client in Version 3.0.0 is the ability to manipulate part contents, that is, it is not possible to check-in, check-out, create or extract parts, although it is possible to view part contents.

Note: This extra ability in handling files was added in fixpak 3.0.1.

All that is required to access a TeamConnection family using the Web Client is the hostname of the family server (or its IP dotted decimal address) and the port number of the family. The URL to be used is:

http://hostname:port

Such as: http://tcaix05.raleigh.ibm.com:2600

To disable the Web client, set the following environment variable in the TeamConnection server:

TC_WWWDISABLED

Lotus Notes

The Lotus Notes interface allows you to use Lotus Notes to manage requirements and test cases (for example), and then manage the corresponding TeamConnection features and defects. This was added in Version 3.

Source Code Control

The Source Code Control (SCC) API, in which other applications such as PowerBuilder can checkout and checkin parts from TeamConnection was added in Version 3, although in the latest refreshes of Version 2 we delivered beta code.

Object Repository

The Object Repository API, allows other applications to use low level API method calls from TeamConnection to navigate thru the model and handle objects inside TeamConnection.

LINE COMMANDS AND GUI OVERVIEW

TEAMCONNECTION GUI AND LINE COMMANDS MODEL

The Figure 1 shows the overall model for the VisualAge TeamConnection components.

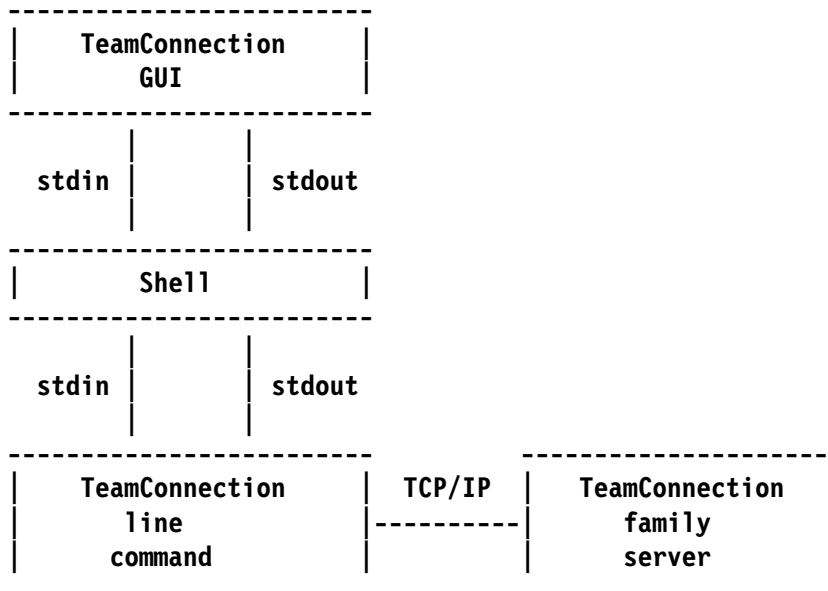


Figure 1. Model: TeamConnection GUI - line command - server

The TeamConnection GUI interacts with the user and behind the scenes it uses the TeamConnection line commands. In other words, the TeamConnection GUI is an application that sends TeamConnection line commands on behalf of the user. The TeamConnection GUI logs all commands it sends into a file called "teamc.log" (See "The teamc.log file" on page 14).

The line commands interact with the TeamConnection family server. This means that the GUI never communicates directly with the TeamConnection family server.

The output from the "stdout" pipe from the TeamConnection line command is routed to the main TeamConnection GUI windows. In a similar manner, the output from the "stderr" pipe from the TeamConnection line command is routed to a TeamConnection error box.

The GUI also uses other commands like “tcmerge”, a TeamConnection utility for showing differences between versions of a file, “print”, for printing, and user written shell scripts as defined in the task main window.

Additional Details

The TeamConnection GUI is just a fancy way to prepare the TeamConnection line commands. The TeamConnection GUI stores the actual line commands that were invoked at the bottom of a log file. The name of the log file is specified on the Setup page of the Settings notebook in the “Log file” field. The default is teamc.log.

The log file contains the TeamConnection line commands that were executed. However, environment variables, such as the TeamConnection Family Name and Become User Name, are the ones specified on the Environment page of the Settings notebook. They are not shown in the teamc.log file.

The environment that surrounds the line commands may be different than the environment used by the GUI. That is, the TeamConnection environment variables for the line commands are used only for the line commands, as defined in:

Unix .profile

OS/2 C:\AUTOEXEC.BAT

Windows 95 C:\AUTOEXEC.BAT

Windows NT System Registry

When the GUI is invoked for the first time, the default values for the GUI Settings are taken from the corresponding TeamConnection environment variables defined for the line commands. The GUI user may then change them in the Settings notebook. The new values will only be used by the GUI. These variables and values are stored in the file:

Unix \$HOME/teamcv3x.ini

OS/2 C:\os2\teamcv3.ini

Windows C:\windows\teamcv3.ini

For example, if you set TC_FAMILY=fam1, and TC_BECOME=usr1, fam1 and usr1 will be used for the line commands without specifying the -family and -become flags. Then, the first time you invoke the GUI, the values for Family and Become User will be fam1 and usr1 respectively. If you want to use a different family and user in the GUI, say fam2 and usr2, you change the values on the Environment page of the Settings Notebook accordingly. Now you will be using fam1 as usr1 in the line commands and fam2 as usr2 in the GUI. In this case, if you want to reuse a TeamConnection command from the log file, you will need to add “-family fam2 -become usr2” in order to match the command used by the GUI. If you do not add

them, then the line command will assume -family fam1 and -become usr1, and the results will be different.

Using the TeamConnection Command Line Interface, the commands are typed on one or more lines and finished with the Enter key.

The commands, action flags, attribute flags and arguments are case sensitive.

You can control the following from the OS/2 and Windows 32-bit command prompts and from the Unix shells:

- wildcard substitution
- shell variable substitution
- background/foreground processing
- input and output redirection
- pipelining
- command aliasing (except when using the Unix Bourne shell)

Client-Server Communication

The client-server communication is implemented using TCP/IP sockets.

The TeamConnection Client creates packets of data depending on the user's request. The TeamConnection family server creates packets of data depending on the result of the user's request.

For each TeamConnection action, there is a specific TeamConnection protocol between the client and server.

- When servicing a request, the client knows how many packets of data it must create, and the server knows how many packets of data it must receive.
- When sending the results obtained from a request, the server knows how many packets of data it must create, and the client know how many packets of data it must receive.

Transmission of files between the client and server is handled by the file transfer interface.

- The file is divided into blocks and transmitted as packets of data through the socket interface.
- Both the client and the server use temporary files created in the /tmp directory on their respective machines. These temporary files hold the contents of the file being transmitted.

LICENSING MECHANISMS

CMVC, the predecessor product of VisualAge TeamConnection, uses a runtime licensing mechanism for the client, which is called the iFOR license management system (previously known as NetLS). This is an enforcer mechanism. If the client cannot obtain a license, the client cannot communicate with the family server.

In contrast, the TeamConnection clients do not use this licensing mechanism. Instead, a server-based mechanism is available to monitor (but not to enforce) the use of licenses. For more information, see the documentation for the command “tcllicmon” (VisualAge TeamConnection License Monitor).

PROCESSING TEAMCONNECTION LINE COMMANDS

1. The message catalog file, `$TC_HOME/nls/cfg/$LANG/teamcv3.cat`, is opened.

No errors or warnings are issued if the file cannot be opened. In that case, the hard-coded English messages that are packaged with the executable during the compilation process will be used.

2. Check for input parameters.

If no input parameters are provided, the command usage statement is displayed (on `stderr`). See “Capturing the help output from a TeamConnection line command” on page 23 for details on how to show the usage help text one screen at a time.

3. Obtain the identity of the user.

Unix In Unix, the identify of the user is obtained with the `getuid` and `getpwuid` system calls. These system calls make use of the `/etc/passwd` file. The `/etc/passwd` file must have at least 664 permission so that the user's identity can be properly obtained.

OS/2 The `USER` environment variable is used.

Windows NT The system login value is used.

Windows 95 If the user logs in, then this value is used. Otherwise, the `TC_USER` environment variable or the “User ID” field of the Settings->Environment page is used.

Note: For important details on the case used for the login id in Windows 32-bit, see “Important Warnings with Windows 32-bit Logins” on page 59.

4. Read applicable environment variables.

All TeamConnection environment variables that are applicable for the specific command are read.

5. Validate input parameters.

- There can only be one action flag per command.
- All attribute flag arguments must be valid.
- All required flags must be specified or their value must be obtainable from TeamConnection environment variables.

6. Connect to the server.

The client connects to the server using the server's address and port number. The client creates a socket to talk to the server and then the connection is established.

7. Authenticate the user.

Host list The default authentication level is HOST_ONLY, in which a host list entry, consisting of a system login id, a TeamConnection user id, and a host name, is necessary.

The gethostbyname system call is used to obtain the server's host entry. If the port number is given, it is used. Otherwise the getservbyname system call is used to find the family's port number in the client's "/etc/services" file.

Password If the PASSWORD_OR_HOST authentication is enabled and if the user has a valid host list entry, then the password is ignored. If there are no valid host list entries, then do the processing mentioned in the next paragraph.

If the PASSWORD_ONLY authentication level is enabled, then the user needs to be logged in first in order for the server to respond to the commands.

If the authentication fails, an error is displayed and the client disconnects from the server.

8. Send request to the server and wait for response.

The request is sent to the server and the client waits for the request to complete. If any of the attribute flags contain more than one argument, a separate request is sent to the server for each. All requests are processed sequentially. If any of them fail, an error message will be stored and the next request will be handled.

9. Server processes the request and handles user exits.

The server processes the request from the client. Specified user exits will be invoked.

The following detailed example illustrates what happens during a Part extract.

- a. When doing a file extract, the routine cl_txFileExtract in niClient.c is executed.
- b. The routine sv_txFileExtract is executed, where User Exit 0 is done.
- c. The actual routine txFileExtract from txFile.c is executed, where User Exit 1 is done.
- d. The file extraction is done, followed by User Exit 2.

