

**Tivoli Data Exchange
Version 1.2.5**

Installation Guide

Copyright 2001, 2002 Tivoli, Inc. All rights reserved.

5/20/02

Tivoli Data Exchange, Tivoli, and the Tivoli logo are registered trademarks or trademarks of Tivoli, Inc. AS/400, AIX, MQSeries, and OS/390 are registered trademarks of IBM Corp. InstallShield is a registered trademark of InstallShield Software Corporation. Win 32 is a registered trademark of Microsoft Corp. UNIX is a registered trademark of The Open Group. Solaris and Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other trademarks are property of their respective owners.

Contents

chapter 1	About This Guide	7
	About Tivoli Data Exchange	7
	Who Should Read This Guide	9
	How This Guide Is Organized	10
	Contacting Customer Support	10
	Where To Look for More Information	11
	Conventions	11
 chapter 2	 Installing Tivoli Data Exchange on	 13
	OS/390	13
	Assumptions	13
	Overview	14
	Planning the Installation	15
	Reviewing System and Software Requirements	15
	Establishing Installation Values	16
	Checking Data Set Size Allocations	19
	Loading the Tivoli Data Exchange Software	20
	Installing Tivoli Data Exchange Software from a CD	20
	Using FTP to Upload	21
	Installing Tivoli Data Exchange from a 3480 Tape Cartridge	23
	Initializing the MQSeries Environment	31
	Setting Up MQSeries Channels	31
	Establishing MQSeries Queue Manager Objects	32
	Initializing the OS/390 Environment	34
	Setting Up OS/390-Specific Configuration Values	34
	Initializing Tivoli Data Exchange	37
	Initializing the ISPF User Interface	40
	Running a Tivoli Data Exchange Test and Verifying Results	41
	Running Tests of Tivoli Data Exchange Using FTFV4TXF	41
	Testing the Installation with FTFPING	44
	Using FTFSTAT To Verify FTFPING	46
	Clearing Queues for Processing	48
	Running CLEARALL Procedure	48
	Installing COBOL Programs	50
	Compiling and Link Editing COBOL programs	50

chapter 3	Installing Tivoli Data Exchange on	
	Win 32	53
	Assumptions	53
	Overview	54
	System and Installation Requirements	54
	Installing Tivoli Data Exchange	55
	Installing the Server	55
	Installing the Tivoli Data Exchange Client	66
	Configuring Clients	72
	Configuring the Tivoli Data Exchange Initialization File	74
	The FTFStart Section	74
	The Environment Sections	75
	Modifying the Tivoli Data Exchange Initialization File	80
	Setting Environment Variables	80
	Restarting the Service	81
 chapter 4	 Installing Tivoli Data Exchange on	
	UNIX.	83
	Overview	84
	System and Installation Requirements	84
	UNIX Installation Steps	85
	Configuration for the switch user feature	87
	Installing the GUI	88
	Configuring the Tivoli Data Exchange GUI	89
	Initializing on Tivoli Data Exchange UNIX	90
	Starting the Tivoli Data Exchange Manager	91
	Starting the Tivoli Data Exchange Sender	92
	Starting the Tivoli Data Exchange Receiver	93
 chapter 5	 Installing Tivoli Data Exchange on	
	OpenVMS	95
	System and Installation Requirements	95
	Installing on OpenVMS	96
	Starting Tivoli Data Exchange on OpenVMS	97
	Determining Successful FTF Commands	98
	Troubleshooting MQSeries on OpenVMS	99
	Invoking a Tivoli Data Exchange Exit on OpenVMS	99
	File Type Support on OpenVMS	99

chapter 6 Installing Tivoli Data Exchange on

OS/400 101

Assumptions	101
Overview	102
System and Installation Requirements	102
OS/400 Installation Steps	102
Initializing Tivoli Data Exchange on OS/400	104
Starting an OS/400 Session	104
Using the OS/400 Tivoli Data Exchange Menu	105
Starting the Manager	106
Starting the Sender	106
Starting the Receiver	107
Starting the Logger	107
Starting the Status Daemon	107

chapter 7 Installing Tivoli Data Exchange on

4690. 109

Overview	110
What Is an MQSeries Client?	110
Why Use MQSeries Clients?	111
System and Installation Requirements	112
Installing the Client for 4690	112
Adding Logical Filenames	113
Specifying File Distribution Types on the Command Line	113
Specifying the Target File Distribution Type in the Configuration	114
Initializing the Client for 4690	114
Tivoli Data Exchange Manager Startup	114
Tivoli Data Exchange Sender Startup	115
Tivoli Data Exchange Receiver Startup	116
Configuring the Client for 4690	117
Setting Up the MQSeries Client	117
Setting Up Background Processes for Components	119
Starting Background Processes for Components	122
Setting Up Clients on Another Platform	122
Using Exits on the 4690 Client	125
Tivoli Data Exchange Commands	126
Overview	126
FTF	126
FTFMGR	126
FTFRCV	126
FTFSDR	126

FTFPING	127
FTFEND	127

chapter 8 Tivoli Data Exchange Configuration 129

Tivoli Data Exchange Configuration File	130
Tivoli Data Exchange Configuration File and MQSeries Scripts ..	130
Working with the Tivoli Data Exchange Configuration File	131
FTFNodeDefault Section	132
Tivoli Data Exchange Receiver Default Properties	132
Tivoli Data Exchange Sender Default Properties	135
Tivoli Data Exchange Manager Default Properties	138
Status Queue Default Properties	139
MVS Default Properties	142
AS/400 Default Property	144
Setting Up Clients	145
Log Queue Property	148
File Types Property	149
Translation Property	151
Notification Message Property	152
Miscellaneous Properties	154
Data Pool Section	156
FTFNodeAlias Section	159
FTFNodeOverride Defaults	161
Security	162

chapter 9 Verifying the Tivoli Data Exchange **Installation 165**

Assumptions	165
Overview	166
The Verification Procedure	166
Expected Results	167
Successful FTFPING	167
Unsuccessful FTFPING	167

Index 169

About This Guide

The *Tivoli Data Exchange Installation Guide* describes how to install and configure *Tivoli Data Exchange (TDE)* on each of its supported operating systems. It also describes how to verify the installation.

This chapter contains guidelines about the information in this manual and the conventions used to present the information. It contains the following sections:

Section	Page
About Tivoli Data Exchange	7
Who Should Read This Guide	9
How This Guide Is Organized	10
Contacting Customer Support	10
Where To Look for More Information	11
Conventions	11

About Tivoli Data Exchange

TDE provides a complete infrastructure for enabling robust Enterprise Application Integration (EAI). This advanced EAI solution bonds disparate intra- and inter-enterprise application by using IBM MQSeries to automate information exchange.

TDE leverages and exploits the inherent strengths of MQSeries to move files across heterogeneous information technology (IT) environments. It is available as an off-the-shelf product that eliminates the need for users to customize proprietary file-transfer solutions. Such solutions are mandatory in many MQSeries environments.

The TDE subsystem provides the following services:

- Moves and accepts files among all supported platforms

About This Guide

About Tivoli Data Exchange

- Provides data compression, if you require it
- Performs binary and ASCII transfers
- Transfers files regardless of their size, format, or destination
- Allows individual status tracking for any phase of the file transfer at any node across the enterprise

TDE provides integration capabilities, including business-process and workflow integration and TDE connectors.

Exits can be customized by users and can be employed in a plug-and-play manner. They are called at strategic points during an TDE transaction. Connectors are integration points that can be used to override internal I/O in favor of custom and specialized I/O. They provide for a level of data access beyond the capabilities and constraints of file-based I/O.

With connectors, data can be passed to the TDE transfer engine with a simple, consistent application programmer's interface (API) or by using an XMSeries script. The data is protected through synchpoint control. Transactional status is distributed across the enterprise. High-performance multiplexing capabilities are employed transparently.

No complex changes, design requirements, or re-engineering efforts are required for existing applications to take advantage of these features.

Who Should Read This Guide

This guide is intended for system designers, system administrators, TDE administrators, and MQSeries administrators who have good working knowledge of the IBM MQSeries product. Before using this manual, you should know how to perform a basic set of actions, including:

- Installing and customizing IBM MQSeries
- Creating a queue manager
- Defining queues
- Configuring channels
- Monitoring MQSeries queues and channels
- Configuring distributed queuing using MQSeries
- Administering MQSeries infrastructure
- Administering network protocols
- Administering operating systems on which TDE is installed and used

This guide also assumes that the user has a good working knowledge of the platforms and associated applications on which TDE operates.

About This Guide

How This Guide Is Organized

How This Guide Is Organized

The following table lists and describes this guide's chapters.

Chapter	Title	Purpose
1	About This Guide	Introduces this manual and its contents.
2	Installing Tivoli Data Exchange on OS/390	Describes how to install TDE on OS/390 systems.
3	Installing Tivoli Data Exchange on Win 32	Describes how to install TDE on Win 32 systems.
4	Installing Tivoli Data Exchange on UNIX	Describes how to install TDE on UNIX systems.
5	Installing Tivoli Data Exchange on OpenVMS	Describes how to install TDE on OpenVMS.
6	Installing Tivoli Data Exchange on OS/400	Describes how to install TDE on AS/400 systems.
7	Installing Tivoli Data Exchange on 4690	Describes how to install TDE on on 4690 systems.
8	Tivoli Data Exchange Configuration	Describes how to set TDE configuration properties and coordinate the settings with MQSeries.
9	Verifying the Tivoli Data Exchange Installation	Describes the process for verifying that TDE has been successfully installed and configured.

Contacting Customer Support

We are very interested in hearing from you about your experience with Tivoli products and documentation. We welcome your suggestions for improvements. If you have comments or suggestions about this documentation, please send e-mail to usib2hpd@vnet.ibm.com.

If you encounter difficulties with Tivoli products, contact Tivoli Customer Support. In the United States, the Tivoli number is 1-800-TIVOLI8 and the IBM number is 1-800-237-5511 (press or say **8** after you reach this number). Both of these numbers direct your call to the Tivoli Customer Support call center. In addition, you can enter <http://www.tivoli.com/support> to view the Tivoli Customer Support home page.

After you link to and submit the customer registration form, you will be able to access many customer support services on the Web. For support services outside the United States and Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Where To Look for More Information

Information about MQSeries issues is available in the MQSeries documentation. The IBM corporate website includes a web page that lists the MQSeries manuals and allows you access to an online version of each manual. At publication time, this page's URL was <http://www.software.ibm.com/ts/mqseries/library/manuals>.

The TDE documentation—manuals and readmes—is available in PDF format for viewing and printing at <http://www.commercequest.com/documentation>.

Conventions

The following elements are employed in this guide to make it easier to use:

Note:

Notes provide additional information about the current subject.

Warning:

Warnings alert you to situations that can cause problems, such as the loss of data, if you do not follow instructions carefully.

Sidebar

Sidebars contain information that do not specifically fit the flow of the current topic but is important to the topic.

All `syntax`, `operating system terms`, and `literal examples` are presented in this typeface.

Italics in a command string signify variables.

About This Guide

Conventions

Text enclosed in angle brackets (<>) denotes variable information. Replace the variable information with the actual value.

Installing Tivoli Data Exchange on OS/390

This chapter describes the steps to installing Tivoli Data Exchange (TDE) on an OS/390 system. It lists system and installation requirements, provides instructions for CD and tape installations, describes how to initialize TDE and the ISPF interface, and demonstrates how to run tests on the installed software.

This chapter includes the following sections:

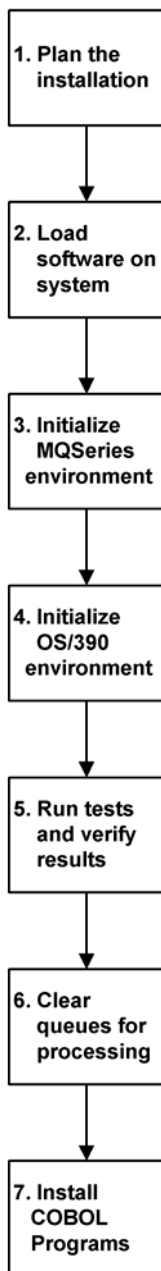
Section	Page
Overview	14
Planning the Installation	15
Reviewing System and Software Requirements	15
Loading the Tivoli Data Exchange Software	20
Initializing the MQSeries Environment	31
Initializing the OS/390 Environment	34
Running a Tivoli Data Exchange Test and Verifying Results	41
Clearing Queues for Processing	48
Installing COBOL Programs	50

Assumptions

This chapter makes the following assumptions:

- You have appropriate authorization to install software on the installation machine.
- MQSeries has been installed and appropriately configured on the installation machine.

Overview

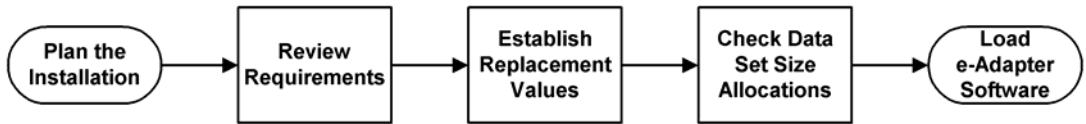


This chapter describes the process for installing TDE on an OS/390 system. The installation process starts with installation planning and ends with testing the installed software.

To install TDE on an OS/390 system, perform the following steps:

1. **Plan the installation** by determining the replacement values (see “Establishing Installation Values” on page 16), checking the system and installation requirements (see “Reviewing System and Software Requirements” on page 15), and determining disk space requirements (see “Checking Data Set Size Allocations” on page 19).
2. **Load the software** from a CD (see “Installing Tivoli Data Exchange Software from a CD” on page 20) or from a 3480 tape cartridge (see “Installing Tivoli Data Exchange from a 3480 Tape Cartridge” on page 23), as appropriate.
3. **Initialize the MQSeries environment** by setting up the MQSeries channels and establishing the queue manager objects so TDE has an environment in which to operate (see “Initializing the MQSeries Environment” on page 31).
4. **Initialize the OS/390 environment** by changing OS/390-specific settings in the TDE configuration file (see “Initializing the OS/390 Environment” on page 34) and initializing the ISPF User Interface (see “Initializing the ISPF User Interface” on page 40).
5. **Run the supplied test procedures** to exercise the TDE software that you have installed (see “Running Tests of Tivoli Data Exchange Using FTFV4TXF” on page 41). Use the FTFPING command to ensure that the TDE software is working (see “Testing the Installation with FTFPING” on page 44). Use the FTFSTAT command to check the FTFPING command’s results (see “Using FTFSTAT To Verify FTFPING” on page 46).
6. **Clear all of the queues** used during testing by using another supplied procedure. The system is ready for use (see “Clearing Queues for Processing” on page 48).
7. **Install COBOL programs** if you plan to interface with TDE through the COBOL API. Compile and link edit the COBOL programs (see “Compiling and Link Editing COBOL programs” on page 50).

Planning the Installation



This section contains the following information:

- Reviewing system and software requirements
- Establishing installation values
- Checking data set size allocations (see page 19)

Reviewing System and Software Requirements

The following software products must be installed before you can install TDE on an OS/390 system:

- Operating Systems:
 - MVS/ESA, versions 5.2.2, 5.3, or higher
 - OS/390, versions 1.3, 2.4, 2.5 or higher
- MQSeries, versions 1.1.4, 1.2, or higher
- ISPF/PDF, version 3.3 or higher
- LE/370, versions 1.5, 1.7, 1.8, or higher

Installing on Pre-OS/390 2.10 Systems: Language Environment Requirements

When installing TDE on pre-OS/390 2.10 systems, please refer to the following APARs that list the PTFs required to enable downward compatibility to your LE environment. You will have to sort through the lists and choose the PTFs applicable to your software configuration. Many of the PTFs on the lists have been superseded by others.

PQ30805 (PE - see PQ33588)
PQ33358
PQ33359
PQ33588 (PE - see PQ35954)
PQ35954
PQ45211 (PE - see PQ47362)
PQ47362

Refer to the following IBM website for additional information:

<http://www-1.ibm.com/servers/eserver/zseries/zos/le/whatsnew/down.html>

Establishing Installation Values

The TDE installation JCL and TDE configuration file contain symbolics. These symbolics represent various settings relating to TDE installation and operation. They are delimited with two plus signs at either end of the symbolic (for instance, ++HLQ++).

Before you start the installation process, you should understand the role of the symbolics and be able to change them to values that are appropriate for your installation.

This section lists and describes each symbolic used in the installation JCL and the TDEconfiguration file.

JCL Replacement Values

The following table contains the variables used in the different JCL members. You should replace these variables with values appropriate to your environment before you run the installation JCL. Each installation step that contains JCL you must run also includes a description of the high-level qualifiers you must change before you run that JCL.

Replacement Variable	Meaning
++DASD++	Unit name for DASD where the TDE data sets will reside (for example, SYSDA).
++DPATH++	Fully qualified path of the destination file (for example, ftfv4.test.dest.file).
++DQMGR++	The destination queue manager where the destination file will reside (for example, MQA2).
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++FTFINI++	Full name of the TDE default configuration file (for example, FTFV4.FTF.INI).
++LEHLQ++	High-level qualifier of the LE/370 data sets (for example, CEE.V1R5M0).
++LOADLIB++	Output data set name for the link-edited load module.

Replacement Variable	Meaning
++LQMGR++	The local queue manager (for example, MQA1).
++MEMBER++	Member name of the sample that needs to be compiled (for example, COBREQ).
++MODE++	Mode for the destination file (for example, APPEND or NOREPLACE).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).
++QMGR++	Queue manager to which TDE will connect (for example, MQA1).
++SPATH++	Fully qualified path to the source file (for example, FTFV4.TEST.SOURCE.FILE).
++SQMGR++	The source queue manager where the source file exists (for example, MQA1).
++TAPE++	Unit for the 3480 tape (for example, 3480).
++TYPE++	Type of data transfer (for example, TEXT or BINARY).
++UNIT++	The unit on which the TDE data sets will reside.
++VOLSER++	Volume serial number where the TDE data sets will reside (for example, TECH01).
++VOLUME++	The volume serial number where the TDE data sets will reside.

Tivoli Data Exchange Configuration File Values

The TDE configuration file establishes the default values by which TDE runs. The following table contains properties you must establish in the configuration file. Replace each property with a value appropriate for your environment.