10. Issue verbose messages, error messages, or reports if applicable.

If the request completed successfully and the verbose option was specified, a message is displayed via stdout on the client machine.

If the request completed with errors, a message is display via stderr on the client machine.

If a report was requested, the contents of the report are routed to stdout.

11. Disconnect from the server.

The client disconnects from the server by closing the TCP/IP socket.

DIFFERENCE BETWEEN TEAMCONNECTION USER AND SYSTEM LOGIN

The purpose of this section is to clarify the difference between the TeamConnection user id and the system login.

The TeamConnection user is an object that is known only to the TeamConnection family. It is represented by the variable TC_BECOME and by the field "Become user" in the Settings notebook.

The system login is the actual identification used when interacting with the Unix login shell or the Windows 32-bit login mechanism. This value can be known to all the applications running in the system.

Because OS/2 does not have an explicit login mechanism and because Windows 95 does not require the user to login, it was necessary to allow the user to explicitly specify the equivalent of the system login by means of the TC_USER environment variable. It is shown as the "User id" in the Settings. There is an optional login in Windows 95, if it is used, then the GUI will use it.

The following example illustrates the difference between these concepts.

A user has 2 TeamConnection user id's but only one system login:

- One TeamConnection user is "su_angel" (a superuser, which is not used too frequently).
Notice the use of the prefix "su_" which indicates that it is a superuser. The use of prefixes is recommended for quick identification of TeamConnection superusers.
- The other TeamConnection user is "rivera" (a normal user, with no superuser powers).
- When the user logs into the Unix host "carcps06", the system login is "rivera". There is no system login of "su_angel" on that Unix host.

The user can login into Unix as the system user “rivera” and then use the TeamConnection user of “rivera” (using the environment variable TC_BECOME=rivera) to interact with TeamConnection as a normal user.

In case the user needs TeamConnection superuser powers, then as system login “rivera” the user can specify the TeamConnection user “su_angel” (TC_BECOME=su_angel).

Notice that the user employs the same system login “rivera” but the user can choose which TeamConnection user to “become”.

In order to be able to interact with TeamConnection as mentioned above, the user needs the following TeamConnection host list entries:

- System login=rivera from host=carcps06, TeamConnection userid=rivera.
- System login=rivera from host=carcps06, TeamConnection userid=su_angel.

To extend this scheme a bit more, let's suppose that in order to do extractions from TeamConnection, the user needs to login into the host “carcps21” as system login “build”, but using the TeamConnection user “rivera”. This means that the user needs the following TeamConnection host list entry:

- System login=build from host=carcps21, TeamConnection userid=rivera.

In short, TeamConnection allows the ability to use the same Unix login or Windows 32-bit login and then you can specify which TeamConnection role you want to play (and vice versa, to use the same TeamConnection role from different system logins/hosts).

Side trip:

One cause of confusion is that the TeamConnection “teamc Host” command does not use the correct terminology (but fortunately the TeamConnection “teamc Host” GUI dialog uses the correct one). For example, if you need to create a host entry for the system login=build from host=carcps06, as TeamConnection user=rivera, then you need to use the TeamConnection “teamc Host” command as follows:

```
teamc Host -create build@carcps06 -login rivera  
                ***** TeamConnection User  
                ***** host  
                ***** system login of the Unix account
```

Wish list: It would be nice if the “-login” flag was changed to “TeamConnectionUserId” or something better, to reflect the correct usage, but this is not possible now.

PRINTING OPERATIONS

For printing operations, the default command is:

Unix lpr

OS/2 print

Windows 95 The default printer.

Windows NT The default printer.

When printing in Windows, the corresponding printing dialog for the default printer will be shown. The user can specify additional details for printing.

INVOKING TCMERGE TO SHOW FILE DIFFERENCES AND TO MERGE FILES

If the “Compare command” on the Setup page of the Settings notebook is set to “tcmmerge”, the new merge tool will be used when comparing two parts to show their differences and when reconciling differences during concurrent development.

It can also be use separately from the GUI to compare and merge up to three files.

What do I do if tcmmerge will not come up?

Question:

What do I do if tcmmerge will not come up?

Answer:

If tcmmerge does not come up as expected, there are few possible things to check.

1. Make sure that the appropriate level of java is installed on the machine where tcmmerge is to run, and check that the CLASSPATH environment variable contains the complete path to the classes.zip file.
2. Make sure that the HELP environment variable points to the directory containing the tcmmerge.htm file.

HANDLING ERROR MESSAGES FROM THE FAMILY SERVER

Error messages are obtained from the message catalog file `$TC_HOME/nls/msg/$LANG/teamcv3.cat` provided the **LANG** environment variable is set to a valid locale installed in your system.

Note: If LANG is set to a locale value that is not installed in your system, then you will get an error message saying that LANG is set to NULL. You can use the command “`locale -a`” to see what are the valid locales in your system.

If this message catalog file cannot be found, then the error messages will come from the English hard coded defaults stored in the executable.

The messages from the server are sent as a string through the `stderr` pipe. The TeamConnection GUI blindly dumps them in an error box, with no intervention.

CONFIGURABLE FIELDS

Configurable fields are attributes about certain TeamConnection objects, such as Defects and Features, that the family administrator manages.

For instance, the family administrator may decide that it is important to know the phase of the moon under which a defect was discovered. He would create a configurable field called something like “`moon_phase`” and specify the data type and possible values. The TeamConnection GUI keeps track of the information that is necessary to display the columns and the fields for the configurable fields.

If this information has not yet been gathered about these fields, the TeamConnection GUI will issue a command appropriate for the given object. For example, for defects, the command would be:

```
teamc defect -configInfo -raw
```

Similarly, the TeamConnection GUI handles the information about the choice items (also called configurable field types) by issuing the following command:

```
teamc Report -raw -view Config \  
-where "1=1 order by choiceOrder, configType"
```

THE TEAMC.LOG FILE

The TeamConnection GUI stores the commands that are issued in a log file. It could be a client command (`teamc defect -open ...`) or a call to an executable (`tcmerge file1 file2`). Line commands that are executed from a system command prompt, outside the GUI, are not stored in the log file.

The default name for the log file is **teamc.log**, however, the user can specify the name of this file in the "Log File" field on the Setup page of the Settings notebook. If the absolute path is not specified, then the log file will be opened relative to the directory set in the "Working directory" entry in the Environment page of the Settings notebook. If that is not specified, the log file will be put in the user's home directory. If the log file cannot be opened, a warning message is posted.

Starting with fixpak 3.0.1, the time stamp when the command was issued is shown in the `teamc.log`. The reason is that during troubleshooting activities, it is necessary to correlate the `audit.log` in the family server with the specific requests that were submitted by the clients, and without the time stamp in the `teamc.log`, it is very difficult to perform this correlation.

Note that for Client commands that have a **-remarks** flags, the text of the remarks is not stored in the log file. The text is sent directly through the pipe. The log would show, for example,

```
teamc defect -note 1234 -remarks -
```

For more details, see "Entering Multi-line Remarks for the Line Commands" on page 26.

INFORMATION ABOUT THE PASSWORD AUTHENTICATION LEVEL

- Does TeamConnection encrypt passwords that are sent/received?

The VisualAge TeamConnection client and server encrypt the passwords that are send/received.

- Does TeamConnection encrypt other information that is sent/received?

Only the passwords are encrypted. There are no facilities inside TeamConnection to encrypt other data.

- How long does a user stay logged on after logging in with a password?

Until the following occurs:

- The user types "`teamc tclogin -logout`".
- The user reboots the workstation client (this kills the LoginManager process in the client, erasing the passwords).

- The user explicitly kills the local LoginManager: "teamc tlogin -KILL_TCLOGINMGR"
- The family is stopped and restarted; this erases the "login tokens" that the family knows.

That is, if family administrators want to log off everybody at once, then, they can stop and restart the families.

- Can a session be monitored to get the password? Can a user's session be "hijacked"?

When the client issues the tlogin command and sends the password (encrypted) to the server, the server determines if the user id and the password are valid, if so, then the server generates a "login token" (which is kept in memory) and then sends it to the client. The Login Manager of the client receives the token and then attaches the token to any subsequent transaction.

The server compares the token received from the client with the copy of the token stored in memory and if they match, then the server proceeds to handle the request.

The server also keeps track from which host the user has logged in.

- If it is possible to change the sessions settings for users that log on?

The TeamConnection family has a file named "\$HOME/config/security" which determines the family-wide settings, which affects all users. The only thing that you can change for a given user is the password.

- Is it possible to enforce a syntax and/or expiration date for the passwords?

No.

- Authentication Resynchronization.

If a client logs into the Team Connection server and the server is recycled, sometimes the client and the server cannot resynchronize the authentication. All client actions are rejected by the server.

To solve the problem, terminate all of the client programs, including any teamc commands, the GUI, and the login manager. After the clients have been terminated, restart the client programs and log back into the server. You may need to do this twice, because when a client has logged onto the server and attempts to log in again, the server automatically logs the client out; the client may not synchronize with the server correctly.

Whenever the client logs into the server, a login manager is started automatically on the client. This login manager runs in the background.

To terminate this manager, issue the following command:

```
teamc tlogin -KILL_TCLOGINMGR
```

Note: The information above applies to the line commands. The web client and the direct API calls have a very similar behavior (encryption of the passwords) but there are minor differences.

GENERIC FREQUENTLY ASKED QUESTIONS

In this section several frequently asked questions are presented which are common to all of the TeamConnection clients: Unix, OS/2 and Windows 32-bit.

MAXIMUM LENGTH OF THE BASE AND PATH NAMES

Question:

What is the maximum length of the base and path name of a file?

Answer:

We recognize that the information provided in the tables for the sizes might be a bit confusing and that is the reason we provide this explanation.

The base name is part of the path name of a file. The base name does not have any directory separators (forward slash, “/” in Unix or backward slash “\” in Intel). For example, “source/file1.txt” is the path name that has 16 characters, including the forward slash that separates “source” from the base name “file1.txt” (which has only 9 characters, including the period).

```
source/file1.txt
***** -> base name ( 9 characters)
***** -> path name (16 characters)
```

The base name may contain a maximum of 63 characters. The path name, which includes the base name, may contain a maximum of 195 characters.