The ++FTFINI++ data set that is part of the installation software contains sample values for each of the variables. The defaults supplied in your ++FTFINI++ file are:

MVSDIRBLOCKS=10

MVSPRIMALLOC=5

Installing Tivoli Data Exchange on OS/390

Planning the Installation

MVSSECALLOC=2

MVSSPACETYPE=TRK

MVSBLKSIZE=3120

MVSLRECL=80

MVTEXTWRAP=FAIL

MVSRECFM=FB

Edit the contents of the ++FTFINI++ data set and make the changes appropriate to your environment.

Configuration File Value	Description
MVSDIRBLOCKS	Determines the number of directory blocks set aside for the target file. This value must be a positive integer.
MVSPRIMALLOC	Determines the default number of primary allocation units set aside for OS/390 data transfers. This value must be a positive integer.
MVSSECALLOC	Determines the default number of secondary allocation units set aside for OS/390 data transfers. This value must be a positive integer.
MVSSPACETYPE	Determines the allocation unit used for the target file. Valid values: BLK (block), CYL (cylinder), TRK (track) Default value: TRK.
MVSUNITNAME	Determines the target's unit name.
MVSVOLUME	Determines the target's volume serial number.
MVSBLKSIZE	Determines the target file's block size. Valid values: 0 –32760.
MVSLRECL	Determines the target file's logical record length. Valid values: 1-32760.

Configuration File Value	Description
MVSTEXTWRAP	Determines the treatment of the source data record if it is longer than the target data set. Valid values: Wrap, Truncate, Fail. Default value: Fail.
MVSRECFM	Determines the record format for OS/390 data transfers. Valid values: F (fixed), V (variable), FB (fixed block), VB (variable block). Default value: FB.
MVSMODELGDG	Defines a base generation group that describes the characteristics of the Generation Data Group (GDG). This contains definitions of the attributes group such as record length, block size, the maximum number of revisions that will be supported, and many others. These definitions form the basis for creating a GDG during processing.

For a complete description of the TDE configuration file, see “TDE Configuration” in this manual.

Checking Data Set Size Allocations

Before installing TDE, you must make sure enough space exists for the TDE software. The following table contains the data sets and the number of tracks that each data set requires on a 3390. (The other space requirement is for TDE logs. The logs are written to the SYSOUT DD.)

If these space requirements changed since this manual was published, the new values will be included in the README.TXT file that is included on the installation CD.

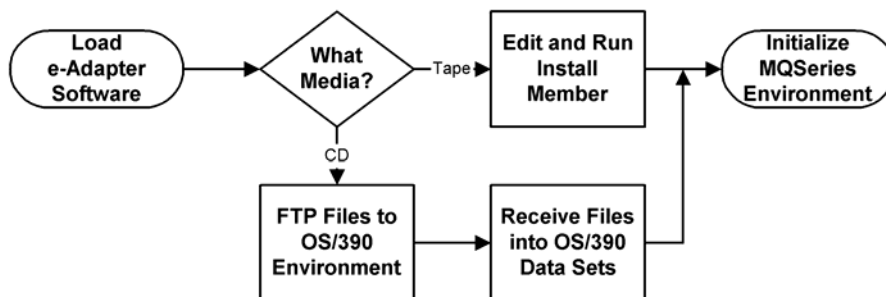
Data Set	Tracks	% Used	Device
++FTFHLQ++.FTF.APILIB	11	100	3390
++FTFHLQ++.FTF.COBSAMP	3	100	3390
++FTFHLQ++.FTF.INI	2	50	3390
++FTFHLQ++.FTF.ISPCLIB	2	50	3390

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

Data Set	Tracks	% Used	Device
++FTFHLQ++.FTF.ISPMLIB	2	50	3390
++FTFHLQ++.FTF.ISPPLIB	7	100	3390
++FTFHLQ++.FTF.JCLLIB	2	100	3390
++FTFHLQ++.FTF.LOADLIB	138	100	3390
++FTFHLQ++.FTF.SAMPLIB	4	100	3390
++FTFHLQ++.FTF.SIDEDECK	1	100	3390

Loading the Tivoli Data Exchange Software



This section contains the following information:

- Installing TDE from a CD
- Installing TDE from a 3480 tape cartridge (see page 23)

Installing Tivoli Data Exchange Software from a CD

Before you install TDE from a CD, the following conditions must exist:

- You must have the appropriate authority to access the OS/390 system.
- You must have a connection from a Win 32 - or UNIX-based system to the OS/390 system to perform the upload from the CD.
- The appropriate version of MQSeries must be operational.
- You must be able to perform a TSO command receive to unload the image into the appropriate data sets.

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

To install TDE from an installation CD to an OS/390 environment, follow these steps:

1. Establish the replacement values for the variables in the following table. The values are used in the upload procedure.

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++UNIT++	The unit on which the TDE data sets will reside.
++VOLUME++	The volume serial number where the TDE data sets will reside.

2. Upload all of the XMIT files from the installation CD to the MVS directory of your MVS host. If you use FTP, the necessary commands are listed in the following box.

Using FTP to Upload

If you use FTP to upload the files, use the following commands:

- bin
- quote site recfm=fb lrecl=80 blksize=3120
- put FTF.JCLLIB.XMIT '++FTFHLQ++.JCLLIB.XMIT'
- put FTF.ISPPLIB.XMIT '++FTFHLQ++.ISPPLIB.XMIT'
- put FTF.ISPMLIB.XMIT '++FTFHLQ++.ISPMLIB.XMIT'
- put FTF.ISPCLIB.XMIT '++FTFHLQ++.ISPCLIB.XMIT'
- put FTF.SAMPLIB.XMIT '++FTFHLQ++.SAMPLIB.XMIT'
- put FTF.INI.XMIT '++FTFHLQ++.FTF.INI.XMIT'
- put FTF.APILIB.XMIT '++FTFHLQ++.APILIB.XMIT'
- put FTF.COBSAMP.XMIT '++FTFHLQ++.COBSAMP.XMIT'
- put FTF.SIDEDECK.XMIT '++FTFHLQ++.FTF.SIDEDECK.XMIT'
- quote site cyl pri=3 sec=2 unit=++UNIT++ volume=++VOLUME++
- put FTF.LOADLIB.XMIT '++FTFHLQ++.LOADLIB.XMIT'
- quit

3. Log on to the OS/390 machine on which you are installing TDE.
4. Proceed to the ISPF command shell, option 6, enter TSO or Workstation Commands.

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

5. Enter the commands listed in the following table. For each data set, enter the command listed in the second column. When you are prompted, enter the command in the third column as shown below.

Data Set	Command	Response
JCL Library	RECEIVE INDA (‘+++FTFHLQ++JCLLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.JCLLIB’)
Sample Library	RECEIVE INDA (‘+++FTFHLQ++SAMPLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.SAMPLIB’)
INI File	RECEIVE INDA (‘+++FTFHLQ++FTF.INI.XMIT’)	DA(‘+++FTFINI++’)
LOADLIB	RECEIVE INDA (‘+++FTFHLQ++LOADLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.LOADLIB’)
ISPPLIB	RECEIVE INDA (‘+++FTFHLQ++ISPPLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.ISPPLIB’)
ISPMLIB	RECEIVE INDA (‘+++FTFHLQ++ISPMLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.ISPMLIB’)
ISPCLIB	RECEIVE INDA (‘+++FTFHLQ++ISPCLIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.ISPCLIB’)
APILIB	RECEIVE INDA (‘+++FTFHLQ++APILIB.XMIT’)	DA(‘+++FTFHLQ++.FTF.APILIB’)
COBSAMP (COBOL Samples)	RECEIVE INDA (‘+++FTFHLQ++COBSAMP.XMIT’)	DA(‘+++FTFHLQ++.FTF.COBSAMP’)
Export Definitions	RECEIVE INDA (‘+++FTFHLQ++FTF.SIDEDECK.XMIT’)	DA(‘+++FTFHLQ++.FTF.SIDEDECK’)

6. Verify that the files are present on the target system.
7. Delete the following files:
 - ++FTFHLQ++.FTF . JCLLIB.XMIT
 - ++FTFHLQ++.FTF . SAMPLIB.XMIT
 - ++FTFHLQ++.FTF . INI.XMIT
 - ++FTFHLQ++.FTF . LOADLIB.XMIT
 - ++FTFHLQ++.FTF . ISPPLIB.XMIT
 - ++FTFHLQ++.FTF . ISPMLIB.XMIT
 - ++FTFHLQ++.FTF . ISPCLIB.XMIT
 - ++FTFHLQ++.FTF . APILIB.XMIT
 - ++FTFHLQ++.FTF . COBSAMP.XMIT
 - ++FTFHLQ++.FTF.SIDEDECK.XMIT
8. If you are ready to set up TDE, then proceed to “Initializing Tivoli Data Exchange” on page 37.

Installing Tivoli Data Exchange from a 3480 Tape Cartridge

Before you install TDE from a 3480 tape cartridge, the following conditions must exist:

- You must have the appropriate authority to access the OS/390 system.
- The appropriate version of MQSeries must be operational.

Installing TDE from a 3480 tape cartridge consists of the following steps:

1. Uploading the tape’s first data set (see page 24)
2. Installing the TDE data sets (see page 25)
3. Uploading Tape’s First Data Set

Uploading Tape’s First Data Set

To upload the tape’s first data set, follow these steps:

- 1. In the FTFUPLOD JCL (displayed below), change the following symbolic names:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++UNIT++	The unit on which the TDE data sets will reside.
++VOLUME++	The volume on which the TDE data sets will reside.

- 2. Replace the job card with a valid job card that is more appropriate for your environment.
- 3. Run the FTFUPLOD JCL. This step creates the data set ++FTFHLQ++.FTF.JCLLIB.

FTFUPLOD JCL

```
//FTFUPLOD JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//INSTALL EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//INPUT1 DD DSN=FTF.JCLLIB,DISP=OLD,VOL=SER=FTFMQ0,
// UNIT=3480,LABEL=(1,SL)
//OUTPUT1 DD DSN=++FTFHLQ++.FTF.JCLLIB,
// DISP=(,CATLG),UNIT=++UNIT++,
// VOL=SER=++VOLUME++,
// SPACE=(TRK,(10,5,10),RLSE)
//SYSIN DD *
COPY INDD=INPUT1,OUTDD=OUTPUT1
/*
//
```


Installing Tivoli Data Exchange Data Sets

To install the TDE data sets, follow these steps:

1. Edit the INSTALL member in the ++FTFHLQ++.FTF.JCLLIB data set.
 - Replace the job card with a valid job card for your system.
 - Replace the following symbols with values appropriate for your system:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++DASD++	Unit for DASD where the TDE data sets will reside (for example, SYSDA).
++TAPE++	Unit for the 3480 tape (for example, 3480).
++VOLSER++	Volume serial number where the TDE data sets will reside (for example, TECH01).

Note:

The INSTALL JCL is displayed on the next several pages. Because of the length of this JCL, it is displayed in functional pieces.

2. Run the INSTALL JCL job. Each of the job steps of the INSTALL job stream is listed on the following pages so that you can see what is being loaded. The JCL runs as one stream and loads all of the necessary data sets from tape.

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

INSTALL JCL

The following section of the INSTALL JCL copies the TDE configuration file to the system.

```
//*-----*
//* DOWNLOAD THE TDE INI FILE FROM 2,SL ON          *
//* VOLSER=FTFMQ0                                   *
//*-----*
//*
//INI      EXEC  PGM=IEBGENER,REGION=4M
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=FTF.INI,DISP=SHR,
//          UNIT=++TAPE++,LABEL=(2,SL),VOL=(,RETAIN,SER=FTFMQ0)
//SYSUT2   DD   DSN=++FTFHLQ++.FTF.INI,
//          DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
//          SPACE=(TRK,(1,1)),UNIT=++DASD++
//SYSIN    DD   DUMMY
//*
```

The following section of the INSTALL JCL copies the TDE LOADLIB to the system.

```
//*-----*
//* DOWNLOAD THE TDE LOADLIB LIBRARY FROM 3,SL ON   *
//* VOLSER=FTFMQ0                                   *
//*-----*
//*
//LOADLIB  EXEC  PGM=IEBCOPY,REGION=4M
//SYSPRINT DD   SYSOUT=*
//IN       DD   DSN=FTF.LOADLIB,DISP=SHR,
//          UNIT=++TAPE++,LABEL=(3,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT      DD   DSN=++FTFHLQ++.FTF.LOADLIB,
//          DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
//          SPACE=(CYL,(10,2,30),RLSE),UNIT=++DASD++
//SYSIN    DD   *
//          COPY INDD=IN,OUTDD=OUT
//*
//*
```

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

The following section of the INSTALL JCL copies the TDE ISPLIB data set to the system.

```
//*-----*
//* DOWNLOAD THE TDE ISPLIB FROM 4,SL ON          *
//* VOLSER=FTFMQ0                                *
//*-----*
//*
//ISPLIB EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.ISPLIB,DISP=SHR,
// UNIT=++TAPE++,LABEL=(4,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.ISPLIB,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//*
```

The following section of the INSTALL JCL copies the TDE ISPMLIB data set to the system.

```
//*-----*
//* DOWNLOAD THE e-Adapater ISPMLIB FROM 5,SL ON    *
//* VOLSER=FTFMQ0                                *
//*-----*
//*
//ISPMLIB EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.ISPMLIB,DISP=SHR,
// UNIT=++TAPE++,LABEL=(5,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.ISPMLIB,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//*
```

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

The following section of the INSTALL JCL copies the TDE ISPPLIB data set to the system.

```
//*-----*
//* DOWNLOAD THE e-Adapater ISPPLIB FROM 6,SL ON          *
//* VOLSER=FTFMQ0                                         *
//*-----*
//*
//ISPMLIB EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.ISPPLIB,DISP=SHR,
// UNIT=++TAPE++,LABEL=(6,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.ISPPLIB,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
```

The following section of the INSTALL JCL copies the TDE SAMPLIB data set to the system.

```
//*-----*
//* DOWNLOAD THE e-Adapater SAMPLIB FROM 7,SL ON          *
//* VOLSER=FTFMQ0                                         *
//*-----*
//*
//SAMPLIB EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.SAMPLIB,DISP=SHR,
// UNIT=++TAPE++,LABEL=(7,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.SAMPLIB,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
```

Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

The following section of the INSTALL JCL copies the TDE APILIB data set to the system.

```
//*-----*
//* DOWNLOAD THE TDE APILIB FROM 8,SL ON          *
//* VOLSER=FTFMQ0                                *
//*-----*
//*
//APILIB EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.APILIB,DISP=SHR,
// UNIT=++TAPE++,LABEL=(8,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.APIPLIB,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
```

The following section of the INSTALL JCL copies the TDE COBSAMP data set to the system.

```
//*-----*
//* DOWNLOAD THE TDE COBSAMP FROM 9,SL ON          *
//* VOLSER=FTFMQ0                                *
//*-----*
//*
//COBSAMP EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//IN DD DSN=FTF.COBSAMP,DISP=SHR,
// UNIT=++TAPE++,LABEL=(9,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT DD DSN=++FTFHLQ++.FTF.COBSAMP,
// DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
// SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
```

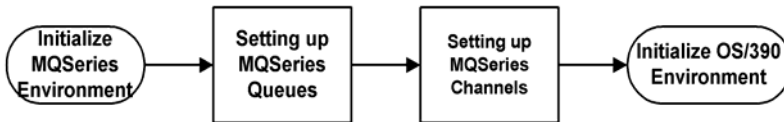
Installing Tivoli Data Exchange on OS/390

Loading the Tivoli Data Exchange Software

The following section of the INSTALL JCL copies the TDE SIDEDECK data set to the system.

```
//*-----*
//*  DOWNLOAD THE TDE COBSAMP FROM 11,SL ON          *
//*  VOLSER=FTFMQ0                                   *
//*-----*
//*
//SIDEDECK EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD   SYSOUT=*
//IN        DD   DSN=FTF.COBSAMP,DISP=SHR,
//           UNIT=++TAPE++,LABEL=(11,SL),VOL=(,RETAIN,SER=FTFMQ0)
//OUT       DD   DSN=++FTFHLQ++.FTF.SIDEDECK,
//           DISP=(NEW,CATLG,DELETE),VOL=SER=++VOLSER++,
//           SPACE=(TRK,(1,1,10),RLSE),UNIT=++DASD++
//SYSIN     DD   *
           COPY INDD=IN,OUTDD=OUT
/*
//
```

Initializing the MQSeries Environment



This section contains the following information:

- Setting up MQSeries channels
- Establishing MQSeries queue manager objects (see page 32)

Setting Up MQSeries Channels

Channels are required in an MQSeries enterprise in order to transfer information from one queue manager to another. MQSeries commands required to establish channels are found in the MQSeries documentation listed in “Where To Look for More Information” on page 11 of this manual. Generally, the steps to setting an MQSeries channel are:

- Starting a queue manager
- Starting a listener
- Defining a channel
- Starting a channel

There are many ways to perform these steps. If you are not familiar with MQSeries administration, please contact your MQSeries administrator.

Establishing MQSeries Queue Manager Objects

Create the queue manager objects needed to run TDE by running FTFMAKEQ JCL (displayed at the bottom of this page). The FTFMAKEQ JOB references the FTFJDEFQL JCL member of the JCLLIB (displayed on page 33). The FTFJDEFQL JCL member establishes the STAGE, SYSTEM, SYNC, and CONTROL queue manager objects.

To set up the MQSeries queue manager objects, follow these steps:

1. Edit the FTFMAKEQ JOB data set and make the changes to the following symbolic names:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).
++QMGR++	Queue manager to which TDE will connect (for example, MQA1)

2. Replace the job card with a valid job card for your environment.
3. Run the FTFMAKEQ job step.

FTFMAKEQ JCL

```
//FTFMAKEQ JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//FTFMAKEQ EXEC PGM=CSQUTIL,REGION=4M,PARM='++QMGR++'
//STEPLIB DD DSN=++MQMHLQ++.SCSQAUTH,DISP=SHR
// DD DSN=++MQMHLQ++.SCSQANLE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD DSN=++FTFHLQ++.FTF.JCLLIB(FTFCOMND),DISP=SHR
//FTFCOMND DD DSN=++FTFHLQ++.FTF.JCLLIB(FTFJDEFQ),DISP=SHR
//
```


FTFJDEFQL JCL

The following JCLLIB member, FTFJDEFQL, does not need to be edited. It is referenced by FTFMAKEQ.

```
*****
*
* THESE COMMANDS WILL CREATE THE REQUIRED QUEUE MANAGER
* OBJECTS FOR TDE
*
*****
DEFINE QLOCAL(FTFRCV.CONTROL) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFRCV.STAGE) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFRCV.STAGE.1) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFRCV.SYNC) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFRCV.SYNC.1) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFRCV.SYSTEM) SHARE DEFSOPT(SHARED)

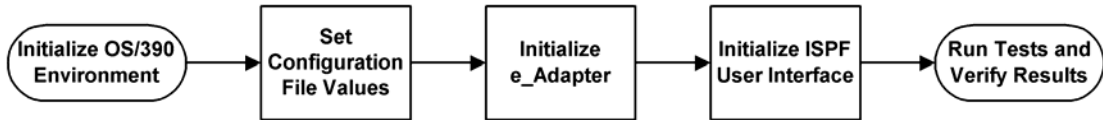
DEFINE QLOCAL(FTFSDR.CONTROL) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.STAGE) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.STAGE.1) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.STAGE.CONTROL) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.SYNC) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.SYNC.1) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSDR.SYSTEM) SHARE DEFSOPT(SHARED)

DEFINE QLOCAL(FTFMGR.CONTROL) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFMGR.SYNC) SHARE DEFSOPT(SHARED)

DEFINE QLOCAL(FTFSTAT.DETAIL) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFSTAT.CONTROL) SHARE DEFSOPT(SHARED)

DEFINE QLOCAL(FTFICC) SHARE DEFSOPT(SHARED)
DEFINE QLOCAL(FTFDATA) SHARE DEFSOPT(SHARED)
```

Initializing the OS/390 Environment



This section contains the following information:

- Setting OS/390-specific configuration values
- Initializing TDE (see page 37)
- Initializing ISPF User Interface (see page 40)

Setting Up OS/390-Specific Configuration Values

The MVS Defaults section of the TDE configuration file contains settings specific to TDE operations in the OS/390 environments. For the most part, these settings relate to the allocation of space for OS/390 target files. The TDE configuration file is located in the data set specified by the ++FTFINI++ symbol.

Specifying an Esoteric Unit Name

To specify an esoteric name for the OS/390 UNIT value, follow these steps:

- Do not set a value in the configuration file's MVSVOLUME stanza.
- Specify the *unit* value in the appropriate interface (FTF command, FTFReq API, ISPF interface, 5250 interface, TDE GUI), or in the MVSUNITNAME stanza in the TDE configuration file on either the TDE Sender or the TDE Receiver.

Installing Tivoli Data Exchange on OS/390

Initializing the OS/390 Environment

The following table lists all of the configuration file variables with a description of each variable. Each variable is listed in the MVS Defaults section of the configuration file supplied in the ++FTFINI++ data set.

Configuration File Value	Description
MVSDIRBLOCKS	Determines the number of directory blocks set aside for the target file. This value must be a positive integer.
MVSPRIMALLOC	Determines the default number of primary allocation units set aside for OS/390 data transfers. This value must be a positive integer.
MVSSECALLOC	Determines the default number of secondary allocation units set aside for OS/390 data transfers. This value must be a positive integer.
MVSPACETYPE	Determines the allocation unit used for the target file. Valid values: BLK (block), CYL (cylinder), TRK (track) Default value: TRK.
MVSUNITNAME	Determines the target's unit name.
MVSVOLUME	Determines the target's volume serial number.
MVSBLKSIZE	Determines the target file's block size. Valid values: 0 –32760.
MVSLRECL	Determines the target file's logical record length. Valid values: 1-32760.
MVSTEXTWRAP	Determines the treatment of the source data record if it is longer than the target data set. Valid values: Wrap, Truncate, Fail. Default value: Fail.

Installing Tivoli Data Exchange on OS/390

Initializing the OS/390 Environment

Configuration File Value	Description
MVSRECFM	Determines the record format for OS/390 data transfers. Valid values: F (fixed), V (variable), FB (fixed block), VB (variable block). Default value: FB.
MVSMODELGDG	Defines a base generation group that describes the characteristics of the GDG. This contains definitions of the attributes group such as record length, block size, the maximum number of revisions that will be supported, and many others. These definitions form the basis for creating a GDG during processing.

The MVS Defaults section contains the following OS/390-specific settings in your ++FTFINI++ file are:

MVSDIRBLOCKS=10

MVSPRIMALLOC=5

MVSSECALLOC=2

MVSSPACETYPE=TRK

MVSUNITNAME=SYSALLDA

MVSVOLUME=TECH01

MVSBKSIZE=3120

MVSLRECL=80

MVTEXTWRAP=FAIL

MVSRECFM=FB

Edit the contents of the ++FTFINI++ data set and make the changes appropriate to your environment.

Initializing Tivoli Data Exchange

After you have installed TDE on an OS/390 system, you must perform the following steps to complete the installation and verify that the installation was successful:

1. If you are using the FTFAUTH security exit provided, copy this member into an APF authorized library.
2. If you are using any MQSeries client connections to TDE on MVS, including the TDE MMC, then copy the FTFCVRT1 module into a data set in the concatenation for the CSQXLIB DD statement in the MQSeries channel initiator job.
3. The member ++FTFHLQ++.FTF.LOADLIB(FTFCVRT1) must reside in an APF-authorized load module library. One way to accomplish this is to move it to the ++MQMHLQ++.SCSQAUTH data set. Another way is to make ++FTFHLQ++.FTF.LOADLIB APF-authorized during installation. Your system programmer should know the best way this to accomplish this at your site.

Additionally, all MQSeries user exits, including FTFCVRT1, must be in the MQSeries Channel Initiator STEPLIB. Refer to the MQSeries Channel Initiator JCL for the appropriate User Exit Library data set. If FTFCVRT1 is not available as described above, non-EBCDIC platforms which attach to an MVS TDE server will experience MQSeries 2119 errors.

4. Log on to the OS/390 machine on which TDE has been installed.
5. Supply a job card that is valid for your environment.
6. Edit the FTFSTART JCL (displayed on page 39). Make the changes to the following symbolic names:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++FTFINI++	Full name of the TDE default initialization file (for example, FTFV4.FTF.INI).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).
++QMGR++	Queue manager that TDE will connect to (for example, MQA1).

Installing Tivoli Data Exchange on OS/390

Initializing the OS/390 Environment

7. To start the TDE components, run the FTFSTART job. Within that job is an EXEC statement that runs the FTFSTART program. The FTFSTART argument takes the following format:

FTFSTART -cfile *configFile* -ofile *optionsFile* -sysoutclass *classChar*

Where:

- **-cfile** *configFile* – The ++FTFINI++ file that you previously configured.
 - **-ofile** *optionsFile* – Contains the fully qualified path and filename of the options file. The options file allows you to specify command-line options in a file, rather than on the command line. An example is not included in the installation set. For more information, see “Using the Options File” in the *Tivoli Data Exchange User’s Guide*.
 - **-sysoutclass** *classChar* – Contains the one-character class designation.
8. Run TDE as a started task.

Tivoli Data Exchange Installation Guide 39

Initializing the ISPF User Interface

The ISPF interface allows you to interactively submit and monitor transactions in an OS/390 environment. For more information about using the ISPF interface, “see “Using the ISPF User Interface” on page 119” in the *Tivoli Data Exchange User's Guide*. This section describes how to initialize the ISPF user interface. Perform the steps in this section after you initialize TDE.

To initialize the ISPF user interface, follow these steps:

1. Customize the following values in the REXX exec EAUI in the ++FTFHLQ++.ISPCLIB:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++QMGR++	Queue manager that TDE will connect to (for example, MQA1).
++INIFILE++	Path to the TDE configuration file data sets (for example, FTFV4.ETF.INI).

2. Include the exec in your ISR@PRIM, ISPF primary function menu. Check with your OS/390 Systems Programmer on how to include this option within your ISPF panels.
3. From a TSO command line facility, enter the following command to execute the REXX EXEC. This command can be invoked at the command line:

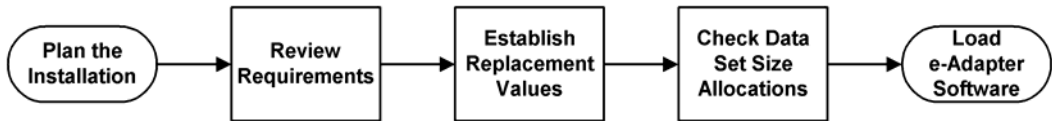
```
EXEC `++FTFHLQ++FTF.ISPCLIB(EAUI)` `lqm  
configFile'
```

Where:

- ++FTFHLQ++ contains the high-level qualifier used for the TDE installation.
- *lqm* identifies the local queue manager.
- *configFile* identifies the TDE configuration file.

If you do not specify the *lqm* and *configFile* values, the default values coded within the EXEC are used.

Running a Tivoli Data Exchange Test and Verifying Results



This section contains the following information:

- Using FTFV4TXF to test the installation
- Using FTFPING to test the installation (see page 44)
- Using FTFSTAT to test results (see page 46)

Running Tests of Tivoli Data Exchange Using FTFV4TXF

The FTFV4TXF JCL that resides in FTF.JCLLIB sends a test message. You can use this program to test the TDE installation. The FTFV4TXF JCL is displayed on page 43.

To use FTFV4TXF to test the TDE installation, follow these steps:

1. Replace the job card with a valid job card for your environment.
2. Edit the FTFV4TXF JCL and make the changes to the following symbolic names:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).
++FTFINI++	Full name of the TDE configuration file (for example, FTFV4.FTF.INI).
++LQMGR++	The local queue manager (for example, MQA1).
++SQMGR++	The source queue manager where the source file exists (for example, MQA1).

Installing Tivoli Data Exchange on OS/390

Running a Tivoli Data Exchange Test and Verifying Results

Symbolic Name	Meaning
++DQMGR++	The destination queue manager where the destination file will reside (for example, MQA2).
++SPATH++	Fully qualified path to the source file (for example, FTFV4.TEST.SOURCE.FILE).
++DPATH++	Fully qualified path of the destination file (for example, FTFV4FTFV4.TEST.DEST.FILE).
++MODE++	Mode of the destination file (for example, APPEND OR NOREPLACE).
++TYPE++	Type of data transfer (for example, TEXT or BINARY).

3. Run the FTFV4TXF job to test the TDE installation.

FTFV4TXF JCL

```
//FTFV4TXF JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID//FTFREQ
EXEC PGM=FTF,REGION=0M,TIME=NOLIMIT,
//      PARM='TRAP(OFF) /-OFILE DD:SYSIN'
//*
//STEPLIB DD DISP=SHR,DSN=++FTFHLQ++.FTF.LOADLIB
//      DD DISP=SHR,DSN=++MQMHLQ++.SCSQAUTH
//      DD DISP=SHR,DSN=++MQMHLQ++.SCSQANLE
//SYSPRINT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FTFLOG00 DD SYSOUT=*
//SYSIN DD *
-CFILE          ++FTFINI++
-LQM            ++LQMGR++
-SQM            ++SQMGR++
-SPATH          ++SPATH++
-DQM            ++DQMGR++
-DPATH          ++DPATH++
-MODE           ++MODE++
-TYPE           ++TYPE++
/*
//
```

Testing the Installation with FTFPING

You can use the FTFPING command to test connectivity with TDE components. This command sends a ping message to each of the components specified in the command-line arguments. You can execute the FTFPING command from the TSO command line. This section lists a partial syntax model for the FTFPING command and an example. For complete information about the FTFPING command, see “FTFPING” in the *Tivoli Data Exchange Technical Reference*.

This FTFPING command has the following syntax:

Using a configuration file:

```
FTFPING -lqm localQueueMgr -oqm origQueueMgr -sqm  
sourceQueueMgr -dqm destQueueMgr -cfile configFile
```

Using a configuration queue:

```
FTFPING -lqm localQueueMgr -oqm origQueueMgr -sqm  
sourceQueueMgr -dqm destQueueMgr -cq configQueueName
```

Where:

- **-lqm localQueueMgr** – Determines the queue manager from which the FTFPING command is issued. This value is required on OS/390 systems.
- **-oqm origQueueMgr** – Determines the queue manager where the TDE Manager operates. If this value is not specified, the oqm is given the same value as the lqm.
- **-sqm sourceQueueMgr** – Determines the queue manager on the TDE Sender.
- **-dqm destQueueMgr** – Determines the queue manager on the TDE Receiver.
- **-cfile configFile** – Can contain the fully qualified path and filename for the TDE configuration file. On OS/390 platforms, if no -cq argument is specified, this value must be specified. You cannot specify both a -cfile and a -cq argument in the same command.

Installing Tivoli Data Exchange on OS/390

Running a Tivoli Data Exchange Test and Verifying Results

- **-cq** *configQueueName* – Displays the queue from which the configuration information is to be retrieved for this TDE instance on this node. On OS/390 platforms, if no -cfile value is specified, this value must be specified. The -cq argument points TDE to the queue name rather than to the standard configuration file. You cannot specify both a -cfile and a -cq argument in the same command.

Configuration File and Queue Order of Precedence

On platforms other than OS/390, if you do not specify either a configuration file or a configuration queue, TDE checks the FTF_CONFIG_QUEUE environment variable and uses the specified queue. If this environment variable is not set, TDE checks the FTF_CONFIG_FILE environment variable and uses the specified file. If neither environment variable is set and no command-line argument is set, the command fails.

Example

In this example, the FTFPING command is issued from the queue manager called ADMIN1. The TDE Manager running on PROD11A is pinged. If it is running, it sends the ping to the TDE Sender running on PROD22. If it is running, it sends the ping to the TDE Receiver on PROD09B. If it is running, it sends an FTFPING response back to ADMIN1.

```
FTFPING -lqm ADMIN1 -oqm PROD11A -sqm PROD22 -dqm  
PROD09B -cfile FTF.INI
```

Using FTFSTAT To Verify FTFPING

The FTFSTAT command retrieves status information about current transactions. You can use this command to get status information about the FTFPING command you used to test TDE components. For complete information about the FTFSTAT command, see “FTFSTAT” in the *Tivoli Data Exchange Technical Reference*. The FTFSTAT command takes the format:

Using a configuration file:

```
FTFSTAT -lqm localQueueMgr -oqm origQueueMgr -sqm  
sourceQueueMgr -dqm destQueueMgr -rqm reqQueueManager  
-cfile configFile
```

Using a configuration queue:

```
FTFSTAT -lqm localQueueMgr -oqm origQueueMgr -sqm  
sourceQueueMgr -dqm destQueueMgr -rqm reqQueueManager  
-cq configQueueName
```

Where:

- **-lqm localQueueMgr** – Determines the local queue manager value for which status records are returned.
- **-oqm origQueueMgr** – Determines the originating queue manager value for which status records are returned.
- **-sqm sourceQueueMgr** – Determines the source queue manager value for which status records are returned.
- **-dqm destQueueMgr** – Determines the destination queue manager value for which status records are returned.
- **-rqm reqQueueManager** – Determines the requesting queue manager for which status records are returned. The rqm is the queue manager from which a data-transfer request was entered.
- **-cfile configFile** – Can contain the fully qualified path and filename for the TDE configuration file. On OS/390 platforms, if no -cq argument is specified, this value must be specified. You cannot specify both a -cfile and a -cq argument in the same command.

Installing Tivoli Data Exchange on OS/390

Running a Tivoli Data Exchange Test and Verifying Results

- **-cq *configQueueName*** – Displays the queue from which the configuration information is to be retrieved for this TDE instance on this node. On OS/390 platforms, if no -cfile value is specified, this value must be specified. The -cq argument points TDE to the queue name rather than to the standard configuration file. You cannot specify both a -cfile and a -cq argument in the same command.

Configuration File and Queue Order of Precedence

On platforms other than OS/390, if you do not specify either a configuration file or a configuration queue, TDE checks the FTF_CONFIG_QUEUE environment variable and uses the specified queue. If this environment variable is not set, TDE checks the FTF_CONFIG_FILE environment variable and uses the specified file. If neither environment variable is set and no command-line argument is set, the command fails.

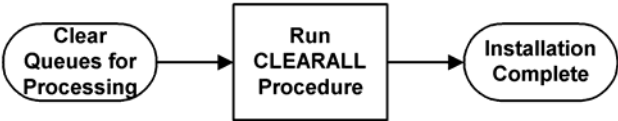
Example

In the following example, values are specified for each type of queue manager.

```
FTFSTAT -lqm PROD15A -oqm PROD22B -sqm PROD22B -dqm  
PROD12C -rqm PROD15A -cfile FTF.INI
```

All status information is displayed in the SYSOUT queue.

Clearing Queues for Processing



This section contains the following information:

- Running CLEARALL procedure (to remove test data).

Running CLEARALL Procedure

Use the CLEARALL procedure to empty all of the TDE queues of the test messages and to prepare the system for live processing. This procedure is included as the CLEARALL member in the supplied JCLLIB.

Note:

If you already have TDE in production, you may not want to run this procedure since it will clear the production messages from the queues along with the test messages.

Perform the following steps to clear the queues of content created in the verification test:

1. Edit the CLEARALL JCL (displayed on page 49) and make changes to the following symbolic names:

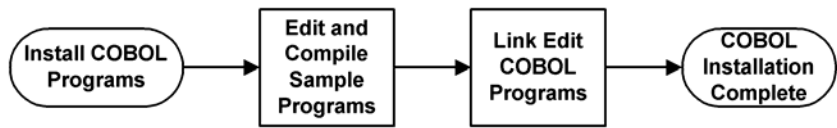
Symbolic Name	Meaning
++QMGR++	Queue manager to which TDE will connect (for example, MQA1).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).

2. Replace the job card with one that is more appropriate to your environment.
3. Run the CLEARALL Job.
4. The TDE Receiver, TDE Sender, and TDE Manager queues are cleared.