Furthermore, the path name or the base name may contain blank characters. If so, when using line commands it is necessary to enclose the path name between double quotes, which are NOT counted toward the limit. For example, “long directory/file name with blanks” is a valid name.

TESTING THE NLS SETTINGS FOR THE CLIENT AND THE SERVER

Question:

How do I test to be sure that the NLS settings, and the message catalog, being used by the client are the correct ones?

Answer:

You can use the command “teamc report -testServer” to verify if the TeamConnection family server is accessing the message catalogs and the command “teamc report -testClient” to verify if the TeamConnection client is also accessing the message catalogs.

An example of the output from “teamc report -testServer” is shown below:

```
$ teamc report -testServer
```

```
Connect to Family Name:          vatc
Server TCP/IP Name:             tcaix05.raleigh.ibm.com
Server IP Address:              9.67.238.8
Server TCP/IP Port Number:      2600
```

```
Server Specific Information -----
Product Version:                3.0.0
Operating System:               AIX
Message catalog language:       English
Server Mode:                    non-maintenance
Authentication Level:           PASSWORD_OR_HOST
```

An example of the output from “teamc report -testClient” is shown below:

```
$ teamc report -testClient
```

```
Product Version:                3.0.0
Message catalog language:       English
Environment variables:
Variable:                       Setting:
-----
TC_USER                         Administrator
TC_BECOME                       rivera
TC_FAMILY                       tcocto@carcps22@1300
TC_RELEASE                      v300
TC_TOP
TC_WORKAREA
TC_BUILDPOOL
```

For more information about the message catalogs, see the Technical Report “VisualAge TeamConnection Version 3: NLS and DBCS”.

TESTING THE NAME RESOLUTION USING PING, HOST, NSLOOKUP

Question:

How do you test the name resolution using ping, host, and nslookup?

In some cases, the host name is not fully resolved by the domain name server, and even though the “ping” command shows that the host is reachable, TeamConnection does not seem to reach that host name. What else can be done to verify that the host name is properly configured?

Answer:

The use of “ping” is the first step to ensure that a host name is properly defined in the network, and that the specified host is connected and operational.

However, in some situations (mainly due to aliases and incomplete TCP/IP configurations) it is possible for the “ping” command to provide a valid output, even though the host name is not fully configured.

Furthermore, with the default authentication level of HOST_ONLY (that is, host list entries), TeamConnection relies on the full reciprocity of a host name and its corresponding IP address.

In these cases, we have to take a second step in the verification process, using the “host” command, which is available on AIX and on OS/2. This command is not provided by TCP/IP on Solaris, HP-UX or Windows. However, on the CD-ROM for TeamConnection, we provide the command “tchost”, which is a reduced version of the AIX “host” command.

Platform	Location on CD-ROM
Windows NT	samples/tchostw.exe
HP-UX	misc/tchost.hp
Solaris	misc/tchost.sol

The “host” command is used to ensure that a specified host name can be resolved to an IP address, and correspondingly, that the same IP address is resolved to the specified host name.

We have seen cases in which a host name is resolved to an IP address, but the IP address is resolved to a different host. In these cases, it is important to first fix the TCP/IP configuration problem. **Do NOT attempt to use TeamConnection until the TCP/IP problem is fixed.**

We have also seen this problem when using DHCP in Windows 32-bit where the host receives a dynamic IP address which exists only for the duration of the session. In this case, it is

better to use the password authentication level (such as PASSWORD_ONLY or PASSWORD_OR_HOST) for TeamConnection.

The recommended verification sequence is:

1. Use the “ping” command. In this example the host “oem-ppc1” is up and running:

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

2. Using the “host” command to resolve the host name into one IP address, for example:

```
$ host oem-ppc1
oem-ppc1.raleigh.ibm.com is 9.67.163.37
```

3. Use the “host” command to resolve the IP address obtained in the previous step into the host name host name, for example:

```
$ host 9.67.163.37
oem-ppc1.raleigh.ibm.com is 9.67.163.37
```

4. Ensure that the output from the two previous steps indicate the same host name and the same IP address.

Note: You need to use the host command twice, once with the host name and the other time with the IP address.

Additional examples of using ping and host

Other examples of using the ping and host commands in different situations are shown below:

- Ping example: the host “rivera” is defined in the network but it is not operational (it is not powered on):

```
$ ping rivera
PING rivera.raleigh.ibm.com: (9.37.194.163): 56 data bytes
Ctrl-C
----rivera.raleigh.ibm.com PING Statistics----
49 packets transmitted, 0 packets received, 100% packet loss
```

- Ping example: the host “xyz123” is not defined in the network:

```
$ ping xyz123
0821-062 ping: host name xyz123 NOT FOUND
```

- Host example: the host “rivera” is defined in the network but it is not operational (it is not powered on).

```
$ host rivera
rivera.raleigh.ibm.com is 9.37.194.163
$ host 9.37.194.163
rivera.raleigh.ibm.com is 9.37.194.163
```

- Host example: the host “xyz123” is not defined in the network:

```
$ host xyz123
host: 0827-801 Host name xyz123 does not exist.
```

- Host example: the IP address “1.2.34.5” is not defined in the network:

```
$ host 1.2.34.45
host: 0827-803 Cannot find address 1.2.34.5.
```

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

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

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

Notice the last 3 lines of the output.

Using the nslookup command

Another TCP/IP command that can be used in the diagnostic process is the command “nslookup” (Name Server Look Up) which skips the local hosts file and directly queries the network name server.

It provides the host name and IP address of the name server, followed by the target host name and IP address. For example:

```
$ nslookup cordoba
Server: ra1ns1.raleigh.ibm.com
Address: 9.37.0.5
```

```
Name: oem-ppc3.raleigh.ibm.com
Address: 9.37.199.98
Aliases: cordoba.raleigh.ibm.com
```

LOOKING AT THE TCP/IP CONFIGURATION NOTEBOOK IN OS/2

In OS/2, you can look at the TCP/IP Configuration Notebook to find out the details on the configuration of TCP/IP.

USING IPCONFIG IN WINDOWS NT

Windows NT provides the following utility to find out the actual TCP/IP values used with the current communication session in your machine:

1. Open a Command Prompt window.
2. Issue: ipconfig

Windows NT IP Configuration

Ethernet adapter NdisWan4:

```
IP Address . . . . . : 9.37.194.164
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 9.37.192.1
```

USING WINIPCFG IN WINDOWS 95

Windows 95 provides the following utility to find out the actual TCP/IP values used with the current communication session in your machine:

1. Select Start -> Run...
2. From the Run window, enter the name: winipcfg
3. You will see a window titled "IP Configuration" and will have data such as:

Ethernet Adapter Information

PPP Adapter

```
Adapter Address: 44-45-53-54-00-00
IP Address: 32.225.77.105
Subnet Mask: 255.0.0.0
Default Gateway: 32.225.77.105
```


4. You can press the button "More Info >>" and then you will see the additional information about the host, such as:

Host Information

Host Name: xxx.raleigh.ibm.com
DNS Server: 9.67.1.5

CAPTURING THE HELP OUTPUT FROM A TEAMCONNECTION LINE COMMAND

Question:

When I try to get the help output from a TeamConnection line command (for example, when entering the "teamc part" command by itself), I see the resulting text scrolling off the screen. Why can't I pipe it to the "more" utility (that is, "teamc part | more") to see one screen at a time?

Answer:

The help text from a TeamConnection line command goes to stderr and not stdout.

To do what you want, you need to do the following:

```
teamc part > part.out 2>&1  
or  
teamc part 2>&1 | more
```

The "2>&1" redirects stderr ("2>") to the same place as stdout. In the first example to the file part.out, and in the second example, to the pipe for "more".

USING ENVIRONMENT VARIABLES WITH THE CLIENT

Question:

Can I use environment variables with the TeamConnection line commands and inside the TeamConnection GUI?

Answer:

In the TeamConnection GUI you can use environment variables such as TC_RELEASE or TC_BECOME in the task list, in the Filter dialogs, and in the space for the line command in the GUI (above the footer).

You can use environment variables with the TeamConnection line commands in the same way.

In OS/2 and Windows 32-bit, you will need to specify the environment variable between % and %, such as:

```
login in ('%TC_BECOME%')
```

In Unix, you will need to specify the environment variable by using a prefix of \$, such as:

```
login in ('$TC_BECOME')
```

CONTROLLING BACKUP COPIES CREATED DURING EXTRACT AND CHECKOUT

Question:

The teamc Part command creates a backup copy of the files involved in an extraction or checkout, changing the suffix of these files. Can I specify whether or not these backup files are created?

Answer:

To preserve any potential changes that the user could have made to the files on their workstation, the “teamc Part” command creates a copy of a file being extracted or checked out. This copy is renamed with a suffix that begins with \$ and uses 2 characters from the original suffix, if any. This identifies it as a backup in a normal directory query. Some examples are shown below:

- If the original file is “file.c”, then the backup file is “file.\$c”.
- If the original file is “file.html”, then the backup file is “file.\$ht”.

The TC_BACKUP environment variable can be used to control whether or not backup files are created by teamc Part. It can be changed within the GUI from Extract page of the Settings notebook. The default is to create a backup file. Setting it to “No” specifies that no backups will be created.

You cannot control the naming of the suffix for the backup file.

This function was added in Version 3.

TEAMC PART CHANGES THE FILE PERMISSIONS TO READ-ONLY AFTER A CHECKIN

Question:

The teamc Part command changes the file permissions involved in a checkin to read-only. Can I specify whether or not these files permissions are changed?

Answer:

The main reason behind changing the file permissions to read-only after a checkin, is to keep the user from inadvertently changing files that they do not have checked out from TeamConnection.

It is strongly recommended that the user should not just change the file permissions of the local file and then obtain a lock of the file in TeamConnection, because there is no guarantee that the local copy of the file in the workstation will have all the changes made by other users. The only way to guarantee it, is to checkout the file.