CLEARALL JCL

```
//FTFCLEAR JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1 EXEC PGM=CSQUTIL,REGION=4M,PARM=++QMGR++
//STEPLIB DD DSN=++MQMHLQ++.SCSQAUT,DISP=SHR
// DD DSN=++MQMHLQ++.SCSQANLE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EMPTY QUEUE (FTFRCV.CONTROL)
EMPTY QUEUE (FTFRCV.STAGE)
EMPTY QUEUE (FTFRCV.STAGE.1)
EMPTY QUEUE (FTFRCV.SYNC)
EMPTY QUEUE (FTFRCV.SYNC.1)
EMPTY QUEUE (FTFRCV.SYSTEM)
EMPTY QUEUE (FTFSDR.CONTROL)
EMPTY QUEUE (FTFSDR.STAGE)
EMPTY QUEUE (FTFSDR.STAGE.1)
EMPTY QUEUE (FTFSDR.STAGE.CONTROL)
EMPTY QUEUE (FTFSDR.SYNC)
EMPTY QUEUE (FTFSDR.SYNC.1)
EMPTY QUEUE (FTFSDR.SYSTEM)
EMPTY QUEUE (FTFMGR.CONTROL)
EMPTY QUEUE (FTFMGR.SYNC)
EMPTY QUEUE (FTFSTAT.CONTROL)
EMPTY QUEUE (FTFSTAT.DETAIL)
EMPTY QUEUE (FTF.ICC)
EMPTY QUEUE (FTFDATA)
/*
/*
```

Installing COBOL Programs



This section contains the following information:

- Compiling and link editing COBOL programs

Compiling and Link Editing COBOL programs

To compile and link edit COBOL programs, follow these steps on the FCOMCOMP JCL (displayed on page 51):

1. Replace the job card with one that is more appropriate to your environment.
2. Edit the FCOBCOMP JCL and make the changes to the following symbolic names in the provided JCL:

Symbolic Name	Meaning
++FTFHLQ++	High-level qualifier of the TDE data sets (for example, FTFV4).
++MQMHLQ++	High-level qualifier of the MQSeries data sets (for example, SYS1.V114).
++LEHLQ++	High-level qualifier of the LE/370 data sets (for example, CEE.V1R5M0).
++MEMBER++	Member name of the sample that needs to be compiled (for example, COBREQ).
++LOADLIB++	Output data set name for the compiled/link-edited output.

3. Before running the FCOBCOMP JCL, replace the IGYWCPL procedure in the COMPILE step with a procedure that is used at your location.

4. Also, the SYSLIB DD should have the following data sets concatenated before compiling:
- ++MQMHLQ++.SCSQLOAD
 - ++FTFHLQ++.FTF.APILIB
 - ++LEHLQ++.SCEELKED

FCOBCOMP JCL is part of FTF.JCLLIB that was downloaded in an earlier step. The statements in FCOBCOMP JCL follow.

FCOBCOMP JCL

```
//FCOBCOMP JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//COMPILE EXEC PROC=IGYWCPL,
// PARM.COBOL='FLAG(E,E),LIST'
//COBOL.SYSIN DD
DISP=SHR,DSN=++FTFHLQ++.FTF.COBSAMP(++MEMBER++)
//COBOL.SYSLIB DD DISP=SHR,DSN=++MQMHLQ++.SCSQCOBC
// DD DISP=SHR,DSN=++FTFHLQ++.FTF.COBSAMP
//*
//PLKED.SYSLIB DD DISP=SHR,DSN=++FTFHLQ++.FTF.OBJLIB
// DD DISP=SHR,DSN=++MQMHLQ++.SCSQ370
//LKED.CSQSTUB DD DISP=SHR,DSN=++MQMHLQ++.SCSQLOAD
//LKED.FTFLIB DD DISP=SHR,DSN=++FTFHLQ++.FTF.APILIB
//LKED.SYSLIB DD DISP=SHR,DSN=++LEHLQ++.SCEELKED
//LKED.SYSIN DD *
MODE AMODE(31),RMODE(24)
INCLUDE SYSLIB(EDCSTART)
INCLUDE SYSLIB(CEEROTB)
INCLUDE SYSLIB(IGZENRI)
INCLUDE SYSLIB(@@CBL2C)
INCLUDE CSQSTUB(CSQBSTUB)
INCLUDE FTFLIB(TERMNULL)
INCLUDE FTFLIB(SETADDR)
INCLUDE FTFLIB(FTFTTOKN)
INCLUDE FTFLIB(FTFATOE)
INCLUDE FTFLIB(TIIGUID)
INCLUDE FTFLIB(FTFALLOC)
NAME ++MEMBER++(R)
//LKED.SYSLMOD DD DISP=SHR,DSN=++LOADLIB++
//
```

Installing Tivoli Data Exchange on Win 32

This chapter lists instructions for installing Tivoli Data Exchange (TDE) on a Win 32 system. It lists system and installation requirements, installation steps, and describes how to configure the TDE service component.

This chapter includes the following sections:

Section	Page
Overview	54
System and Installation Requirements	54
Installing Tivoli Data Exchange on Win 32	53
Configuring the Tivoli Data Exchange Initialization File	74
Setting Environment Variables	80
Restarting the Service	81

Assumptions

This chapter makes the following assumptions:

- You have appropriate permissions to install software on the installation machine.
- MQSeries has been installed and properly configured on the installation machine.
- If you are replacing a current installation of TDE, the current installation has been shut down.

Overview

To install TDE on a Win 32 system, perform the following steps:

1. Make sure the system and installation requirements have been satisfied (see “System and Installation Requirements” on page 54).
2. Install TDE from the installation CD (see “Installing the Server” on page 55).
3. Configure the TDE initialization file (see “Configuring Clients” on page 72).
4. Restart the TDE service (see “Restarting the Service” on page 81).

System and Installation Requirements

The following minimum software levels must be installed before you can install TDE:

- Win 32
- MQSeries, version 2.1 or higher for MQSeries Server; or MQSeries, version 5.0 or higher for MQSeries Client.

The following software is required to run the TDE MMC:

- Internet Explorer 4.0 SP1 or greater
- Microsoft Management Console 1.1 or greater

The following additional requirements exist for running the TDE GUI:

- Java Platform 2 (recommended)

- Or -

- Java Runtime Environment (JRE), included in the TDE installation

You can download the latest versions of the JDK from the Sun Products and APIs web page (www.javasoft.com/products/index.html).

The JRE is included in the installed version of TDE and is found in the FTFGUI directory in the self-extracting Zip files called jre12i-win32.exe.

Installation of the TDE server requires approximately 20 MB of free disk space.

Installing Tivoli Data Exchange

You can install one of two versions of TDE on Win 32. The TDE server is configured to run on MQSeries servers. The TDE client is configured to run on an MQSeries client.

Although the server and the client perform the same basic TDE functions, there are differences in the installation process and the process for starting TDE client components.

For information about installing the TDE server, see “Installing the Server” on page 55.

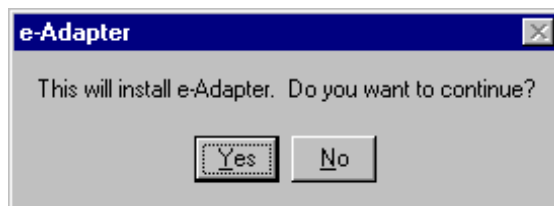
For information about installing and configuring the TDE client, see the following sections:

- “Installing the Tivoli Data Exchange Client” on page 66
- “Configuring Clients” on page 72

Installing the Server

To install the server version of TDE in the Win 32 environment, follow these steps:

1. Run *Setupex.exe* in the server directory found in the installation CD's \Win 32 directory.
2. The TDE dialog box appears. Click **Yes** to continue.

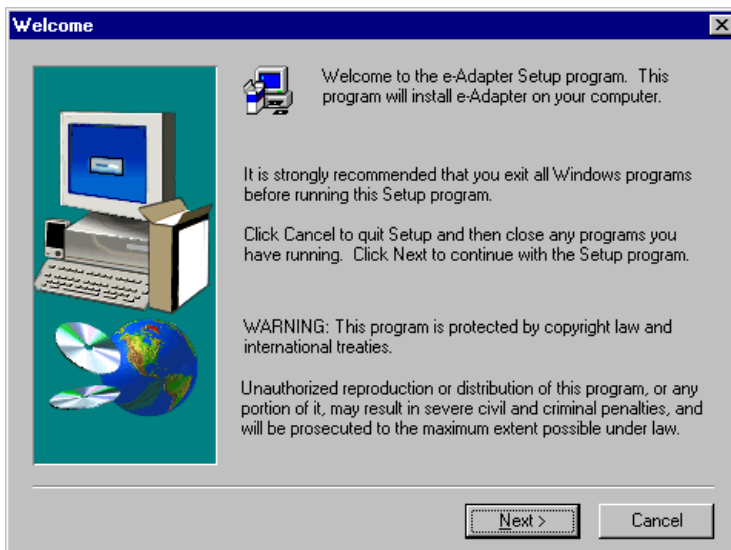


Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

The *Unpacking* TDE box appears and copies TDE files for installation into a temporary directory.

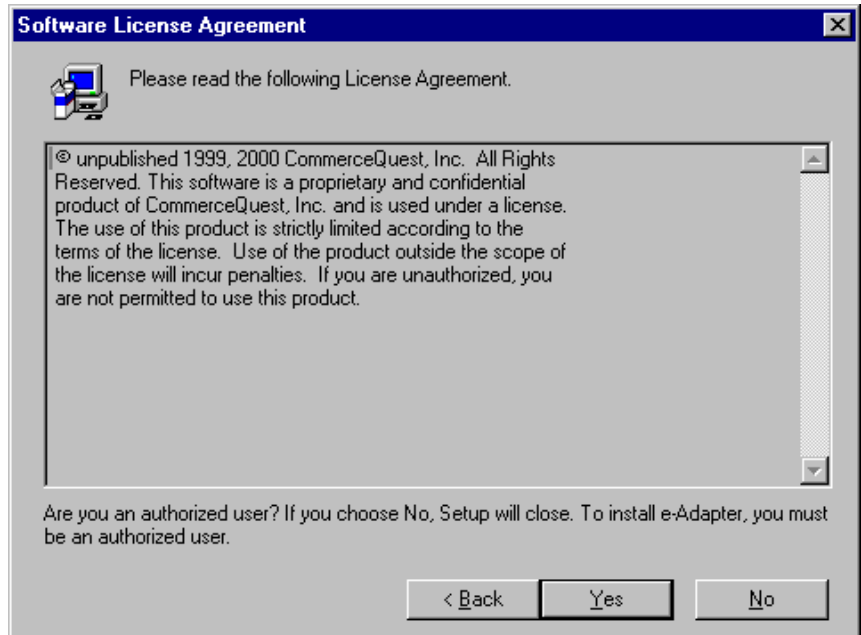
3. After the files are copied, the *Welcome* window appears. Exit all Windows programs and click **Next** to continue.



Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

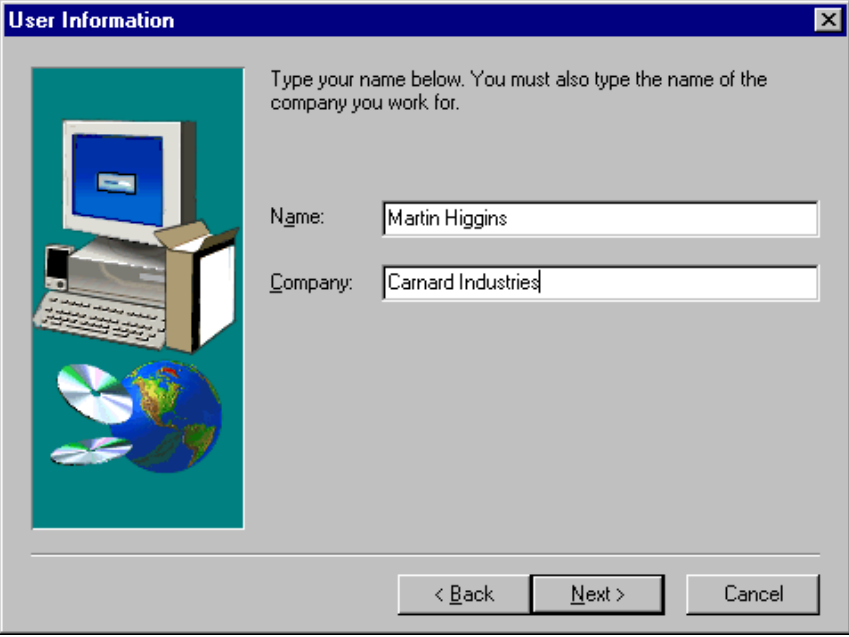
4. The *Software License Agreement* window appears. Read the license agreement. If you are not authorized to install the TDE software, click **No** to stop the installation and consult your system administrator. If you are authorized to install the TDE software, click **Yes** to continue.



Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

5. The *User Information* window appears. Type your name and your company's name. Click **Next** to continue.



The image shows a Windows-style dialog box titled "User Information". On the left is a graphic with a computer monitor, keyboard, and CD-ROMs. The main area contains the text: "Type your name below. You must also type the name of the company you work for." Below this are two text input fields. The first is labeled "Name:" and contains the text "Martin Higgins". The second is labeled "Company:" and contains the text "Carnard Industries". At the bottom right are three buttons: "< Back", "Next >", and "Cancel".

User Information

Type your name below. You must also type the name of the company you work for.

Name:

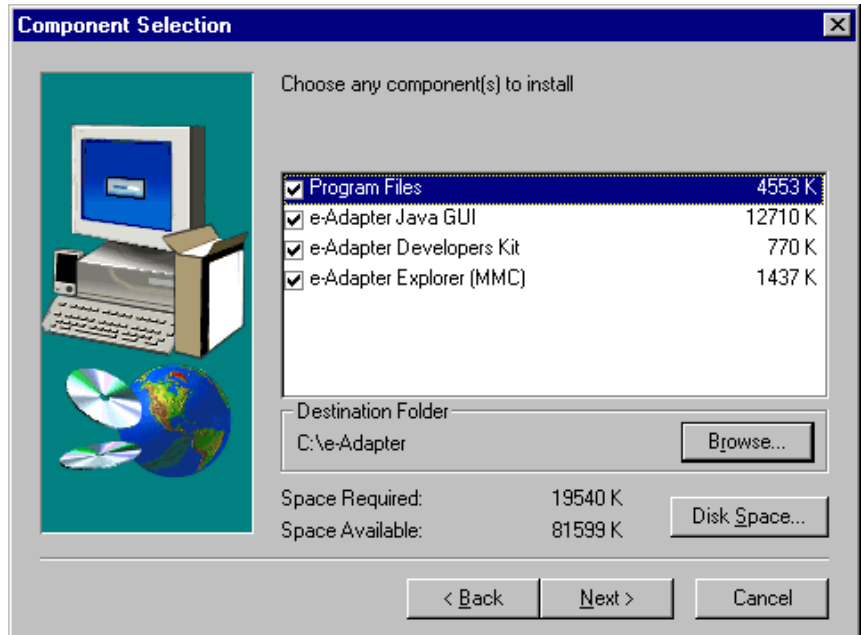
Company:

< Back Next > Cancel

Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

6. The *Component Selection* window appears. Click **Browse** to choose a destination folder other than the default or **Next** to continue



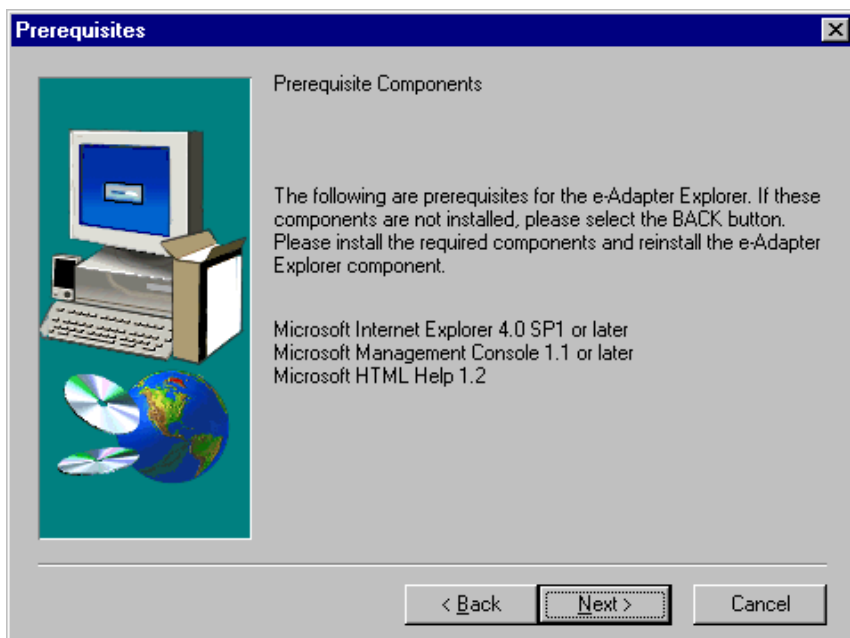
You can select from among the following components to install:

- **Program Files** – Interface commands and all configuration files.
- **TDE Java GUI** – The installation program, JRE 1.2, and all files required to run the GUI.
- **TDE Developers Kit** – Files that enable you to write your own programs using the FTF/MQ API.
- **TDE Explorer (MMC)** – TDE snap-in for the Microsoft Management Console.

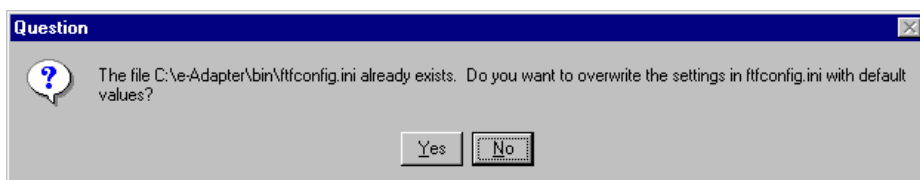
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

7. The **Prerequisites** window appears and lists software that must be installed prior to installing the TDE Explorer. Click the **Next** button to continue.



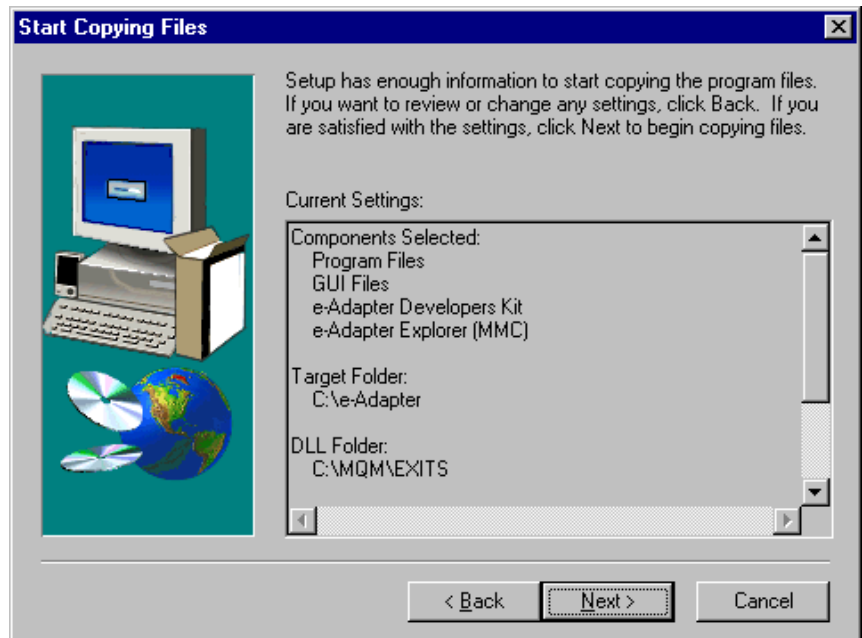
8. If TDE already exists on your system, the following window appears. If you want the values to be retained, click **No**, otherwise click **Yes**.