Change the file permissions upon checkin is controlled by the TC_MODPERM environment variable. This can be changed on the Extract page of the Settings notebook. The default is to change the file permissions to read-only.

Furthermore, during check-in, you can specify to retain the lock. This action will not change the file permissions of your file in your workstation.

This function was added in Version 3.

CREATING A TASK TO DISPLAY A SPECIFIC TREE VIEW

Question:

Is there a way to create a task that displays a tree view (such as the component structure) that has some items expanded and other items not expanded? I want to avoid displaying the unexpanded tree view, before expanding the items I want to see.

Answer:

The GUI does not have the facility to remember which items were expanded and which were not. Thus, when you display a window that has a Tree view, then all the items shown in the window will not be expanded.

ENTERING MULTI-LINE REMARKS FOR THE LINE COMMANDS

Question:

How do I put multiple-line remarks, as opposed to one long line, into a command like “teamc Defect -open”?

Answer:

The TeamConnection GUI makes it easy to imbed newline characters, that is, Carriage Return (CR) and Line Feed (LF) characters into remarks, but the command line does not give you explicit instructions on how to do it.

Look up “Using standard input for arguments” on page 6 of the VisualAge TeamConnection Commands Reference Version 3 manual. You can use '-' as the value for remarks in a TeamConnection line command, then redirect the standard input. You are allowed one “-” symbol for which TeamConnection substitutes the value of stdin.

You can use the pipe or input-redirectation as follows in all platforms. The remarks.txt file contains the text of your comment.

- All platforms

Conclude the input string with Ctrl-Z in the first column of a line. Here is an example:

```
teamc Defect -open ... -remarks -  
This is line one  
and this is line two  
<Ctrl-Z>
```

- Unix:

```
teamc Defect -note 1234 -remarks - < /tmp/remarks.txt  
or  
cat /tmp/remarks.txt | teamc Defect -note 1234 -remarks -
```

- Intel:

```
teamc Defect -note 1234 -remarks - < c:\tmp\remarks.txt  
or  
type c:\tmp\remarks.txt | teamc Defect -note 1234 -remarks -
```

Unix

Unix makes it very easy to imbed newline characters into a string:

- Escaping the newline:

You can escape a newline character with a backslash, '\'. For example:

```
Defect -open ... -remarks "This is line one \  
and this is line two"
```

- You can also redirect from standard input:

You can define an “end-of-file” character string (such as !EOF), then redirect from standard input. This uses a feature of the TeamConnection command line interface to support redirection of standard input. Here is an example:

```
teamc defect -open -remarks - <<!  
> this is a long  
> description of the problem  
> !
```

You will be prompted with the character >, but the line will continue and you can enter the additional text. To end the remark, you need to enter “!”

Quoting rules in Unix shell scripts

In the above examples, quotes (") were used. There are some important quoting rules to keep in mind when using Unix:

- No quotes allows the shell do anything it wants.

There is a very long list of operations performed on the commands you type in a command shell. The most obvious are the ones listed below. In only a few cases is the order really important. If the two examples below are not sufficient, look at a book on Unix shell scripts.

- Double quotes delay interpretation until after filenames are expanded.

This is a subtlety, but in order for many commands to interpret a string like "*.c", it is necessary to avoid letting the shell convert the string to something like: a.c, b.c, c.c. That is what double quotes do for you.

However, double quotes do not prevent the shell from value substitution. For example, "\$HOME" will be substituted for something like: /home/guest

Double quotes are usually what you want to use.

- Use single quotes to avoid names that begin with a dollar sign (\$) to be interpreted in the current shell.

A string surrounded by single quotes, like '\$HOME', will be displayed exactly as seen, without \$HOME being replaced with the value for the environment variable. This is useful when you want to pass something to a shell script that is not interpreted at all. If you pass '\$HOME' as an argument to a shell script, it will arrive as \$HOME and be subject to interpretation there.

SPECIFYING THE INITIAL TEAMCONNECTION WINDOW

Question:

Is it possible to start the TeamConnection GUI directly in a relevant object windows?

Answer:

No, the TeamConnection Tasks window is the window that always appear when invoking TeamConnection. However, we provide the Action pulldown that allow users to manipulate specific TeamConnection objects from the Tasks window.

BYPASS THE HOSTS/SERVICES FILES FOR A FAMILY NAME

Question:

Is it possible to bypass the hosts and services files when specifying a TeamConnection family name from the TeamConnection client?

Answer:

You can always use the long family name that has a format:

familyName@hostName@portNumber

Such as:

vatc@tcaix05.raleigh.ibm.com@2600

This will bypass the need to update the etc/hosts and etc/services files when using the TeamConnection client. However, for the TeamConnection family server, you must update the services file in order to properly assign the port number to the TeamConnection family. For details on the location of these files, see "Location of the Host and Services Files" on page 29.

If you want to avoid using the long family name, you must update the etc/services file and optionally the etc/hosts file in case your family is not registered in the name server.

Note: If you specify the long family name with the 3 fields, the 1st field is really ignored. The hostName and the portNumber provide the system with all the information to connect to the socket used by the family. Thus, if you specify the correct family name but a wrong port number, the client will ignore the family name and will try to connect to a family specified by the incorrect port number.

LOCATION OF THE HOST AND SERVICES FILES

Question:

Where are the host and services files?

Answer:

The “hosts” file is used to indicate the IP address of a host and aliases for that host. The “services” file is used to indicate the port number of the service (TeamConnection family).

These files are located in:

Unix /etc/hosts and /etc/services

It is recommended that only the system administration utilities be used to update these files, such as smit in AIX and sam in HP-UX.

OS/2 They are located in the directory described in the environment variable “SET ETC”

Windows 95 They are located with the operating system, for example C:\WINDOWS. The “hosts” file by default does not exist, but it can be constructed from the sample “Host.sam”.

Windows NT They are located in the SYSTEM32\DRIVERS\ETC subdirectory, where the operating system is located. The default is C:\WINNT.

USE OF FORWARD OR BACKWARD SLASHES IN THE PATH NAME

Question:

How are the forward or backward slashes in the path name handled by the TeamConnection client?

Answer:

Inside the TeamConnection server only the forward slashes are used (for normalization purposes), and the backward slashes are converted into forward slashes.

However, the TeamConnection client will treat the path name according to the target platform, and will use forward slashes for Unix and backward slashes for OS/2 and Windows.

That is, you can create from OS/2 the file tmp\file.txt and it will be stored as tmp/file.txt (notice the change from backward to forward slashes):

- If you extract the file into an OS/2 directory, then the file is extracted as tmp\file.txt (notice the use of backward slashes).
- If you extract the file into a Unix directory, then the file is extracted as tmp/file.txt (notice the use of forward slashes).

WHY ARE BACKWARD SLASHES IGNORED IN QUERIES FOR PATH NAMES

Question:

I use the Intel clients for TeamConnection. In the Filter dialog for Parts, when I specify the path names with backward slashes (which are the appropriate ones to use for Intel platforms), the result of the query always contains 0 elements. Why?

Answer:

As explained in "Use of Forward or Backward Slashes in the Path Name" on page 29, in order to get a normalized representation of the path names inside the family database, all the backward slashes in the path names are converted to forward slashes. Thus, when doing a query that involves the path name, you need to use forward slashes.

SPECIFYING LONG FILE NAMES WITH IMBEDDED SPACES

Question:

Can I create, checkout, and checkin a file in TeamConnection that has a long name with spaces in it, such as "This is a long file name.txt"?

Answer.

You need to enclose the file name between double quotes. This is done from both the GUI and the line commands, such as:

```
teamc part -checkout "This is a long file name.txt"
```

Notice that the file name with spaces will be shown as-is by the GUI (without the double quotes).

This is not a restriction from TeamConnection, rather, it is from the operating system. For example, if you want to use notepad in Windows 32-bit in the command line for such a file, you would need to do:

```
notepad "This is a long file name.txt"
```


If you forget to enter the double quotes when using the GUI or the line commands, you may see the following error message repeated many times, one for each “token” separated by spaces in the long file name (6 in our example):

```
0010-860 The action cannot be completed. Unable to read the file mode or
attributes of the file specified with the following error.
The file or directory specified cannot be found.
Verify that the filesystem running correctly.
```

SEARCHING FOR % AND _ WHEN USING THE "LIKE" SQL OPERAND

Question:

I need to look for object names that have % (percent) and _ (underscore) characters in them. However, these characters are used as wildcards with the SQL “like” operand. Thus, the results of my query are not correct. For example, I need to look for the files whose names have the '_en' string. I tried the following search operand and I get all the file names that have the string 'en', including those which do not actually have the '_' in them.

```
baseName like '%_en%'
```

How could I specify that % and _ should not be treated as wildcards?

Answer.

Normally, with the SQL “like” operand, the % (percent) character is a wildcard for zero or more characters, and the _ (underscore) character is a wildcard for only one character.

There are 2 ways to search for % or _ and the easiest to use is shown first:

1. Using the “escape” function.

In order to search for % or _, you may use the “escape” function.

Using the above example, you will need to replace the '_' (underscore) with '_' (back slash and underscore) in the “like” command and specify with the “escape” function that the back slash is the escape character:

```
baseName like '%\_en%' escape '\'
```

2. Using the “translate” function.

In order to search for % or _, you may use the “translate” function. First you need to find a string that does not appear in the set of names that you try to find.

Using the above example, you will need to replace the '_' (underscore) with '#' (number sign) in the “like” command and specify with the “translate” function that the number sign represents the underscore:

```
translate(baseName,'#','_') like '%#en%'
```

FILTER DIALOGS ARE AIMED TO NOVICE USERS WHO USE BASIC SQL COMMANDS

Question:

Why do the filter dialogs lack the OR SQL operand? Why cannot I prepare a subselect directly from the filter dialogs?

Answer:

The filter dialogs in the GUI help users to prepare an SQL query that is used to obtain the desired subset of objects to be shown in a window. Under the covers, the filter dialog help the user to prepare the SQL statement to be used in the "where" clause of the "teamc Report" command.

These filter dialogs are intended for basic SQL items and complex SQL functions are not handled:

- Statements are concatenated with the SQL AND

When the user selects two or more statements, such as "releaseName like 'ibm%'" and "dropDate is null", these statements are concatenated with the SQL AND operator:

```
"releaseName like 'ibm%' and dropDate is null"
```

That is, the filter dialogs do not have an explicit choice for AND or for OR.

- Only one "like" can be used.

The filter dialog does not allow multiple "like" statements.

- No subselects (subqueries) can be formed.

The filter dialog does not allow the formation of subselects or subqueries.

In these cases the user may want to prepare the basic query with the help of the filter dialog and then manually change the resulting query.

DIFFERENCE BETWEEN “CHOICES” AND “IMPORT” BUTTONS

Question:

What is the difference between the “choices” and “import” buttons?

Answer:

- Choices button.

Most of the fields in the TeamConnection GUI that have a box titled “choices” get the list of available choices from the Config table in the database, which is populated by the TeamConnection Administration utilities (TCADMIN) from the config.ld file. The TeamConnection family administrator is able to modify this file.

Note: The exception is the “choices” button for the “Part Name” in the “Parts Create” dialog. This button retrieves the files in the directory specified by the “source directory” field.

- Import button.

The “import” button allows the user to get the corresponding data for a field from the selected entry from another TeamConnection window.

For example, if I select “release-A” from the Release window, and then do a “Driver Create” from the “Drivers” window, I will see that there is a “Release” field in the “Driver Create” dialog. If I place the cursor in this “Release” field and press the “import” button, the “release-A” string will be placed in the field. I will not have to type it by hand! That is, the value for the release field is “imported” from the item selected in the release window.

The import button is at the bottom of the dialog and it “knows” which field is being used by means of tracking the position of the cursor. Thus, if the cursor is in field-A, then the import button will be enabled if it is possible to import something in that field.

Also, we added the facility to “remember” the last 10 instances of an object (such as releases) that the user has specified, and thus, when the user is in the Open Defects dialog, the release field becomes a drop down field where the user can select any of the last 10 releases used before.

MISSING FILES FOR THE CHOICES BUTTON

Question:

When I use the Choices button in the “File Create” dialog, sometimes I see only a subset of the files that are listed in the directory. Why?

Answer:

- Ensure that the directory specified in the “Source directory” field the Create Parts dialog is indeed the directory where the actual files reside.

The default, the value of the field “Relative directory” in the Environment page of the Settings notebook, is the value that is placed in the “Source directory” field of the mentioned dialog.

- If you are using the Unix GUI, the files in a mounted directory whose NFS mount point has become stale are not shown.
- If you are using Netware or another system as a file server, the files that do not have an archive bit set will not be shown. You can use the command “attrib -a” to set the archive bit.

CAN I CREATE A USER EXIT IN MY TEAMCONNECTION CLIENT?

Question:

Can I create a user exit in my TeamConnection client?

Answer:

The usual reason for requesting client side user exits is to overcome the limitation of TeamConnection in which no value may be changed in a server side user exit. TeamConnection cannot permit changes in user exits because data validation is distributed throughout each command, beginning on the client side (before user exit 0) and not completing until right before the action is committed in the database (the last command before user exit 2 in most cases).

The next question is usually a request for a client side user exit. While there are no client side user exits in TeamConnection, it is possible to simulate client side exits.

A client user exit may be simulated by:

1. Creating a wrapper shell script or executable of the same name as the TeamConnection command, before the actual TeamConnection command in the PATH.
2. Processing the input parameters in this wrapper.
3. If necessary issuing TeamConnection commands, such as teamc report, to get more information.
4. Updating the parameters as appropriate.
5. Calling the TeamConnection command using an explicit PATH.

If you are concerned about users changing your tool, thereby defeating your goal, there are a few more things you can add to maintain some control:

1. Put your wrapper in a directory where the user cannot update the file or mount the commands using a read-only file system.
2. Use a rexec command to the server in order to set a semaphore before the TeamConnection command is issued.
3. In a user exit in the TeamConnection family server, verify that the semaphore is set.
4. After the TeamConnection command is complete, release the semaphore before exiting the wrapper.

In most cases, you will likely use a wrapper to aid the user and all of this extra effort and overhead will not be necessary; however, the workaround is possible if needed.

Note: If you are wrapping a TeamConnection command, it will be necessary to set signal handlers to insure that the wrapper runs to completion even if the user interrupts the TeamConnection command.

STRANGE ERROR MESSAGES CAUSED BY WRONG MESSAGE CATALOG

Question:

Why are we seeing some messages (informational, warnings, and errors) from TeamConnection that seem not to be related to the task at hand?

Answer:

If your family is experiencing strange messages (informational, warnings, and errors), they are likely to be the result of an incompatibility between the message catalogs and the executables for TeamConnection. This could happen if you have an older version of TeamConnection installed in your computer and the NLSPATH variable is pointing to the older one.

In earlier versions of TeamConnection we provided the message catalogs in `/usr/lib/nls/msg/%L/%N` (where %L is the language defined by \$LANG, and %N is a variable that represents the actual message catalog name). However, in recent versions we are storing the message catalogs in `$TC_HOME/nls/msg/%L/%N`.

Thus, the problem might be that the NLSPATH variable points simply to `/usr/lib/nls/%L/%N`, instead of `$TC_HOME/nls/msg/%L/%N`.

Another possibility is that the Unix user is using LANG=En_US (code page 850 which is compatible with the OS/2 multilingual code page) but the actual message catalog is stored under LANG=en_US (ISO code page 819, which is different than the 850 code page).

You have these choices to overcome the problem:

- Change the LANG variable from en_US to En_US, logoff and login again to refresh the environment.
- Add the following to the beginning of the NLSPATH:
`NLSPATH=$TC_HOME/nls/msg/en_US/%N:$NLSPATH`
- If any of the above fixes the problem, then you may need to reinstall the product to ensure that you are using the correct message catalog.
- Ensure that both the TeamConnection client and TeamConnection family server code that are installed in the same host are at the same level.

USING THE TC_CASE ENVIRONMENT VARIABLE

Question:

How do I use the TC_CASE environment variable?

Answer:

CMVC, the predecessor product to TeamConnection, was born in the Unix environment and therefore, it is sensitive to lower and upper case file names. TeamConnection inherited this characteristic.

When the Intel workstation clients for TeamConnection were developed (OS/2 and Windows, which are NOT case sensitive), an environment variable was added called TC_CASE. It can be set in the settings notebook. It allows customers to control the mapping of file names to the client portion of TeamConnection.

- If the value is “ignore”, then the TeamConnection client will NOT change the file names. This is the default.

This is the only option that allows you to handle file names in mixed mode.
- If the value is “upper”, then the TeamConnection client will change the file names to upper case. Thus, the TeamConnection family server will see the request that comes from the TeamConnection client as being with file names in uppercase.
- If the value is “lower”, then the TeamConnection client will change the file names to lower case. Thus, the TeamConnection family server will see the request that comes from the TeamConnection client as being with file names in lowercase.

This variable does NOT change the response that comes back from the TeamConnection family server.

CAN I HAVE MULTIPLE INI FILES FOR DIFFERENT FAMILIES?

Question:

I constantly work with a couple of TeamConnection families and the queries and values that are use in one family, are different from the other family. Is it possible to have one INI for one family, and another INI for another family?

Answer:

In Version 3 we added the capability of specifying which settings file you want to use. This is specified in the "Preference file ..." item, above the "Settings" item in the Windows pull down menu from the menu bar.

Using this, you can have one INI file for one family and another INI file for another. Simply by changing which INI file you use, you can change the queries and the default values.

IN SOME PARTS QUERIES, IT SEEMS THAT THE WRONG WORKAREA IS USED

Question:

When I use some queries for parts (PartView, BuildView, etc.), I am getting parts that belong to a specific workarea, even though I have not specified any workarea in the current dialog. Why?

Answer:

The Workarea field in the Settings notebook works like an environment variable and if you enter a value in this field, then it will be used in ANY of the commands in which you can specify a workarea.

To avoid the problem, it is recommended that you should avoid entering a value in this field (or to give a value to TC_WORKAREA for the line commands). Thus, whenever you want to specify a workarea, just add it to the corresponding dialog (but not to the Settings).

CR (CARRIAGE RETURN) AND LF (LINE FEED)

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

In TeamConnection, the model of what-you-see-is-what-you-get is used. This means that if a user creates a file in TeamConnection, regardless of the platform of the server, then TeamConnection will NOT do any conversion of LF or CRLF on that file. There are choices in the -extract action to allow for more fine tuning of these on-the-fly-conversions. For example, a Unix user may wish to extract with only LF a file that was stored originally from OS/2 that has CRLF.

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

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

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

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

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

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

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

Using the tool fixcrlf to fix CR-LF conversion problems

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

This tool can be obtained from the TeamConnection ftp site:

1. ftp ftp.software.ibm.com
2. login anonymous
3. cd ps/products/teamconnection/tools
4. ascii
5. get fixcrf.txt
6. binary
7. Download the corresponding file for your platform:
 - AIX: aixfixcrf
 - HP-UX: hpfixcrf
 - Solaris: fixcrf.solaris
 - OS/2: fixcrf2.exe
 - Windows: fixcrfw.exe
8. quit

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

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

HOW TO ENABLE/DISABLE THE TIME STAMP IN THE GUI TEAMCV3.LOG

Question:

In order to correlate the actions shown in the audit.log of the family server with the actions performed in the client, I would like to use the time stamps of these actions (the audit.log uses time stamps). Is there a way to enable/disable time stamps in the GUI teamcv3.log?

Answer:

Starting with Version 3.0.1, the teamcv3.log generated by the GUI will use by default a time stamp for each action. To disable these time stamps, set the following environment variable before starting the GUI:

```
Unix:  export TC_NOLOGTIME=on
Intel:  set TC_NOLOGTIME=on
```


UNIX FREQUENTLY ASKED QUESTIONS

In this section several frequently asked questions are presented which are specific to the TeamConnection client for Unix.

SAMPLE PROFILE FOR A TEAMCONNECTION UNIX USER

Question:

Is there a ready-to-use sample profile for a TeamConnection Unix user?