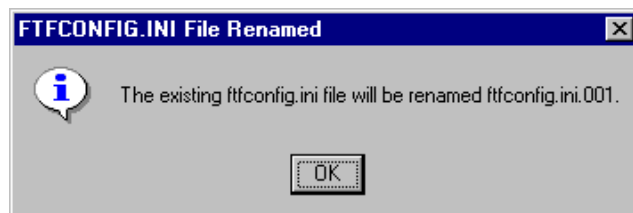
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

9. The *Start Copying Files* window appears and displays your installation choices. Review the choices to make sure they are appropriate, then click **Next** button to continue the installation.



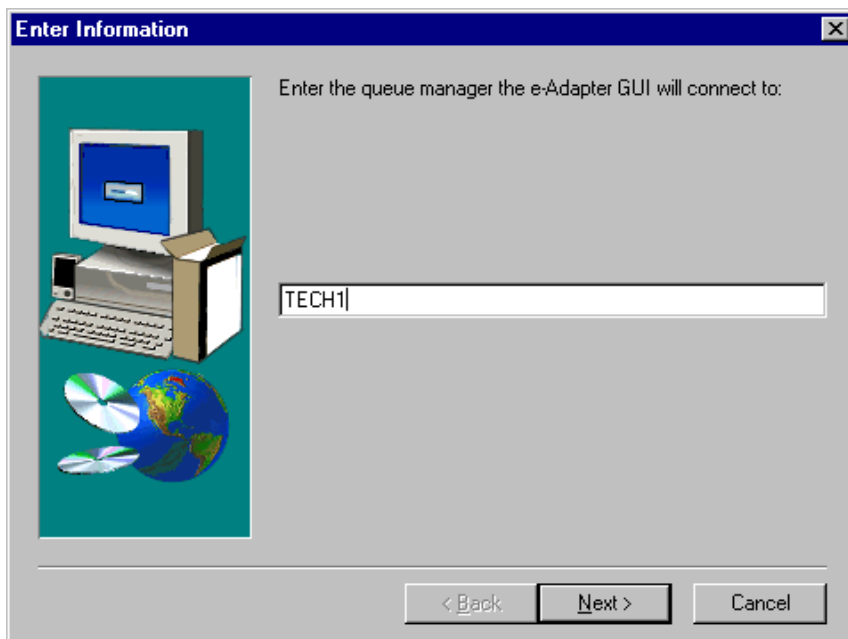
10. If you selected **Yes** on the window in step 8 to retain the values, TDE issues this message telling you that the existing filename will be changed to a new filename. Click **OK** to continue.



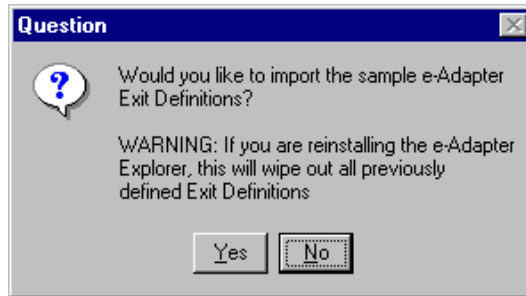
Note:

If the `ftfconfig.ini` or `ftf.ini` file already exists in the folder that you choose as your destination folder, you are given the option to keep the existing file or overwrite it. If you choose to overwrite it, InstallShield® gives the existing file a `.001` extension and overwrites the original file. You must configure the new file to match the old file.

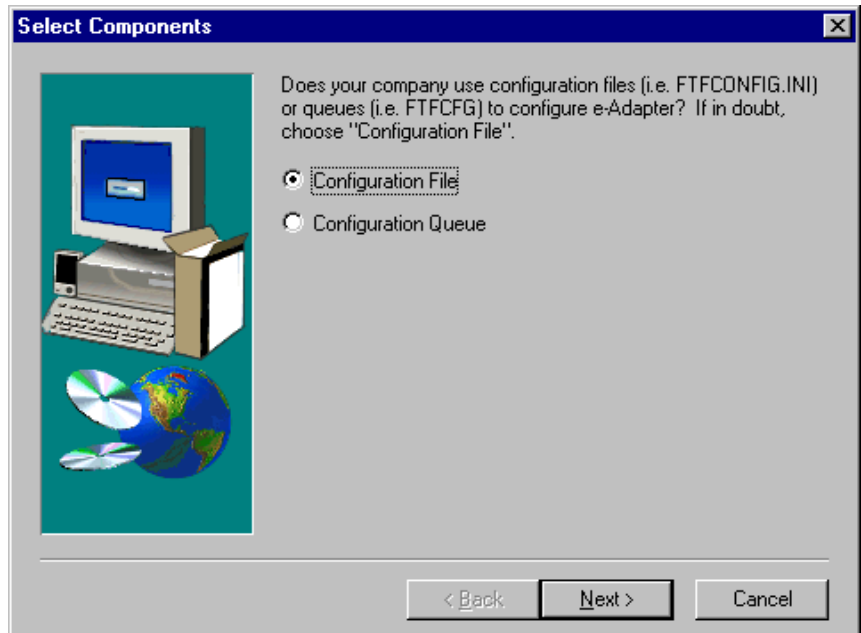
11. The **Enter Information** window appears. Type the name of the queue manager to which TDE will be connected. Click **Next** to continue.



12. This **Question** window asks about importing sample TDE Exit Definitions that are used by the TDE Explorer. If you already have Exit Definitions, click **No** so that your definitions are not overwritten. If you want to import the sample definitions, click **Yes**.



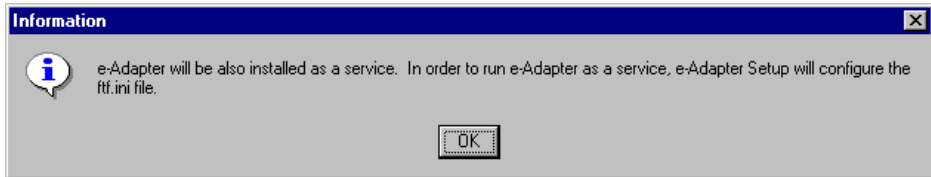
13. The **Select Components** window allows you to select a configuration file or configuration queue for storing the configuration parameters. If you are uncertain, select configuration file. Click **Next** to continue.



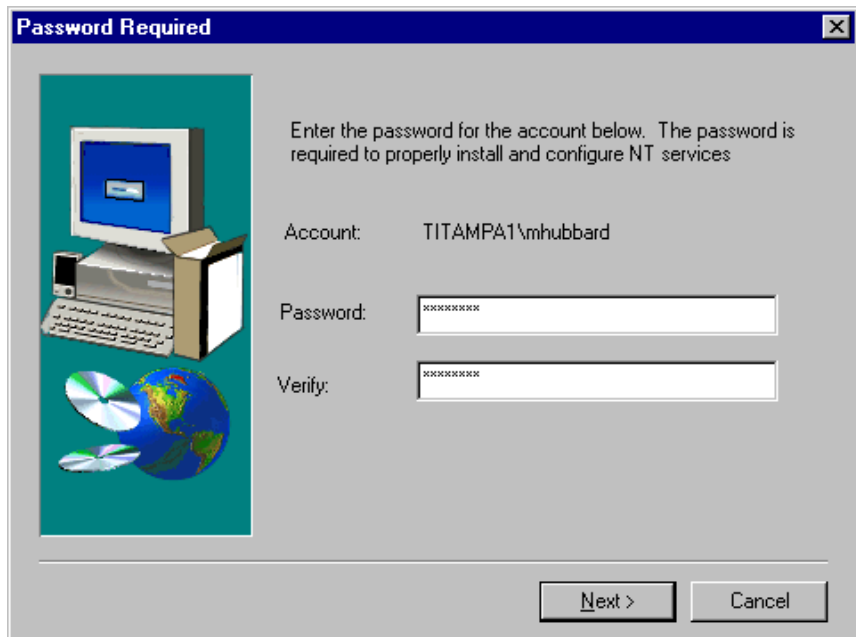
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

14. An *Information* window appears telling you that TDE can be run as a service. Click **OK** to continue.



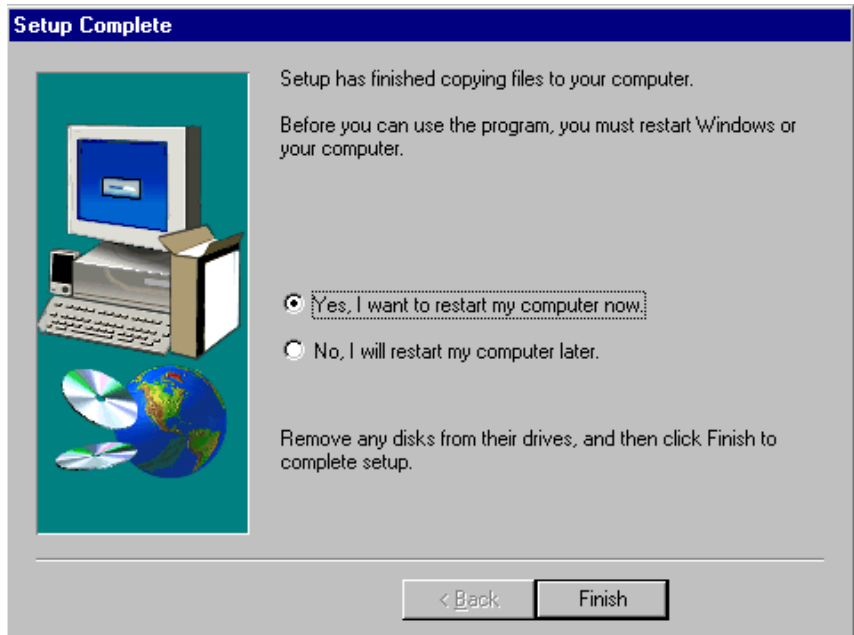
15. A *Password Required* window appears. Type and verify your Windows Win 32 password. This password is used to start the TDE service component. If you enter the password incorrectly, the TDE service component will not start. Click **Next** to continue.



Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

16. After the installation is complete, the *Setup Complete* window appears. Click **Finish** to complete the setup.



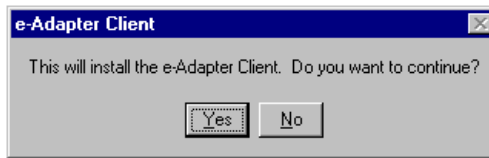
The TDE product is installed.

After TDE is installed on Win 32, you must reboot your system in order to update the TDE environment.

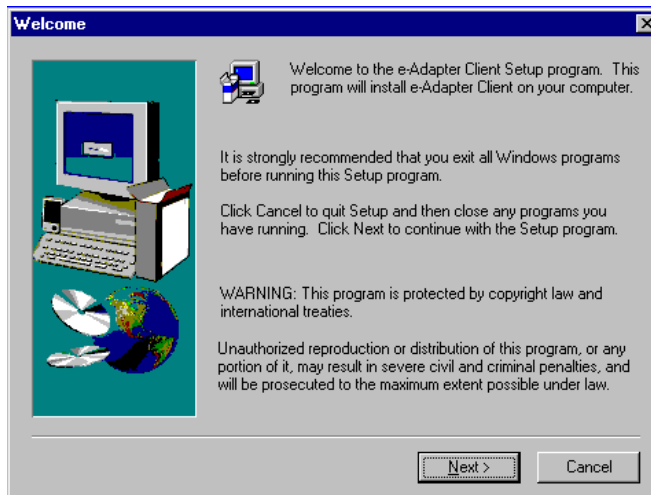
Installing the Tivoli Data Exchange Client

To install the client version of TDE in the Win 32 environment, follow these steps:

1. Run *Setupec.exe* in the client directory found in the installation CD's Win 32 directory.
2. The TDE *Client* dialog box appears. Click **Yes** to continue



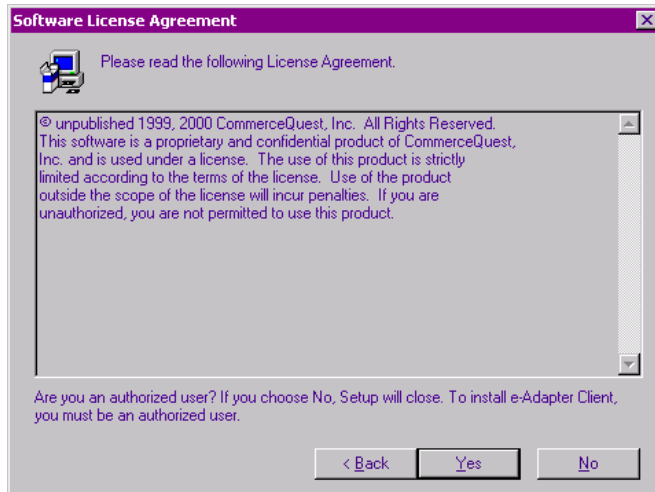
3. The *Unpacking this Package* box appears and copies TDE files for installation into a temporary directory.
4. After the files are copied, the *Welcome* window appears. Exit all Windows programs and click **Next** to continue.



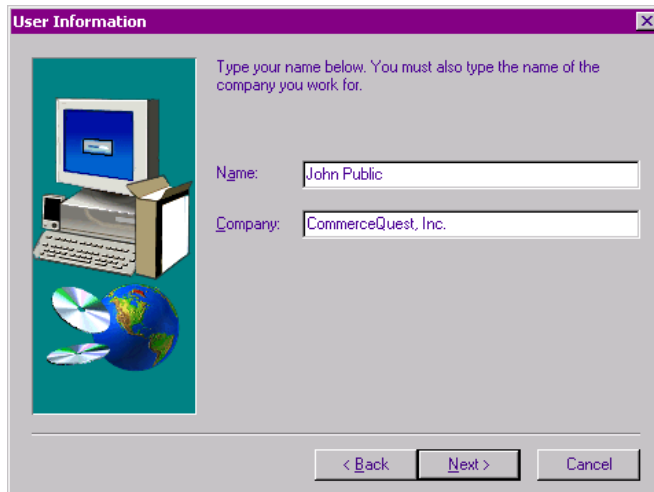
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

5. The *Software License Agreement* window appears. Read the license agreement. If you are not authorized to install the TDE software, click **No** to stop the installation and consult your system administrator. If you are authorized, click **Yes** to continue.



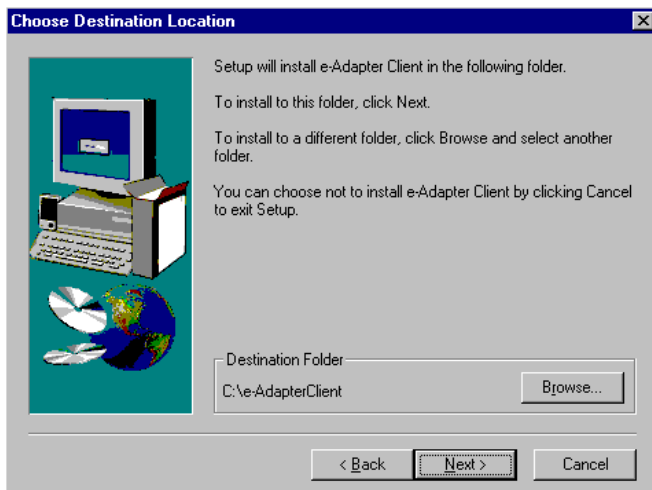
6. The *User Information* window appears. Type your name and your company's name. Click **Next** to continue.



Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

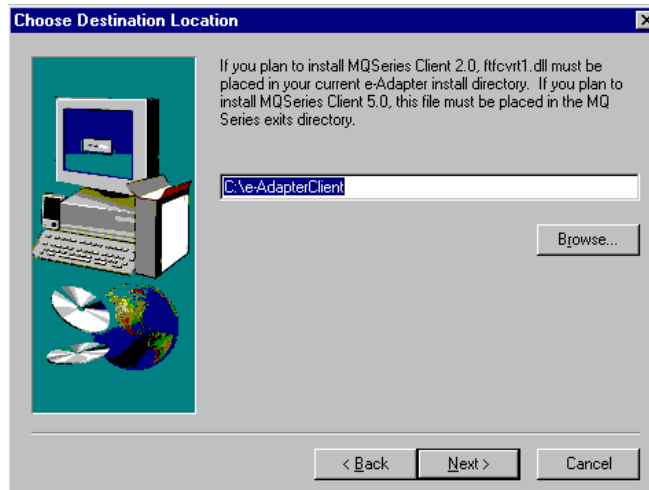
7. The *Choose Destination Location* window appears. If you do not want to use the default directory, specify an alternative. When you have selected the desired installation directory, click **Next** to continue.



Note:

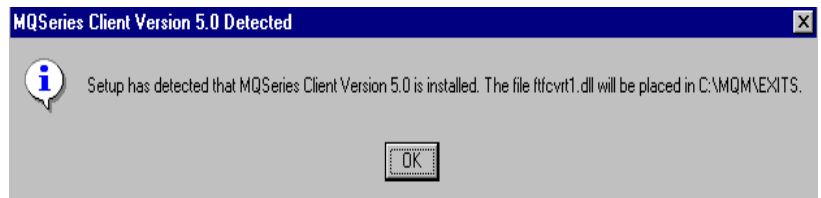
If the `ftfconfig.ini` or `ftf.ini` file already exists in the folder that you choose as your destination folder, you are given the option to keep the existing file or overwrite it. If you choose to overwrite it, InstallShield[®] gives the existing file a `.000` extension and overwrites the original file. You must configure the new file to match the old file.