Answer:

We provide an excellent sample profile for a TeamConnection Unix user. We strongly recommend that every new user, regardless of their expertise, uses this sample profile, or at least uses it as the basis for the profile. We are constantly adding new environment variables, hints, and tips.

Furthermore, each sample profile has variables that are dependent on the operating system. They are commented out. It is the responsibility of the TeamConnection user to review the entire profile and uncomment those variables that are appropriate.

The user should make a backup copy of their current profile before getting the sample profile. For example:

```
cd $HOME
mv .profile .profile.original
cp $TC_HOME/install/$LANG/profile.user .profile
```

CDE: MY LIBPATH STATEMENT IN .PROFILE IS NOT EXPORTED, WHY?

Question:

When I login into the Common Desktop Environment, CDE, and open a dtterm window, I notice that the LIBPATH is not set. The strange thing is that the LIBPATH is properly specified in the .profile and the .dtprofile has the statement "DTSOURCEPROFILE=true" which starts the .profile.

This causes some problems with TeamConnection. When I issue a command such as "teamc", I get the following errors:

```
exec(): 0509-036 Cannot load program teamc because of the following errors:
0509-022 Cannot load library libfhccmnc.a
```

Answer:

The default behavior for the CDE Session manager is to invoke dtterm with the +ls flag. This starts a normal (non-login) shell, that is, the .profile is not executed. Even if your .dtprofile has the statement DTSOURCEPROFILE=true, the LIBPATH in the .profile is not exported.

In order to force the execution of the .profile when the dtterm window is opened, and to force the export of LIBPATH, you need to do the following:

1. Edit your \$HOME/.Xdefaults and add the following entry to indicate that the login profile must be executed (the same as the -ls flag for dtterm)

```
Dtterm*loginShell: true
```

2. After you change the .Xdefaults file, you must exit from all your windows.
3. Log out from the CDE environment.
4. Log in to the CDE environment.
5. Open a new dtterm window.
6. Verify that the LIBPATH is now exported, by doing the following:

```
$ echo $LIBPATH
```

FILES TO BACKUP FOR THE TEAMCONNECTION GUI

Question:

In case I have to reset my TeamConnection GUI from scratch, either on my machine or on another machine, which files should I backup?

Answer:

In case of a disk crash or similar situation in which your local files are damaged, the following files need to be backed up regularly in order to reset your TeamConnection GUI:

- \$HOME/teamcv3x.ini, which contains the values for the Settings page.
- Any other preference (INI) files which contain values for other families.

Although you can store these preference files in other directories, it is recommended that you store these alternative preference files in the same directory as teamcv3x.ini, \$HOME.

- \$HOME/Teamcgui, which contains your customized Motif resources.

- \$HOME/tcadmin.ini, which contains the settings for the TeamConnection Administration tool (TCADMIN). This is only relevant if you are a family administrator.
- \$HOME/tcmrgprf.txt, which contains the settings for the TeamConnection VisualMerge tool (tcmerge).

For more information on these files, see “Difference Between INI and Teamcgui Resource Files.”

DIFFERENCE BETWEEN INI AND TEAMCGUI RESOURCE FILES

Question:

What is the difference between the local \$HOME/teamcv3x.ini file, the system wide \$TC_HOME/nls/msg/\$LANG/Teamcgui resource file, and the local \$HOME/Teamcgui resource file?

Answer:

The following files are used by the TeamConnection GUI to find out the values to use for specific Motif resources:

- The local INI file (\$HOME/teamcv3x.ini) contains the operational settings used by the GUI, such as the contents of the Settings notebook, the tasks lists, the list of the recently used objects, the geometries and layout for specific windows, etc.

It is strongly recommended that the users do not manually modify this file. Let the GUI do it.

If this file does not exist when the GUI is invoked, it will be created using the existing environment variables.

If you need to reset your INI file, you can get the default one as follows:

```
cp $TC_HOME/nls/cfg/$LANG/teamcv3x.ini $HOME/.
```

- The system wide resource file contains the text for all the strings displayed in the GUI. In a sense, this is the message catalog for the GUI. It is located in \$TC_HOME/nls/msg/\$LANG/Teamcgui.

It is strongly recommended that users do not modify this file.

It is also recommended that the users do NOT copy this message catalog into their home directories, because it will interfere with the system-wide message catalog. This could cause problems when a different version or fixpak of TeamConnection is installed on the system. Instead, the user can modify the local Teamcgui resource file. See the item below.

- The `$HOME/Teamcgui` file is the local resource file used to control X and Motif resources, such as fonts.

If you are not happy with the default fonts for the TeamConnection GUI, you can edit this resource file. On the other hand, if you are satisfied with the default values, there is no need to have this local resource file.

Use the following sample local resource file in case you want to modify the fonts or colors:

```
cp $TC_HOME/nls/cfg/$LANG/Teamcgui.user $HOME/Teamcgui
```

For more information on this local resource file, see “Setting the Fonts for the TeamConnection GUI.”

For details on the search path (order of precedence) for the Motif resources, see “The Search Path for the Motif Resources” on page 46.

SETTING THE FONTS FOR THE TEAMCONNECTION GUI

Question:

How do you set the fonts for the TeamConnection GUI?

Answer:

As shown in “The Search Path for the Motif Resources” on page 46, the file `$HOME/Teamcgui` is searched last for the Motif resources. Therefore, those resources defined in this file are the ones that will be used for the TeamConnection GUI.

The main resources that you can control from this file are the font type, font size and background/foreground colors (for more information on configuring the colors, see “Customizing the Colors of the GUI” on page 45.) This is specially important to users of DBCS (Double Byte Character Set) locales. If the proper DBCS font is not specified, then the DBCS characters will not be shown.

Copy the sample Teamcgui resource file into your `$HOME` directory:

```
cp $TC_HOME/nls/cfg/$LANG/Teamcgui.user $HOME/Teamcgui
chmod u+w $HOME/Teamcgui
```

Unix provides the “xlsfonts” tool to list all of the fonts available on your X-Server. You may have to go looking for it, as it is often not on the `PATH` provided when your user account is created.

You can also use the Custom tool to choose from the available fonts and colors. It allows you to preview your selections, and save them to a sample file.

CUSTOMIZING THE COLORS OF THE GUI

Question:

How do you customize the colors of the GUI?

Answer:

There are several options for customizing the colors of the TeamConnection GUI:

- Use the CDE Style Manager (fourth icon from the right in the CDE console) and select the Color palette tool (the first one on the left).

The user should start the `teamcgui` and bring up an information window. This can be done by selecting “Settings” from the “Windows” menu, and pressing the “Test Connection to Server” button. The information window sometimes have poor contrast, depending on your configuration.

The user can quickly explore the options available from the Color palette and see how it affects the contrast. For example, using the default color scheme for area 1, we selected NorthernSky, and the contrast was good.

- Invoke `teamcgui` as follows:

```
teamcgui -bg BackgroundColor -fg ForegroundColor &
```

The following example is ugly. However, it illustrates how the user can explore to get colors that provide better contrast:

```
teamcgui -bg black -fg white &
```

- To set the foreground and background colors for the `teamcgui` application, add the following lines to either the `Teamcgui` or `.Xdefaults` file in the `$HOME` directory.

```
Teamcgui*background: black  
Teamcgui*foreground: white
```

Choose any available colors if black and white are unacceptable:

- A good way to pick colors is to use the Unix utility called “`custom`”. It lets you choose color from a list and preview what they will look like.
- You can find the list of available colors in the file:

```
/usr/lib/X11/rgb.txt
```

THE SEARCH PATH FOR THE MOTIF RESOURCES

Question:

What is the search path (order of precedence) for the Motif resources used by teamcgui?

Answer:

There are a couple of common problems related to the Motif resources used by teamcgui (geometry, fonts, text labels, etc.). Some of these problems are caused by the way these Motif resources are searched for in the system, that is, the order of precedence in which these resources are found can cause some problems.

- If the menu bar or the pull downs show labels that begin with STR, instead of more intelligible labels, then the overall system-wide Teamcgui resource file has not been located.

To fix this problem, ensure that the following variables are properly enabled in your .profile:

```
# The following variable indicates the directory where the system-wide
# Motif resource file (Teamcgui) is located:
export XAPPLRESDIR=$TC_HOME/nls/msg/$LANG
```

```
# The following variable indicates the full path name where the local
# Motif resource file (Teamcgui) is located. These resources will
# override the ones from the system-wide file. Also, this is the place
# to customize the fonts.
export XENVIRONMENT=$HOME/Teamcgui
```

- Sometimes when users change their local resource file (\$HOME/Teamcgui), they do not get the expected results. This is a common problem when handling X motif resources.

It is necessary to understand the search path for the resources to isolate the problem. This is the sequence used by the X Window System to find the resource files and merge them into the resources data base. The order is extremely important, because the files that are read last are the ones that take effect. That is, the last resources to be stored in the resources data base are the winners!

The following sequence is the complete one. The file name, "Teamcgui", is used here for clarification, although you will not necessarily find the file there.

1. Read the resources from one of the following files:

```
    /usr/lib/X11/$LANG/app-defaults/Teamcgui
OR  /usr/lib/X11/app-defaults/Teamcgui
```

2. Merge the resources found in the previous step with the ones in the following files (in this particular order):


```
$XUSERFILESEARCHPATH/Teamcgui
OR $XAPPLRESDIR/$LANG/Teamcgui
OR $XAPPLRESDIR/Teamcgui
OR $HOME/$LANG/Teamcgui
OR $HOME/Teamcgui
```

3. Merge those resources with the ones:

```
defined by xrdb (this is the RESOURCE_MANAGER)
OR in $HOME/.Xdefaults
```

4. Merge those resources with the ones in the following file:

```
$XENVIRONMENT (such as /home/userId/Teamcgui)
OR $HOME/.Xdefaults-hostname
```

5. Merge those resources with the ones specified in the teamcgui line command, such as:

```
teamcgui -xrm '*fontList: ergo11.isol'
```

COMMON USAGE ERRORS WITH MOTIF RESOURCES

Question:

What are some common errors when using Motif resources?

Answer:

Here are some difficulties users have encountered:

- The user changes one line, ***fontList: rom22**, in the local `$HOME/Teamcgui` file and expects the fonts in all parts of the GUI to be changed.

General resource settings do not override specific setting. It is the other way around. For example, ***XmText*fontList: erg14** will override ***fontList: rom22**.

- Changing the value of a resource has no effect. It may be possible that the resource is defined more than once and the second instance of the resource in the file will be the winner.
- Users expect that commenting out a resource in `$HOME/Teamcgui` will set it to blank. However, if the resource is set to some value in the system wide Teamcgui file, then this value will take effect. The system wide Teamcgui file is there to supply a value to resources in case no values are supplied in the `$HOME/Teamcgui` file.
- Users get the following warning on stdout:

```
X Toolkit Warning: Cannot convert string erg14 to type FontList
```

Note the 2 blanks after `erg14`. X Windows does not remove trailing blanks. An editor like `vi` will save the file with trailing blanks. This affects all resources, and most of them are not within the scope of `$HOME/Teamcgui`.

- Users get the following error message in the GUI when trying to print:

sh: lpr: not found.

Either the user forgot to set the complete path to the command when setting the print command, or the command is not in the path. If the search path was not set properly when the GUI was invoked, the GUI will not be able to find the command.

- Specifying ***fontList: variable** in the `.Xdefaults` file.

This setting overrides any setting done in the `$HOME/Teamcgui` file. Moreover, it applies to all applications since a wildcard, the asterisk, is specified. It is recommended that there should be no resource specification beginning with a "*" in the `.Xdefaults` file.

- Specifying proportional fonts instead of monospaced.

If proportional fonts are used, information in the TeamConnection windows will not align nicely. Only monospaced fonts should be used in order to align the columns of data.

In addition, the font called **variable** has a bug, the underscore is displayed as a right arrow (`->`).

HINTS FOR ISOLATING MOTIF RESOURCE PROBLEMS

Question:

Are there any hints on how to isolate Motif resource problems?

Answer:

If you suspect that the TeamConnection GUI has a problem that might be due to a resource specification problem, here are some sets of hints to isolate the problem.

- Set 1: Determine which resource file is causing the problem.
 1. Rename `$HOME/.Xdefaults`. Exit `teamcgui` and restart.
 2. Rename `$HOME/.kshrc`. Exit `teamcgui` and restart.
 3. Rename `$HOME/Teamcgui`. Exit `teamcgui` and restart.
 4. Rename `.mwmrc` or the name of resource file for your window manager. Restart the window manager and try again.
 5. Rename `.xinitrc`. Kill the X session and try again.
 6. Echo the `XENVIRONMENT`, `XAPPLRESDIR`, and `XUSERFILESEARCHPATH` environment variables and note their values. Unset any of these you may find that have values.

If the problem is introduced by one of these files, try to isolate which resource is causing the problem by commenting them out one by one. As a last resort, reboot the workstation. Do not make early conclusions. Removing a resource sometimes drastically changes memory usage and the relationship with other resources. It may not be the one in error. It may just change the environment enough so the problem goes away.

- Set 2: Test new values.
 1. Stop the TeamConnection GUI.
 2. Rename \$HOME/.Xdefaults, \$HOME/Teamcgui (or change XENVIRONMENT, XAPPLRESDIR, and XUSERFILESEARCHPATH).
 3. xrdp -remove -all
 4. xrdp -load Teamcgui.new (try some values defined in a temporary file, Teamcgui.new)
 5. Restart the TeamConnection GUI.

WHICH RESOURCE IS USED: DOUBLECLICKTIME OR MULTICLICKTIME?

Question:

What does the TeamConnection GUI use to set the double click time? Is it doubleClickTime or multiClickTime?

Apparently teamcgui is ignoring the attributes I have in my \$HOME/.Xdefaults and/or \$HOME/Teamcgui files. Is there a TeamConnection resource that I can set?

Answer:

TeamConnection does not set the resource for double click (the interval in which an action is valid, between the time of the first click and the second click). The GUI uses the Motif default value for multiClickTime, 500 or 1/2 sec.

If you want to set multiClickTime to 2 seconds, set the following variable to a value of 2,000 milliseconds or 2 seconds, in the \$HOME/Teamcgui resource file:

```
*multiClickTime: 2000
```

CONNECTING TO THE X SERVER.

Question:

What do you do if you get the following error messages when you issue "teamcgui"?

2059-023 Cannot make a connection to the X server.
If the X server is not running, run the xinit command.
If the X server is running, check the specified display number.

or

Error: Can't Open display

Answer:

This problem occurs when you are running TeamConnection remotely. That is when you are using the telnet or rlogin commands to display the TeamConnection windows on a different machine than it is running on. From the X server, before you enter the telnet or rlogin commands, enter the following to allow the client access:

```
xhost + server_hostname
```

Where "server_hostname" is the name of the host where the actual TeamConnection client will be running, even though the window will be displayed in another host.

After you have logged in remotely to "server_hostname", you need to enter the following in order to set the display to your local machine:

```
export DISPLAY=client_hostname:0.0
```

Where "client_hostname" is the actual name of the hostname on which you want to see the TeamConnection windows (the X client).

SETTING THE DISPLAY TO GET THE BEST X-WINDOWS PERFORMANCE

Question:

How do I set the DISPLAY variable to get the best possible X-Windows performance?

Answer:

X-Windows uses shared memory for communications between the X-Server (the application that manages your display) and X-Client (the TeamConnection GUI) on the same workstation. In this case, the DISPLAY variable is set to the following (notice the absence of a hostname):

```
export DISPLAY=:0.0
```

Note: This only works when the user that started the X-Server is the same user that is starting the X-Client.

If you are communicating between workstations, you must specify a hostname. X-Windows uses a socket, which is much slower than shared memory.

```
export DISPLAY=hostname:0.0
```

USING TELNET TO ACCESS THE TEAMCONNECTION GUI

Question:

Are there any special things to consider when using telnet to access the TeamConnection GUI.

Answer:

As with any X application, a user may telnet from a display from one system into another system. In theory, due to the Client/Server architecture of X Window System, there should be no problem. The TeamConnection Client, "teamcgui", executable is running on one system and the windows are displayed on the X Server of another system. Problems that are caused by Motif and/or X server are beyond the control of our application.

Chances are that the default fonts are not available on the display. The user should change all instances of the fonts in the local resource file \$HOME/Teamcgui resource file to a font name that exists on the local workstation. For example, the font "ergo11" is available in all CDE systems.

For low resolution monitors, the user must choose fonts that are small enough to show the entire dialog box. With the use of scrollable entry fields in the dialogs, this may not be a problem.

THE GUIs DO NOT APPEAR (THERE ARE NO ERRORS)

Question:

In Unix, I start the TeamConnection Family Administrator GUI (tcadmin), the TeamConnection client GUI (teamcgui), and the Visual Merge tool (tcmerge), but the windows do not appear. I do not even get any error messages. It seems to be hung up.

Answer:

One very likely cause is that the DISPLAY environment variable has not been properly established, and thus, the application is up and running, but it is being displayed on another terminal!

To fix the problem, do the following:

1. Verify that the application is indeed up and running, by executing the command:

```
ps -ef | grep tcadmin
or
ps -ef | grep teamcgui
or
ps -ef | grep tcmerge
```

2. If there is a process, you can terminate it by issuing the command:

```
kill -15 <pid>
```

Where “<pid>” is the process ID identified in the output of the ps command.

3. Ensure that your DISPLAY variable is setup to the proper X server.

- From a local host, you can use shared memory, which will be faster:

```
Using Korn shell:
export DISPLAY=:0.0
```

```
Using Bourne shell:
DISPLAY=:0.0
export $DISPLAY
```

In Bourne shell you cannot combine the definition and the export in the same statement. The rest of this document will show only the syntax for the Korn shell.

- From remote hosts, if you have logged into a host A, then telnet to another host B, you need to use sockets (which are slower than shared memory):

```
export DISPLAY=hostName:0.0
```

It is important to note that the terminal where the window will be actually displayed (host A) needs to allow the use of the X server, by doing the following in host A:

```
xhost +
```

4. Verify that you can correctly display an X Windows application. You can try the following harmless application that will display a clock in a window (as a background process):

```
xclock &
```

Note: If you do not see the xclock window, then STOP! This means that your DISPLAY variable is not setup correctly, and you will NOT see the window from the tcadmin tool either. Return to the previous steps to fix the DISPLAY variable, and re-issue the xhosts command to properly display an X Windows application. If you are still having problems, consult your Unix manuals or local help desk.

5. Finally, invoke the TeamConnection GUI application:

- Client GUI and merge GUI:

By default, the TeamConnection Client GUI and the Visual Merge tool always run in the foreground, which means that if you invoke them as “teamcgui” and “tcmerge”, the GUI windows will appear but you will not be able to do anything else in the X window (dtterm) from which you started them.

If you want dtterm window to remain active, you need to invoke the GUI in the background, as follows:

```
teamcgui &  
tcmerge &
```

- Family Administration GUI:

The Family Administration GUI runs in the background by default. Thus, there is no need to specify the “&”. That is “tadmin” will behave the same as “tadmin &”.

USE OF .KSHRC MAY CAUSE WEIRD BEHAVIORS

Question:

I am running the TeamConnection GUI, using the Korn shell. I get really weird problems, specifically when doing a telnet and exporting the DISPLAY back to my workstation. The family is fine, because I do not see these problems from other clients.

Answer:

We have seen cases where a “sophisticated” .kshrc file causes weird problems with the GUI.

If after unsuccessfully trying several ways of overcoming the problem, you may try to rename the .kshrc file to something like “.kshrc.original”, then logoff and login again.

If the problem no longer occurs, it is time to debug the .kshrc file!

A common problem is the redefinition or aliasing of the “cd” command.

WINDOW MANAGERS USED WITH THE TEAMCONNECTION GUI

Question:

Which window managers can be used with the TeamConnection GUI?

Answer:

Officially, the TeamConnection GUI works only on top of the Common Desktop Environment (CDE) which is available in AIX 4, HP-UX 10 and Solaris 2.5. This is the only supported window manager.