8. The *Choose Destination Location* window appears directing you to install the DLL in the appropriate directory.



9. Specify the DLL location and click **Next** to continue.

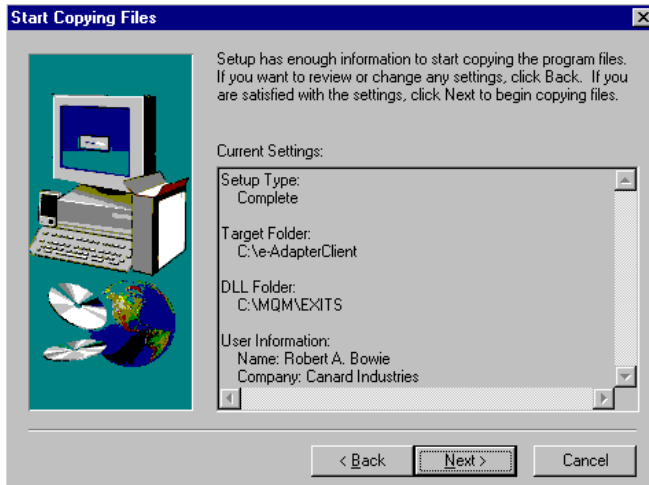
If you click **Yes** at step 8, InstallShield detects the version of MQSeries and provides you with information regarding the placement of the ftfcvrt1.dll file that is appropriate for your installation.



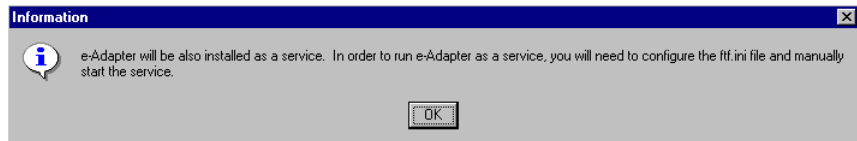
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

10. The *Start Copying Files* window appears and displays your installation choices. Review the choices and click **Back** to go back to the appropriate screen and modify the settings or **Next** to continue the installation.



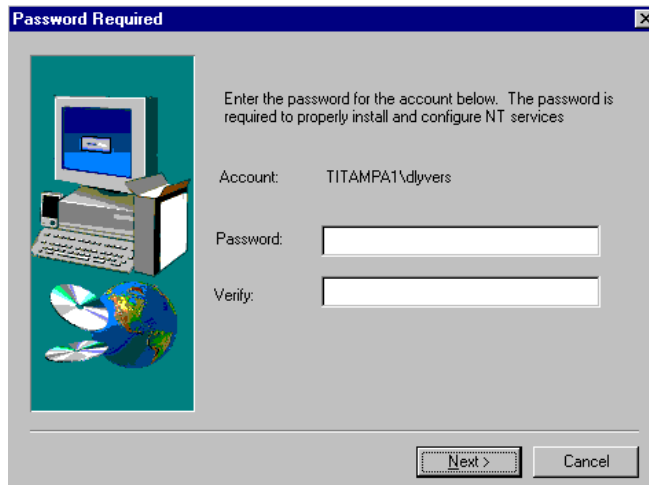
11. An *Information* window appears informing you that to run TDE as a service, you must configure the `ftf.ini` file and manually start TDE.



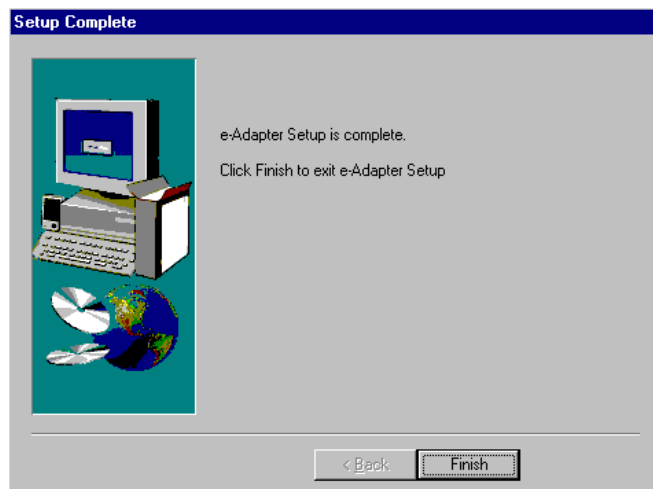
Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

12. The *Password Required* window appears. Type and verify your Win 32 password. This password is used to start the TDE service component. If you enter your password incorrectly, the TDE service component will not start. Click **Next** to continue.



13. If the TDE service is active on the installation machine, a dialog box appears and allows you to shut down the service or stop the installation.
14. After the installation is complete, the *Setup Complete* window appears. Click **Finish** to complete the setup and restart your machine.



Installing Tivoli Data Exchange on Win 32

Installing Tivoli Data Exchange

The TDE product is installed.

15. On the machine that runs the queue manager to which the TDE client is connecting, enter the following command to apply the `ftf.mqs` script:

```
runmqsc -e queueMgrName < ftf.mqs
```

Where *queueMgrName* is the name of the queue manager to which the client is connecting.

16. Define the connection from the MQSeries client to the MQSeries server. For more information, see the “MQSeries Clients” manual in the MQSeries documentation.

Configuring Clients

The startup parameters used to start the TDE Manager, TDE Sender, and TDE Receiver must be set up for client by using the `-nodename` argument. The `-nodename` argument accepts an identifier that will be used in the respective `-sqm` or `-dqm` arguments of the FTF request.

In this example, the FTF command’s `sqm` and `dqm` argument values refer to queue managers.

```
FTF -sqm MQM1 -dqm MQM2 -spath C:\FILE.1 -DPATH  
C:\FILE.2
```

In this example, the FTF command’s `sqm` and `dqm` argument values refer to clients. In this case, the FTF command is sending a file from an MQSeries client to another MQSeries client.

```
FTF -sqm CLIENTA -dqm CLIENTB -spath C:\FILE.1  
-DPATH C:\FILE.2
```


However, the ftf components on CLIENTA and CLIENTB must be started with the -nodename CLIENTA and -nodename CLIENTB, respectively. This name must be defined in the TDE configuration file at every node that wishes to participate with this client. The following figure displays a sample configuration file that establishes CLIENTA and CLIENTB.

```
FTFNodeOverride:
name=CLIENTA, CLIENTB

CLIENTA:
QueueManager=MQM1
ManagerControlQueue=A.FTFMGR.CONTROL
ManagerSyncQueue=A.FTFMGR.SYNC

SenderNumInstances=1
SenderStageControlQueue=A.FTFSDR.STAGE.CONTROL
SenderControlQueue=A.FTFSDR.CONTROL
SenderSyncQueue=A.FTFSDR.SYNC
SenderStageQueue=A.FTFSDR.STAGE
SenderMaxStageQueues=1
SenderSystemQueue=A.FTFSDR.SYSTEM
SenderCancel=NO

ReceiverNumInstances=1
ReceiverControlQueue=A.FTFRCV.CONTROL
ReceiverSyncQueue=A.FTFRCV.SYNC
ReceiverStageQueue=A.FTFRCV.STAGE
ReceiverSystemQueue=A.FTFRCV.SYSTEM
ReceiverCancel=NO

CLIENTB:
QueueManager=MQM1
ReceiverNumInstances=1
ReceiverControlQueue=B.FTFRCV.CONTROL
ReceiverSyncQueue=B.FTFRCV.SYNC
ReceiverStageQueue=B.FTFRCV.STAGE
ReceiverSystemQueue=B.FTFRCV.SYSTEM
ReceiverCancel=NO
```

Configuring the Tivoli Data Exchange Initialization File

The TDE initialization file is the key to using the TDE service. Its path and name are defined in the `FTF_IFILE` environment variable, which is set during installation. You can use any text editor to modify this file.

TDE includes a sample initialization file named `ftf.ini.sam`. This file resides in the `TDE\bin` directory. You can use this sample as the basis for your initialization file.

Note:

Although you can assign any filename and location to the TDE initialization file, you should use a consistent name and location on all machines for easier maintenance.

The TDE initialization file contains two sections:

- The `FTFStart` section
- The environment sections

The FTFStart Section

The `FTFStart` section determines the number of TDE instances to be started on the Win 32 machine and establishes names for each instance. One `FTFStart` section exists in each initialization file. Each instance specified in the `FTFStart` section should have a corresponding section in the environment sections.

The `FTFStart` section has the following syntax:

```
FTFStart:
    name = instance1, [instance2, ...instanceN]
```

Where *instance1*, *instance2*, and *instanceN* are the names of the TDE instances you are starting on the current machine. You must specify at least one name. If you specify more than one name, you must separate the names with commas.

The sample file's FTFStart section contains the following specifications:

```
FTFStart:
    name = production,test
```

In this case, two TDE instances are being started: *production* and *test*. Each TDE instance in your environment should be given a name and all names should be separated by commas.

The Environment Sections

The environment sections specify the following specific attributes for each TDE instance listed in the FTFStart section. It has the following syntax:

```
name:

lqm = localQueueMgr
cfile = configFile
nodename =
RetryCount = 3
RetryTimeout = 60
manager.commandline = mgrArgs
sender.commandline.1 = sdrArgs1
sender.commandline.2 = sdrArgs2
...
sender.commandline.N = sdrArgsN
receiver.commandline.1 = rcvArgs1
receiver.commandline.2 = rcvArgs2
...
receiver.commandline.N = rcvArgsN
```

Where:

- *name* is an instance name specified in the FTFStart section. Each instance name specified in the FTFStart section must contain its own section in the environment sections.
- *localQueueMgr* contains the name of the queue manager from which the startup commands are run. This value is required and it must refer to a valid MQSeries queue manager.

Installing Tivoli Data Exchange on Win 32

Configuring the Tivoli Data Exchange Initialization File

- *configFile* contains the name of the TDE configuration file for all components in the current instance. This value is required and must point to a valid configuration file for TDE to function correctly. It applies to all TDE components in the current instance. Each instance should have a different TDEconfiguration file.
- *nodename* is the nodename identifier for each component. This value is required.
- *RetryCount* is the number of times that the Win 32 TDE Service attempts to restart a component, after the component has failed, within the time specified by *retryTimeout*. The Win 32 TDE Service senses that a component has gone down and attempts to restart the component.
Default value: 3
- *RetryTimeout* is the number of seconds specified for the Win 32 TDE Service to continue to restart a component. **Default value: 60**
- *mgrArgs* contains the optional command-line arguments for starting the TDE Manager.
- *sdrArgs1*, *sdrArgs2*, and *sdrArgsN* contain the optional command-line arguments for each TDE Sender in the current instance.

Installing Tivoli Data Exchange on Win 32

Configuring the Tivoli Data Exchange Initialization File

- *rcvArgs1*, *rcvArgs2*, and *rcvArgsN* contain the optional command-line arguments for each TDE Receiver in the current instance.

Note:

You can specify the following optional command-line arguments:

- The *-lfile* argument, which creates a log file for the current component,
- The *-ofile* argument, which reads an options file for the current component.
- Command-line arguments for security exits. (For more information about user exits, see “Tivoli Data Exchange User Exits” in the *Tivoli Data Exchange User's Guide*.)

If you specify an options file, any command-line arguments you specify override the matching options file arguments.

If you specify command-line arguments for more TDE Senders or TDE Receivers than are specified in the TDE configuration file, the extra settings are ignored.

The sample file's first environment section contains the following attributes:

```
production:
lqm = PRODMQM
cfile = c:\ftfb\bin\ftfconfig.ini
manager.commandline = -lfile c:\ftf\mgr.out
sender.commandline.1 = -lfile c:\ftf\sdr.out1
sender.commandline.2 = -lfile c:\ftf\sdr.out2
receiver.commandline.1 = -lfile c:\ftf\rcv.out1
receiver.commandline.2 = -lfile c:\ftf\rcv.out2
```

The first line of the sample specifies that the attributes being listed are for the *production* environment. The second line determines the local queue manager setting. The third line lists the fully qualified path name and filename for the TDE configuration file. The additional lines specify command-line parameters for the TDE Manager, TDE Senders, TDE Receivers, and TDE status daemon. The information for each sender, receiver, and status daemon is distinguished by the numeric suffix at the end of each identifier (for instance, *receiver.commandline.1*, *receiver.commandline.2*, etc.). In each of the instances listed, the sample specifies only a log file as a command-line option.

Installing Tivoli Data Exchange on Win 32

Configuring the Tivoli Data Exchange Initialization File

```
FTFStart:
  name = prod

prod:
  lqm = PRODMQM
  cfile = c:\DIWin\bin\ftfconfig.ini
  nodename =
  RetryCount = 3
  RetryTimeout = 60
  manager.commandline    = -lfile
                        c:\DIWin\bin\mgr.out
  sender.commandline.1   = -lfile
                        c:\DIWin\bin\sdr.out1
  sender.commandline.2   = -lfile
                        c:\DIWin\bin\sdr.out2
  receiver.commandline.1 = -lfile
                        c:\DIWin\bin\rcv.out1
  receiver.commandline.2 = -lfile
                        c:\DIWin\bin\rcv.out2
```

Example

The following example shows configuration information for a sample of the TDE initialization file that ships with TDE. In the sample, two instances are specified (production and test). In the production instance, the lqm and configuration file values are set and command-line arguments are specified for the manager, two senders, and two receivers. In the test instance, the local configuration and configuration file values are set, but no command-line arguments are specified.

```
FTFStart:
  name = production, test

production:
  lqm = PRODMQM
  cfile = c:\ftfbin\ftfconfig.ini
  manager.commandline      = -lfile c:\ftf\mgr.out
  sender.commandline.1     = -lfile c:\ftf\sdr.out1
  sender.commandline.2     = -lfile c:\ftf\sdr.out2
  receiver.commandline.1   = -lfile c:\ftf\rcv.out1
  receiver.commandline.2   = -lfile c:\ftf\rcv.out2

test:
  lqm = TESTMQM
  cfile = e:\ftftest\testconfig.ini
```

Modifying the Tivoli Data Exchange Initialization File

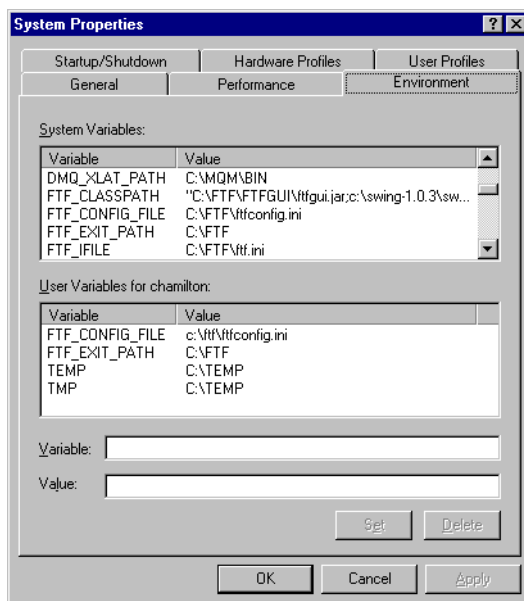
To modify the TDE initialization file, follow these steps:

1. Use a text editor to open the sample TDE initialization file (ftf.ini.sam).
2. Change the values listed in this file as appropriate for your environment.
3. Save the file. Make note of its path and name.

Setting Environment Variables

Environment variables specify values needed for TDE to run. To change an environment variable in this file, follow these steps:

1. Start the **Control Panel**.
2. Double-click the **System** icon. The System Properties window appears.
3. Click the **Environment** tab. The Environment tab page appears



4. Click the environment variable. Change the values in the *Variable* and *Value* fields to the values listed in the following table.

Variable	Value
FTF_IFILE	The path and name of the TDE initialization file you previously created and saved.
FTF_CONFIG_FILE	Must be set to the name of the TDE configuration file with the fully qualified path.
FTF_EXIT_PATH	Must be set to the fully qualified path of the directory in which the TDE exits (shared libraries) exist.
MQSERVER	Use this environment variable to point to the queue manager where the FTFSTAT Control and Detail queues reside. The default is to point to the Queue Manager on the Local machine. There must be a TCP Listener running.

5. Click the **Set** button.
6. Click the **OK** button.
7. Restart the computer.

Restarting the Service

When you stop and restart the TDE service, it stops and restarts all TDE components in all instances listed in the TDE configuration file. If any components were started from the command line, they are stopped when you stop the service, but they are not restarted.

Moving TDE Executables

When you start the TDE service, the service looks in the installation directory for the startup executables. If it does not find them, it looks in each directory listed in the PATH environment variable. If you move the TDE executables to another directory, make sure that directory is included in the PATH variable.

Installing Tivoli Data Exchange on Win 32

Restarting the Service

Installing Tivoli Data Exchange on UNIX

This chapter describes how to install and configure Tivoli Data Exchange (TDE) on each of its supported UNIX platforms. TDE currently runs on any of the following UNIX operating systems:

- AIX version 4.2 or higher
- HP-UX version 10.20 or 11.x
- Solaris version 2.6 or higher
- DEC UNIX version 4.0
- Tru64 UNIX 5.1
- NCR 3.0
- Linux RedHat 7.0
- MQSeries version 5.0 or higher

This chapter includes the following sections:

Section	Page
Overview	84
System and Installation Requirements	84
UNIX Installation Steps	85
Initializing on Tivoli Data Exchange UNIX	90

Assumptions

This chapter makes the following assumptions:

- You have installed the appropriate version of MQSeries before you attempt to install TDE.
- You have appropriate permission to install software on the installation machine.

Overview

To install TDE on a UNIX system, perform the following steps:

1. Make sure the system and installation requirements have been satisfied (see “System and Installation Requirements” on page 84).
2. Install TDE from the installation CD (see “UNIX Installation Steps” on page 85).
3. Initialize TDE (see “Initializing on Tivoli Data Exchange UNIX” on page 90).

System and Installation Requirements

In addition to one of the UNIX operating systems previously listed, the following software products must be installed before you can install TDE on a UNIX system:

- MQSeries, version 2.1, 5.0 or higher

To run the TDE GUI on Solaris, the following Java-related software is required:

- Java Development Kit (JDK), version 1.1.6 with Swing 1.1
- Or -**
- Java Runtime Environment (JRE) which is included in the TDE installation, and Swing 1.1

You can download the latest versions of the JDK from the Sun Products and APIs web page (www.javasoft.com/products/index.html).

Other UNIX systems, at this point, cannot run the TDE GUI.

The following conditions must exist before you can install TDE on a UNIX system:

- You must have the proper authority to install TDE in the desired directories.
- The appropriate version of MQSeries must be operational.

No group or user is required for the installation. However, at run time, TDE will use MQSeries, which requires access to the *mqm* group.

UNIX Installation Steps

To install TDE on UNIX, follow these steps:

1. Insert the installation CD.
2. Change to the appropriate directory on the installation CD for your version of the UNIX operating system.
3. Change to the server directory.
4. Type **./install.sh** to install the base TDE product. With this installation you can issue TDE commands from the command line. If you running on Solaris and want to install the TDE GUI also, type **./instgui.sh** after you have completed the base installation procedure.

Note:

Press **[Ctrl+C]** at any point in the installation process to exit the installation script.

5. When you are prompted, enter the fully qualified TDE installation directory.
6. When you are prompted, enter the fully qualified installation path for FTFCVRT1. The following table lists the appropriate locations for these shared libraries for each supported version of MQSeries.

MQSeries Version	Required Shared Library Location
2.1	TDE installation directory
5.0 or 5.1	MQSeries exits directory

Connecting a TDE Client to a UNIX Server

When you connect an TDE client to a UNIX server, copy the FTFCVRT1_r Shared Library to one of the following locations:

If you are running MQSeries, version 2.1, copy it to the TDE installation directory.

If you are running MQSeries, version 5.0 or 5.1, copy it to the MQSeries exits directory (default value: /var/mqm/exits)

If you do not copy this Shared Library, an MQSeries 2110 error may occur and the TDE Manager or the TDE Receiver may terminate.

7. If the directory does not exist, a prompt appears and asks if you want to create it. To create the directory, enter y.

Note:

The installation script installs TDE in a default location. If you do not use the default location, change the ftfgui shell script to match the installation location of TDE.

8. Add the following environment variables:
 - **FTF_CONFIG_FILE** – Must be set to the name of the TDE configuration file with the fully qualified path.
 - **FTF_EXIT_PATH** – Must be set to the fully qualified path of the directory in which the TDE exits (shared libraries) exist.
9. Enter the following command to apply the ftf.mqs script file to the appropriate queue manager:

```
runmqsc -e queueMgr < ftf.mqs
```

Where *queueMgr* is the name of the queue manager against which the script is being run.

Configuration for the switch user feature

This feature enables TDE to switch the user account that the processes are running under to the requestor's user ID in order to allow the UNIX system to check the requestor's write file access. By switching to the requestor's ID, the file is then written by that requestor's ID.

To configure Tivoli Data Exchange to be able to switch to the requestor's user ID the following must be done. The ownership of the `ftfunxio` executable must be changed to root and the "set UID" bit must be set. It is advisable to create an FTF group and add only the user account which the TDE processes will be started under to that group. This will prevent any users other than root or the appropriate FTF user account from executing this process. Run the following commands to accomplish this.

```
>chown root ftfunxio
>chgrp ftf ftfunxio
>chmod 4710 ftfunxio
```

The permissions of the executables should look like the following:

```
-rws--x--- 1 root ftf 1138688 Mar 29 16:43 ftfunxio
```

This allows Tivoli Data Exchange to switch the real user of the process while not running under the root user ID.

If an `ftf` group is not created then use the mode 4711 for the `chmod` command.

Note

Matching user ID's must be created in lowercase for systems that typically have uppercase user IDs, MVS, AS/400, etc. Create the same user ID on UNIX in lowercase.

To utilize the switch user (`-su`) feature the following must be done during installation of Tivoli Data Exchange.

- The switch user feature will switch the user account, which is running Tivoli Data Exchange to that of the requestor.
- The switch user change will create the target file as the requestor ID.

Installing Tivoli Data Exchange on UNIX

UNIX Installation Steps

This feature is available on UNIX only.

Installing the GUI

To install the TDE GUI on UNIX, follow these steps:

1. Insert the installation CD.
2. Change to the appropriate directory on the installation CD for your version of the UNIX operating system.
3. Change to the client or the server directory under the gui directory, as appropriate.
4. Type `./instlgui.sh`

Note:

Press **[Ctrl+C]** at any point in the installation process to exit the installation script.

5. When you are prompted, enter the fully qualified TDE GUI installation directory.
6. If the directory does not exist, a prompt appears and asks if you want to create it. To create the directory, type **y**.
7. Add the following environment variable:
 - **FTF_EXIT_PATH** – Must be set to the fully qualified path of the directory in which the TDE exits (shared libraries) exist.

Configuring the Tivoli Data Exchange GUI

To configure the TDE GUI for operation on UNIX, follow these steps:

1. Open the TDE GUI script file. This file contains environment variable settings required to run the TDE GUI. It also starts the TDE GUI.
2. Set or append the following environment variables:
 - **TDE_GUI_HOME** – Must contain the directory into which you installed TDE. **Example:** \$HOME/FTFGUI_240
 - **PATH** – Must contain \$FTFGUI_HOME.
Example: \$PATH:\$FTFGUI_HOME
 - **FTF_CLASS_PATH** – Must contain the location of the ftfgui.jar file, the location of the JDK, and the location of the Swing class libraries.
Example: \$FTFGUI_HOME/ftfgui.jar:/usr/java/lib/classes.zip:/opt/swing-1.1/swingall.jar
 - **FTF_CONFIG_FILE** – Must contain the fully qualified path and filename of the TDE configuration file.
Example: \$HOME/FTFGUI/ftfconfig.ini
 - **TDE_GUI_LIBRARY_PATH** – Must contain the location of the ftfgui library. **Example:** \$HOME/FTFGUI_HOME
 - **LD_LIBRARY_PATH** – Must contain \$FTFGUI_LIBRARY_PATH.
Example: \$LD_LIBRARY_PATH:\$FTFGUI_LIBRARY_PATH

Also set the following environment variable for the ftfgui client:

- **MQSERVER** – Must contain ChannelName/TransportType /ConnectionName. **Example:** FTF.CLIENT/TCP/IP ADDRESS

Include the port number in the IP address if you do not use the default port of 1414 [e.g., 192.168.100.150(1491)].

3. Save the file.

Installing Tivoli Data Exchange on UNIX

Initializing on Tivoli Data Exchange UNIX

4. Use MQSeries to add a SVRCONN channel (system.def.svrconn) to the server machine. For example:

```
2 : dis chl(SYSTEM.DEF.SVRCONN)all
AMQ8414: Display Channel details.
CHANNEL(SYSTEM.DEF.SVRCONN) CHLTYPE(SVRCONN)
TRPTYPE(TCP) DESCR( )
SCYEXIT( ) MAXMSGL(4194304)
SCYDATA( ) HBINT(300)
MCAUSER( ) ALTDATE(2000-05-23)
ALTTIME(14.36.15)
SENDEXIT( )
RCVEXIT( )
SENDDATA( )
RCVDATA( )
```

Executing the ftfgui Script

At the command prompt, enter the following command:

```
ftfgui qMgrName
```

Where *qMgrName* is the name of the queue manager from which you are monitoring.

In the following example, the TDE GUI is started and monitors the queue manager named PROD11A.

```
ftfgui PROD11A
```

Initializing on Tivoli Data Exchange UNIX

This section contains the following information:

- Starting the TDE Manager (page 91)
- Starting the TDE Sender (page 92)
- Starting the TDE Receiver (page 93)

Starting the Tivoli Data Exchange Manager

To start the TDE Manager, enter the following command:

Using a configuration file:

```
ftfmgr -lqm localQMgr -cfile configFile  
-lfile logFile
```

Using a configuration queue:

```
ftfmgr -lqm localQMgr -cq configQueueName  
-lfile logFile
```

Where:

- *localQMgr* determines the queue manager from which the FTFMGR command is issued. If it is not specified, the command connects to the default queue manager that is set in the MQSeries configuration.

Otherwise, whenever a command or interface starts up it tries to connect to the lqm. If an lqm value is not specified, the interface attempts to connect to the specified default queue manager on platforms where MQSeries supports them.

- *configFile* contains the fully qualified path and filename for the TDE configuration file. If this value is not set, it defaults to the value specified in the FTF_CONFIG_FILE environment variable. For more information about this environment variable, see “UNIX Installation Steps” on page 85.

Installing Tivoli Data Exchange on UNIX

Initializing on Tivoli Data Exchange UNIX

- *configQueueName* – Displays the queue from which the configuration information is to be retrieved for this TDE instance on this node. The -cq argument points TDE to the queue name rather than to the standard configuration file. You cannot specify both a -cfile and a -cq argument in the same command.

Configuration File and Queue Order of Precedence

If you do not specify either a configuration file or a configuration queue, TDE checks the FTF_CONFIG_QUEUE environment variable and uses the specified queue. If this environment variable is not set, TDE checks the FTF_CONFIG_FILE environment variable and uses the specified file. If neither environment variable is set and no command-line argument is set, the command fails.

- *logFile* contains the fully qualified path and filename for the TDE Manager log file. If you do not specify a log file, output is sent to the monitor.

Starting the Tivoli Data Exchange Sender

To start the TDE Sender, enter the following command:

Using a configuration file:

```
ftfsdr -lqm localQMgr -cfile configFile  
-lfile logFile
```

Using a configuration queue:

```
ftfsdr -lqm localQMgr -cq configQueueName  
-lfile logFile
```

Where:

- *localQMgr* determines the queue manager from which the FTFSDR command is issued. If it is not specified, the command connects to the default queue manager that is set in the MQSeries configuration.

Otherwise, whenever a command or interface starts up it tries to connect to the lqm. If an lqm value is not specified, the interface attempts to connect to the specified default queue manager on platforms where MQSeries supports them.

- *configFile* contains the fully qualified path and filename for the TDE configuration file. If this value is not set, it defaults to the value specified in the FTF_CONFIG_FILE environment variable. For more information about this environment variable, see “UNIX Installation Steps” on page 85.
- *configQueueName* – Displays the queue from which the configuration information is to be retrieved for this TDE instance on this node. The -cq argument points TDE to the queue name rather than to the standard configuration file. You cannot specify both a -cfile and a -cq argument in the same command.

Configuration File and Queue Order of Precedence

If you do not specify either a configuration file or a configuration queue, TDE checks the FTF_CONFIG_QUEUE environment variable and uses the specified queue. If this environment variable is not set, checks the FTF_CONFIG_FILE environment variable and uses the specified file. If neither environment variable is set and no command-line argument is set, the command fails.

- *logFile* contains the fully qualified path and filename for the TDE Sender log file. If you do not specify a log file, output is sent to standard output.

Starting the Tivoli Data Exchange Receiver

To start the TDE Receiver, enter the following command:

Using a configuration file:

```
ftfrcv -lqm localQMGr -cfile configFile  
-lfile logFile
```

Using a configuration queue:

Installing Tivoli Data Exchange on UNIX

Initializing on Tivoli Data Exchange UNIX

```
ftfrcv -lqm localQMGr -cq configQueueName  
-lfile logFile
```

Where:

- *localQMGr* determines the queue manager from which the FTFRCV command is issued. If it is not specified, the command connects to the default queue manager that is set in the MQSeries configuration.

Otherwise, whenever a command or interface starts up it tries to connect to the lqm. If an lqm value is not specified, the interface attempts to connect to the specified default queue manager.

- *configFile* contains the fully qualified path and filename for the TDE configuration file. If this value is not set, it defaults to the value specified in the FTF_CONFIG_FILE environment variable. For more information about this environment variable, see “UNIX Installation Steps” on page 85.
- *configQueueName* displays the queue from which the configuration information is to be retrieved for this TDE instance on this node. The -cq argument points TDE to the queue name rather than to the standard configuration file. You cannot specify both a -cfile and a -cq argument in the same command.

Configuration File and Queue Order of Precedence

If you do not specify either a configuration file or a configuration queue, TDE checks the FTF_CONFIG_QUEUE environment variable and uses the specified queue. If this environment variable is not set, TDE checks the FTF_CONFIG_FILE environment variable and uses the specified file. If neither environment variable is set and no command-line argument is set, the command fails.

- *logFile* contains the fully qualified path and filename for the TDE Receiver log file. If you do not specify a log file, output is sent to the monitor.

Installing Tivoli Data Exchange on OpenVMS

This chapter describes how to install and configure Tivoli Data Exchange (TDE) on OpenVMS for VAX and Alpha and includes the following sections:

Section	Page
System and Installation Requirements	95
Installing on OpenVMS	96
Starting Tivoli Data Exchange on OpenVMS	97
Invoking a Tivoli Data Exchange Exit on OpenVMS	99

Assumptions

This chapter makes the following assumptions:

- You have either installed or will install the appropriate version of MQSeries before you attempt to install TDE.
- You have appropriate permission to install software on the installation machine.

System and Installation Requirements

The following minimum software levels must be installed before you can install TDE on an OpenVMS system:

- Alpha, Versions 6.2, 7.1, or 7.2

- VAX, Version 6.2

Installing on OpenVMS

Note:

TDE should be installed using the system account. If this is an upgrade, save the FTF.MQS file or do not purge old files during install.

Complete the following steps:

1. If installing by FTP, copy the TDE041.ZIP file to a temporary working directory.
2. Unzip the installation kit by running the following:

`unzip TDE041.ZIP`

This will produce the file TDE041.A.

3. Run the following: `@sys$update:vmsinstal`
4. There are a series of questions to answer:
 - Are you satisfied with the backup of your system disk?
 - Where is the distribution volume?

(Enter the disk and directory for the KIT location.

Example: DKA0:[KITS])

 - Enter the products to be processed:

Example: TDE041

5. After the installation completes, you are asked for more products to install. Press **Return** twice or type **CTL+Z** to exit.

Starting Tivoli Data Exchange on OpenVMS

Note:

If TDE was not installed in the common working area, the `cqdi_logicals.com` file must be edited accordingly.

Perform the following steps to start TDE on an OpenVMS system:

1. Edit `systartup_vms.com` and add the following line:

```
$ @sys$startup:cqdi_logicals.com
```

2. Edit `sylogin.com` and add the following line:

```
$ @sys$startup:cqdi_symbols.com
```

3. The TDE default queues must be defined prior to starting the TDE components. The sample file `ftf.mqs` contains the queue definitions needed for TDE. Add the following line at the beginning of the file:

```
$runmqsc
```

4. To define the queues, the MQSeries queue manager must be running.

Run `@ftf.mqs` to define the queues.

Note:

The logical `FTF_CONFIG_FILE` is defined in the `cqdi_logicals.com` file. This defines the TDE configuration and the default queues. See the *Tivoli Data Exchange User's Guide* for more information on the `FTF_CONFIG_FILE` and the `ftfconfig.ini` file.

Installing Tivoli Data Exchange on OpenVMS

Starting Tivoli Data Exchange on OpenVMS

5. The TDE_STARTUP.COM_SAMPLE file located in SYSHLP.EXAMPLES.TDE shows how to start the TDE components as detached processes. It executes three command files DI_FTFMGR_START.COM, DI_FTFSDR_START.COM, and DI_FTFRCV_START.COM.

Change the sample versions of these files by altering the name of the local queue manager to which the TDE processes connect. Remove the _SAMPLE extensions and copy the files to sys\$startup. This is where TDE_STARTUP.COM looks for them. Rename DI_STARTUP.COM_SAMPLE to DI_STARTUP.COM.

6. To start the TDE processes run the following command:

```
@SYS$STARTUP:TDE_STARTUP
```

7. Type **show system** to verify the processes are running. The TDE processes are FTFMGR, FTFSDR, and FTFRCV. By default the TDE log files are written to TDE_ROOT_DIR:[DI.LOGS].

Note:

When specifying an MQSeries object as a TDE argument, the object name must be enclosed in double quotes to retain its case. Example: FTFPING -LQM "VMS62QM" Where VMS62QM is the name of the MQSeries Queue Manager.

Determining Successful FTF Commands

On the OpenVMS operating system, a successful operation is indicated by the low bit of the status return being set (i.e., an odd-valued return status). Therefore, the return value of 1 at the OpenVMS Digital Command Language (DCL) level indicates success. You can also reference the numeric value of the symbol SS\$_NORMAL, a systemwide status value that indicates a successful operation.

The following example DCL command procedure tests the return value of a BACKUP operation. You can use a similar fragment to determine if an FTF command is successful.

Installing Tivoli Data Exchange on OpenVMS

Starting Tivoli Data Exchange on OpenVMS

```
$ BACKUP/IMAGE INDISK: OUTTAPE:INDISK.BCK/SAV
$
$ IF $STATUS
$ THEN
    $ WRITE SYS$OUTPUT "The BACKUP operation was successful."
$ ELSE
    $ WRITE SYS$OUTPUT "The BACKUP operation FAILED!"
$ DIF
```

Troubleshooting MQSeries on OpenVMS

There are a series of system parameters that may need to be modified for MQSeries to run properly. These are GBLSECTIONS, GBLPAGES, and GBLPAGFIL. The values suggested in the *MQSeries for Digital OpenVMS V2R2 System Management Guide* seem to be too low. In addition the user accounts that start the TDE components and MQSeries need sufficient paging file quotas.

Invoking a Tivoli Data Exchange Exit on OpenVMS

To invoke a TDE exit, the following sample command line can be used:

```
ftf -lqm "ALPHA2QM" -dqm "ALPHA2QM" -spath
sys$startup:text.txt -dpath cqdi_root_dir:test.txt -exit 8 -exitdll
FTFEX78 -exitEntry "FTFRcvPost8" -exitData "$copy
cqdi_root_dir:test.txt cqdi_root_dir:test_good.dat"
```

The -exitdll argument must contain only the filename. The file type cannot be indicated. If any file specification punctuation character is included, an error is returned.

Refer to the *Tivoli Data Exchange User's Guide*, "User Exits," for detailed information on this topic.

File Type Support on OpenVMS

Currently, TDE has no support for all the various file types on OpenVMS, except when type=BINARY on a TDE transfer. Designating the type as binary results in sequential, fixed-length (512 bytes) record files. This is used for transferring executables. The default value type=TEXT is used for all other TDE transfers and results in sequential, stream_LF record files, which are variable in length.

Installing Tivoli Data Exchange on OS/400

This chapter describes how to install and initialize Tivoli Data Exchange (TDE) on OS/400. It contains the following sections:

Section	Page
Overview	102
System and Installation Requirements	102
OS/400 Installation Steps	102

Assumptions

This chapter makes the following assumptions:

- MQSeries has been installed and properly configured on the installation machine.
- You have appropriate permission to access the OS/400 system and to install software on the installation machine.

Overview

To install TDE on an OS/400 system, perform the following steps:

1. Make sure the system and installation requirements have been satisfied (see “System and Installation Requirements” on page 102).
2. Install TDE from the installation CD (see “OS/400 Installation Steps” on page 102).
3. Initialize TDE (see “Initializing Tivoli Data Exchange on OS/400” on page 104).