We have not actively tested other window managers.

LINE COMMAND ARGUMENTS FOR THE STANDALONE TEAMCONNECTION GUI

Question:

What are the line command arguments for the TeamConnection GUI?

Answer:

These are the line command arguments supported for teamcgui:

```
Usage: teamcgui [-bg BackgroundColor]
               [-fg ForegroundColor]
               [-display Host]{:Number}
               [-geometry {WidthxHeight|{+|-}xoff{+|-}yoff}]
               [-iconic]
```

Figure 2. teamcgui line command arguments.

CUSTOMIZATION HINTS FOR THE CDE ENVIRONMENT

How to replace Editor icon with Terminal icon in CDE console.

Some users use the Terminal icon more frequently than the Editor icon from the CDE console. However, the Editor icon is more accessible than the Terminal icon, which is located inside a "drawer" which the user needs to open and close.

To customize the CDE console to replace the Editor icon for the Terminal icon, do the following:

1. Expand the Personal Applications drawer that has the list of the applications that are frequently used by the user.

To expand this drawer, move the cursor to the up-arrow icon above the 4th icon from the left (this is the default location), between the drawer icon and the mail icon. By default this 4th icon shows the Editor icon

2. Use the arrows to move in the context menu to select Terminal.
3. Click on the right mouse button in "Copy to Main Panel".
4. Click again on the arrow icon to close this drawer.

How to have all window icons inside a single icon box.

In the default style for CDE, all the windows that are opened do not have an icon on the desktop; only when you minimize (iconify) these windows, the corresponding icons are shown on the left side of the screen. However, it is rather cumbersome to switch quickly to another open window because you have to click on many of the open windows to find it.

One recommendation is to enable the icon box, which is another window which will show the icons for ALL the windows, regardless if they are opened or minimized. Thus, it is very easy to switch to another opened window by simply clicking on its associated icon from the icon box.

You can place the icon box whenever you like; on the lower right corner seems to be a good place.

1. Open the Style Manager folder, by clicking on the 4th row from the right between the printer and the drawer.
2. Click on the 3rd item from the left, called "Window".
3. Go to "Windows Icons" group and enable "Use Icon Box".
4. Click OK on the notice that the workspace manager will restart. After the restart, you will see the new window called "Icons".
5. Close the Style Manager folder, if you want.

How to change the title name of a dtterm window

When opening a dtterm window the default icon name and window title will be "dtterm". The problem is that if you open many such windows, all of them are called "dtterm".

You may use the following command to open a dtterm window with a specific title, in that way you can differentiate between the several dtterm windows that you may have:

```
dtterm -name Name
```

How to customize the context menu from the desktop.

You can customize the context menu from the desktop (the menu that is shown when you move the cursor to the plain desktop and press the right mouse button).

One frequent customization is to add entries for different dtterm windows that have their own title. For example, a user may need to access frequently several AIX and HP-UX systems by opening dtterm windows with specific names. This user would like to have a context menu that will have the following customized entries:

```
AIX windows -> tcaix01
                oem-ppc3
```

To make such customization, the user would need to do:

1. Make a backup copy of the file `.dt/dtwmrc`.
2. Edit the file `.dt/dtwmrc` as follows:
 - In the section of the top Menu, add the desired entry that will cascade into another menu:

```
Menu DtRootMenu
{
    "Workspace Menu"  f.title
    no-label          f.separator
    "AIX Windows"     f.menu "aixConnect"
    "HP-UX Windows"   f.menu "hpuxConnect"
    ...
}
```

- Add the cascading menu for the AIX windows:

```
# AIX Connections menu

menu "aixConnect"
{
    "tcaix01"          f.exec "dtterm -name tcaix01 -e telnet tcaix01 &"
    "oem-ppc3"         f.exec "dtterm -name oem-ppc3 -e telnet oem-ppc3 &"
}
```

3. Restart the workspace manager.

Customizing the Motif Environment for Left-handed Users

To customize the Motif environment for left-handed users, do the following:

- If running the CDE, use the “Style Manager”, select the “Mouse” and then select the desired mode: right hand (default) or left hand.
- If not using the CDE, add the following entries in your `.profile`:

```
xmodmap -pp  
xmodmap -e 'pointer=4 3 2 1'
```


INTEL FREQUENTLY ASKED QUESTIONS

FILES TO BACKUP FROM THE TEAMCONNECTION GUI

Question:

In case I have to reset my TeamConnection GUI, which files should I backup?

Answer:

The following files need to be backed up regularly in order to reset your TeamConnection GUI in case of a disk crash or similar situation in which your local files are damaged:

- %HOME%\teamcv3.ini, which contains the values for the Settings page.

Where %HOME% is the location of the system-wide INI files, such as C:\OS2 or C:\WINDOWS.

- Other preference (INI) files which contain values for other families. Although you can store these alternative preference files in any directory, it is recommended that you store them in the same directory as teamcv3.ini.
- %HOME%\tcadmin.ini, which contains the settings for the TeamConnection Administration tool (TCADMIN). This is only relevant if you are a family administrator.

Other possible locations for the tcadmin.ini are:

- %HOME_DRIVE%:\%HOME_PATH%\tcadmin.ini
 - Java's home directory.
- %HOME%\tcmrgprf.txt, which contains the user preferences for the Visual Merge tool.

IMPORTANT WARNINGS WITH WINDOWS 32-BIT LOGINS

In Windows 32-bit, when the TeamConnection client queries the system to find out the system login, the value that was used when the system login was created is the actual value that is passed from the system to the client. Keep in mind that this value may not have the same case that was entered during the login process.

For example, a user can login as “administrator” (all in lower case), but the actual value of “Administrator” (notice the capital A) is passed to TeamConnection.

This has very important implications for the host lists, because there must be a host list entry for “Administrator” in order for the client to access the server. A host list entry of “administrator” will not be correct.

RESTRICTION ON FILE EXTRACTS INTO OS/2 FAT DRIVES

Question:

Are there any restrictions on file extracts into OS/2 FAT drives?

Answer:

Files from the TeamConnection family server that do not conform to the 8.3 naming convention (file name longer than 8 characters or file extensions longer than 3 characters) cannot be extracted to a FAT drive on OS/2.

Windows 95 and Windows NT can handle long file names in FAT drives.

CHANGING THE FONTS IN THE TEAMCONNECTION GUI

Question:

How do I change the fonts in the TeamConnection GUI?

Answer:

In the TeamConnection client for Windows, select “Options”, then “Settings”. Select the “GUI” page from the notebook and at the bottom of the page you can specify the font height to be used for information windows and other windows that use fixed fonts.

The font types are Arial for proportional, and Courier for fixed.

In OS/2 it is possible to specify both the type and the height.

PROBLEMS WITH APM DRIVERS WHEN NAMED PIPES ARE SUSPENDED IN OS/2

Question:

After several uninterrupted hours of using the TeamConnection GUI, the lists in the GUI are truncated and the user is not warned. For example, instead of the usual 100 entries in the

defect list, only 10 show up. When using the TeamConnection line commands directly (that is, bypassing the GUI) the problem does not show up. This is a possible short-term workaround to the problem. Is there a long term solution?

Answer:

The PMR, 5X399, was opened against the OS/2 operating system to report a problem with the named pipes used by the TeamConnection GUI in machines that use the Advance Power Management (APM) option.

Here is an easy test to determine if you have this problem:

1. Bring down the TeamConnection GUI.
2. Bring up the TeamConnection GUI.
3. Select "Actions, Defects, Open..."

If the Defect screen appears, cancel it - No Problem.

If TeamConnection GUI dies, you have the problem.

This test is more definitive than getting a list that you think might be too short, but you do not have any way to know for certain.

Note: In most cases, the following easy test confirms the problem. However, if the test does not behave as described here, it does not mean that you do not have the problem.

Summary of the workaround:

A medium-term workaround is to comment out (with the REM keyword in the first three positions of the row) the drivers for the Advance Power Management (APM) from CONFIG.SYS, APM.SYS, and VAPM.SYS. This workaround might not be acceptable to all customers. In that case, the customer needs to contact the support for their hardware to get the latest versions of these drivers, which should have the fix for this problem.

ENABLING THE NETSCAPE NAVIGATOR FOR JAVA ON OS/2

Question:

After installing the Java Development Kit (JDK) for OS/2, the Netscape Navigator web browser is still not enabled for Java. How do I do that?

Answer:

To enable Netscape Navigator for Java after installing the JDK on OS/2, do the following:

1. Click on the "OS/2 Warp" icon on the leftmost position on WarpCenter.
2. Click "Program."
3. Click "Netscape."
4. Click "Java Version Selection for Netscape Navigator."
5. Select "Java 1.1" radio-button, and click OK.
6. Shutdown and reboot.

You can perform the following simple verification to test if Java 1.1 is enabled:

1. Start Netscape.
2. Drop down "File" menu and select "Open File."
3. There are some demo files under \java11\demo\. For example, select
 \java11\demo\animator\example1.html
4. First, a Java 1.0.2 applet will start.
 Check if the 1.0.2 applet runs successfully.
5. Next, click "The 1.1 version" link on the Netscape screen.
 Check if the 1.1 applet runs fine.

APPENDIX A. COPYRIGHTS, TRADEMARKS AND SERVICE MARKS

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

TRADEMARK, REGISTERED TRADEMARK OR SERVICE MARK	COMPANY
NetLS iForLS, iFor	Gradient Technologies, Inc.
HP-UX	Hewlett-Packard Company
Exceed	Hummingbird Communications Ltd.
IBM AIX OS/2 CMVC TeamConnection	IBM Corporation
Lotus Notes	Lotus Corporation
X Window System	Massachusetts Institute of Technology
Microsoft Windows Windows NT Internet Explorer	Microsoft Corporation
Netscape	Netscape Communications Corp.
Netware	Novell Inc.
OSF Motif	Open Software Foundation, Inc.
PowerBuilder	PowerSoft Corporation.
NFS Solaris Java	Sun Microsystems Inc.
Unix	X/Open Co., Ltd.

| CDE | |
+-----+-----+

END OF DOCUMENT