System and Installation Requirements

The following software products must be installed before you can install TDE on an OS/400 system:

- OS/400, version 4 or higher
- MQSeries, version 4.2.1 or 5.1

OS/400 Installation Steps

To install TDE on OS/400, follow these steps:

1. Create a *SAVF file on your OS/400 that has the same name as the *SAVF file in the OS/400 directory on the installation CD.
2. FTP the *SAVF file in binary mode.
3. Use the DSPSAVF command to view the library saved information. This allows you to determine the name of the saved libraries. For example, if you created a library called FTF and placed the save file in that library, the command would be DSPSAVF FILE (FTF/FTF). The output would be as follows:

Display Saved Objects - Save File

Library saved :	FTF	Release level :	V3R7M0
ASP :	1	Data compressed . . . :	No
Save file :	FTF	Objects displayed . . :	6
Library :	FTF	Objects saved :	63
Records :	21976	Access paths :	0
Save command :	SAVLIB		

```
Save active . . . . : *NO
Save date/time . . . : 08/24/00 15:34:47
```

4. To restore the save file, use the RSTOBJ command. The SAVLIB parameter is the library saved information you retrieved in the previous step. For example, if you were restoring the objects to library FTF and the save file was called FTF, the command would be as follows:

```
RSTOBJ OBJ(*ALL) SAVLIB (XXXXX) DEV(*SAVF)
SAVF (LIB/FTF) RSTLIB (FTF)
```

where SAVLIB(XXXXX) is the value of the library saved from displaying the save file.

Note:

For illustration purposes to install TDE, we have used FTF as an example library in the following steps for where to install the software.

5. For user access, add the library FTF to the User Library List or to a System Library List. Set permissions according to the user's environment.
6. Use STRMQMMQSC command and the ftf.mqs file to create TDE queues. The MQS file matches the provided TDE configuration file.

If you are using MQSeries 5.1 with multiple queue managers defined on the system, create TDE queues against the correct queue managers.

7. Use the following CRTJOBQ command to create a JOB queue in the library FTF:

```
CRTJOBQ JOBQ (FTF/FTFV4)
```

8. Use the following ADDJOBQE command to add a JOB queue entry to an active subsystem (e.g., QBATCH):

```
ADDJOBQE SBSD (QBATCH) JOBQ (FTF/FTFV4)
MAXACT (*NOMAX)
```

9. Use the following CRTJOBQ command to create a JOB description (JOBQ) in the library FTF:

```
CRTJOBQ JOBQ (FTF/FTFJOBQ) JOBQ (FTF/FTFV4)
```

Initializing Tivoli Data Exchange on OS/400

The TDE Manager, Sender, Receiver, Logger, and Status Daemon are run as batch jobs. When the correct menu option is selected, the jobs are submitted to the job queue (JOBQ) you created using the CRTJOBQ command and use the job description (JOBQ) you created using the CRTJOBQ command.

Starting an OS/400 Session

When you start a session in OS/400, enter the following command at the command prompt:

```
STRCMTCTL LCKLVL (*ALL) CMTSCOPE (*JOB)
```

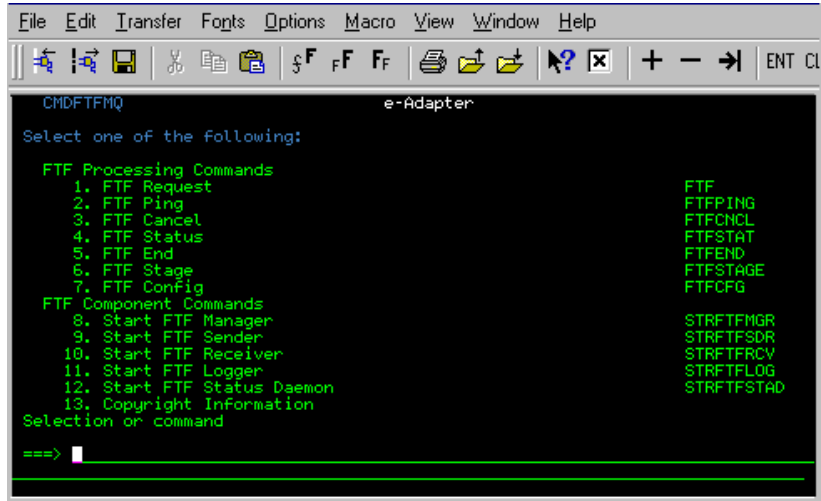
This command lasts for the entire TDE session and allows you to successfully run all e-TDE commands.

Note:

If you are using MQSeries 5.1 with multiple queue managers defined on the system, you must have defined TDE queues against correct queue managers in step 5 of the installation.

Using the OS/400 Tivoli Data Exchange Menu

On OS/400, the TDE menu is used to run TDE components. To access the TDE menu, enter GO CMDFTFMQ. The TDE menu appears and lists seven options (1-7) to execute FTF Processing Commands, five options (8-12) to execute FTF Component Commands, and one option (13) to display copyright information.



The column to the left of the screen lists the Processing and Component Commands by number. The column to the right of the screen lists the corresponding commands that can be entered on the command line on any screen. To choose a command, perform one of the following steps:

- Type the number of the command and press [ENTER].

- Or -

Installing Tivoli Data Exchange on OS/400

Initializing Tivoli Data Exchange on OS/400

- Type the command and press [F4]. The corresponding command screen appears.

Notes:

- Options must be 9 characters or shorter. To see command-line options abbreviations, press [F11].
 - In the command-line interface, you specify each exit with the same arguments (for instance, -exit 4 -exitdll *dllName* -exitentry *entrypointName* [-exitdata *userData*]). On OS/400, each argument is listed. To specify arguments, go to the desired argument field.
-

Starting the Manager

To start the TDE Manager, type **8** and press [ENTER], or type **STRFTFMGR** at the command line and press [F4].

A panel appears that lists the input fields for TDE arguments. Enter the appropriate values. Because information written to the job log is buffered, you should specify a log file. All log messages are immediately written to the log file from which they can be viewed. The panel submits a job called FTFMGR.

Note:

The log file name should be in the LIBNAME/FILENAME (MEMBER) format.

Starting the Sender

To start the TDE Sender, type **9** and press [ENTER] or type **STRFTFSDR** at the command and press [F4].

A panel appears that lists input fields for TDE arguments. Enter the appropriate values. Because information written to the job log is buffered, you should specify a log file. All log messages are immediately written to the log file from which they can be viewed. The panel submits a job called FTFSDR.

Note:

The log file name should be in the LIBNAME/FILENAME (MEMBER) format.

Starting the Receiver

To start the TDE Receiver, type **10** and press **[ENTER]** or type **STRFTFRCV** at the command line and press **[F4]**.

A panel appears that lists input fields for TDE arguments. Enter the appropriate values. Because information written to the job log is buffered, you should specify a log file. All log messages are immediately written to the log file from which they can be viewed. The panel submits a job called FTFRCV.

Note:

The log file name should be in the LIBNAME/FILENAME (MEMBER) format.

Starting the Logger

To start the TDE Logger, type **11** and press **[ENTER]** or type **STRFTFLOG** at the command line and press **[F4]**.

A panel appears that lists input fields for TDE arguments. Enter the appropriate values. The panel submits a job called FTFLOG.

Starting the Status Daemon

To start the TDE Status Daemon, type **12** and press **[ENTER]** or type **STRFTFSTAD** at the command line and press **[F4]**.

A panel appears that lists input fields for TDE arguments. Enter the appropriate values. The panel submits a job called FTFSTATD.

Installing Tivoli Data Exchange on 4690

This chapter describes how to install and configure Tivoli Data Exchange (TDE) Client on the 4690 platform. The chapter also describes the commands which start up the Tivoli Data Exchange (TDE) Client components within the 4690 environment. Once the TDE Manager, Sender, and Receiver components are configured, the data transfers may be initiated. The chapter includes the following sections:

Section	Page
Overview	110
System and Installation Requirements	112
Installing the Client for 4690	112
Initializing the Client for 4690	114
Configuring the Client for 4690	117
Setting Up the MQSeries Client	117
Setting Up Clients on Another Platform	122
Using Exits on the 4690 Client	125
Tivoli Data Exchange Commands	126

Assumptions

This chapter makes the following assumptions:

- You have appropriate permission to install software on the installation machine.
- You have a working knowledge of Tivoli Data Exchange components and their interrelationships.
- You know how to access and use a command line in the appropriate operating system.
- You have connectivity to ana Tivoli Data Exchange server instance.

Overview

This sections explains the features and benefits of MQSeries Client architecture.

What Is an MQSeries Client?

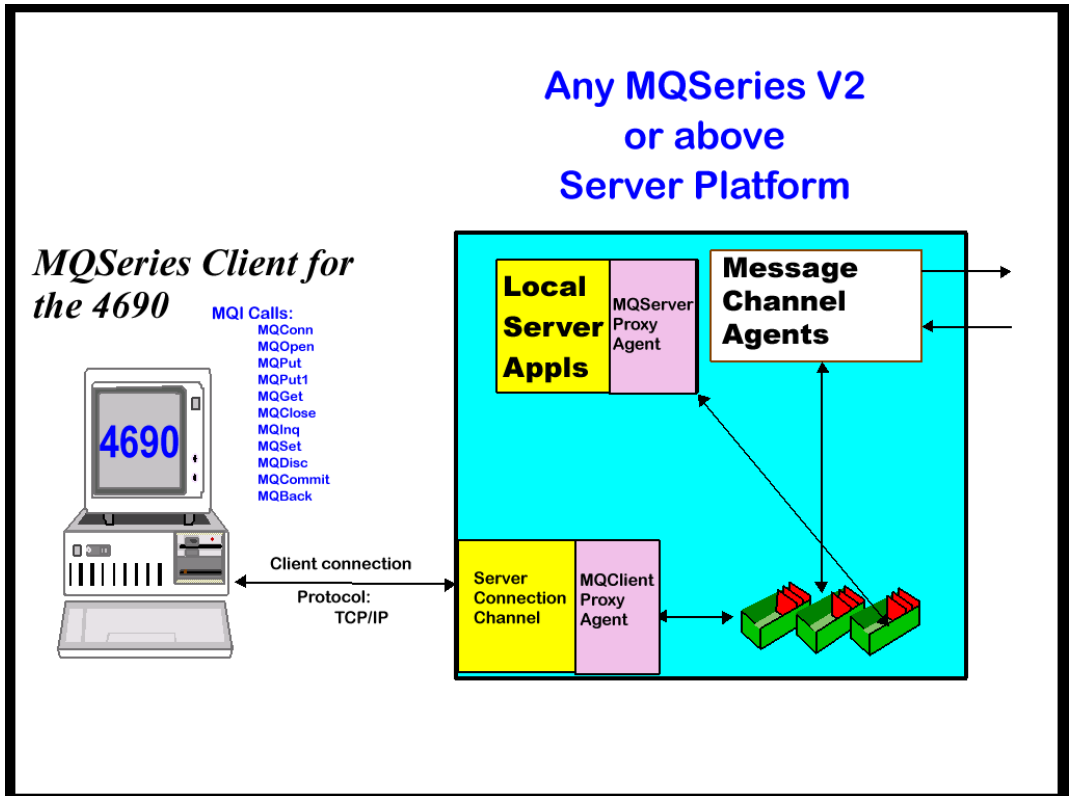
Note:

The MQSeries Client functionality is embedded within the Tivoli Data Exchange Client for 4690. Therefore, installation of the MQSeries Client is not required.

An MQSeries client is part of the MQSeries product and can be installed on a machine without installing the full queue manager. It accepts Message Queue Interface (MQI) calls from application programs and passes these MQI requests to an MQSeries server.

The MQSeries server is a full queue manager that can accept MQI calls directly from application programs that are running on the server processor. In addition, it accepts MQI requests from MQSeries clients.

The following configuration has the MQSeries client running on one machine (the client machine) and the queue manager running on a different machine.



Why Use MQSeries Clients?

Using MQSeries clients gives you an efficient way to implement MQSeries messaging and queuing without the footprint of a server installation. You can run the MQSeries client on one machine and the MQSeries server on a different machine, either physically or virtually. The benefits are:

- There is no need for a full MQSeries implementation on the client machine. This allows for installation on such platforms as: DOS, Win 32, or 4690 POS Controller.
- Hardware requirements on the client system are reduced.
- System administration requirements are reduced.

Installing Tivoli Data Exchange on 4690

System and Installation Requirements

- An MQSeries application running on a client can connect to multiple queue managers on different systems.
- Alternative channels using different transmission protocols may be used.

For more information about MQSeries Clients, see the following IBM supplied manual:

Title	Number
MQSeries Clients	GC33-1632-03

System and Installation Requirements

The following are the system requirements for the successful installation and operation of the TDE Client for 4690:

- 4690 Store System
- IBM 4690 OS/386 V1 or V2
- TCP/IP Protocol support

The following conditions must exist before you can install TDE Client for 4690:

- You must have the proper authority to install TDE Client for 4690 in the desired directories.
- An MQSeries server environment must be accessible.

No group or user is required for the installation. However, at run time, TDE Client for 4690 uses MQSeries, which may require access to the *mqm* group.

Installing the Client for 4690

To install TDE Client for 4690 locally from a floppy diskette:

1. Insert the diskette into the 4690 floppy drive.
2. Create a directory on the C drive to install into (example: md eAdpCLNT).
3. Copy the contents of the floppy into the new directory.

To install TDE Client for 4690 from a CD-ROM on another machine:

1. Put the CD-ROM into the drive that has access to the 4690.
2. CD (change directory) to the 4690\Client directory.
3. Create the target directory on the 4690.
4. FTP all files in binary from the 4690\Client directory to the 4690 target directory.

Adding Logical Filenames

- FTFCFGFI – must be set to the name of the TDE Client for 4690 configuration file with the fully qualified path.

Example:

Define FTFCFGFI = C:\FTFMQ\CONFIG\FTFCONFI.INI

- FTFCFGQ – can optionally be set to specify the default configuration queue.

Example:

Define FTFCFGQ = FTFCFG

To permanently define these logical filenames as part of the system environment, follow the same procedure as defined on page 117 for MQSERVER.

Specifying File Distribution Types on the Command Line

There are five distribution types of files on the 4690. They are associated with the target type (-ttype) and log file type (-ltype) arguments. Valid file distribution values may be 1 to 5, as follows:

- **1** – Local. Only the local machine is involved in the process. This is the default value. If the type is not specified, then it is set to **1**.

Installing Tivoli Data Exchange on 4690

Initializing the Client for 4690

- **2** – Mirrored on update. In a dual configuration the file is also written to the second 4690 controller; with Type 2 distribution type, the file is mirrored on a record-by-record basis.
- **3** – Mirrored on close. In a dual configuration the file is also written to the second 4690 controller; with Type 3 distribution type, the file is written to the second controller after the full file is written and closed on the first controller.
- **4** – Compound on update. Use when you have multiple 4690 controllers in the configuration. The file is written to all controllers on a record-by-record basis. This works like Type 2 if you have only two controllers in the configuration.
- **5** – Compound on close. Use when you have multiple 4690 controllers in the configuration. The file is written to all controllers after the full file is written and closed. This works like Type 3 if you have only two controllers in the configuration.

Specifying the Target File Distribution Type in the Configuration

You can also set up the TDE Receiver to specify the distribution type for the target files, instead of adding the `-ttype` argument to the command line. To do this, use the `4690TargetType` stanza in the `ftfconfig.ini` file with the appropriate target type value, as follows:

```
4690TargetType=2
```

Initializing the Client for 4690

This section describes how to start up the TDE Client for 4690 components in a 4690 environment using the command line.

Tivoli Data Exchange Manager Startup

To start the TDE Manager from the command line, enter the following command:

```
ftfmgr -lqm localQMgr -cfile configFile -lfile  
logFile -ltype logType -nodename nodeName
```

Where:

- **-lqm** *localQMgr* – Name of the MQSeries queue manager to which you want to connect.
- **-cfile** *configFile* – The fully qualified path and filename for the TDE Client for 4690 configuration file. If this value is not set, it defaults to the value specified in the FTFCFGFI environment variable. For more information about this logical filename, see “Installing the Client for 4690” on page 112.
- **-lfile** *logFile* – The fully qualified path and filename for the TDE Manager log file. If you do not specify a log file, output is sent to the monitor.
- **-ltype** *logType* – The distribution type of the log file. Applies to the 4690 Client only. See “Specifying File Distribution Types on the Command Line” for more information. **Valid values:** 1 - 5 **Default value:** 1
- **-nodename** *nodeName* – Associates a client definition that is described in the alias or override section of the configuration file with the message that is being sent. The nodeName can be anything, but it must be defined in the FTF configuration table at every node that wishes to participate with it.

Tivoli Data Exchange Sender Startup

To start the TDE Sender, enter the following command:

```
ftfsdr -lqm localQMgr -cfile configFile -lfile  
logFile -ltype logType -nodename nodeName
```

Where:

- **-lqm** *localQMgr* – Name of the MQSeries queue manager to which you want to connect.
- **-cfile** *configFile* – The fully qualified path and filename for the TDE Client for 4690 configuration file. If this value is not set, it defaults to the value specified in the FTFCFGFI environment variable. For more information about this logical filename, see “Installing the Client for 4690” on page 112.

Installing Tivoli Data Exchange on 4690

Initializing the Client for 4690

- **-lfile** *logFile* – The fully qualified path and filename for the TDE Sender log file. If you do not specify a log file, output is sent to the monitor.
- **-ltype** *logType* – The distribution type of the log file. Applies to the 4690 Client only. See “Specifying File Distribution Types on the Command Line” for more information. **Valid values:** 1 - 5 **Default value:** 1
- **-nodename** *nodeName* – Associates a client definition that is described in the alias or override section of the configuration file with the message that is being sent. The nodeName can be anything, but it must be defined in the FTF configuration table at every node that wishes to participate with it.

Tivoli Data Exchange Receiver Startup

To start the TDE Receiver, enter the following command:

```
ftfrcv -lqm localQMGr -cfile configFile  
      -lfile logFile -ltype logType -nodename nodeName  
      -ttype targetType
```

Where:

- **-lqm** *localQMGr* – Name of the MQSeries queue manager to which you want to connect.
- **-cfile** *configFile* – The fully qualified path and filename for the TDE Client for 4690 configuration file. If this value is not set, it defaults to the value specified in the FTFCFGFI environment variable. For more information about this environment variable, see “Installing the Client for 4690” on page 112.
- **-lfile** *logFile* – The fully qualified path and filename for the TDE Receiver log file. If you do not specify a log file, output is sent to the monitor.
- **-ltype** *logType* – The distribution type of the log file. Applies to the 4690 Client only. See “Specifying File Distribution Types on the Command Line” for more information. **Valid values:** 1 - 5 **Default value:** 1

- **-nodename** *nodeName* – Associates a client definition that is described in the alias or override section of the configuration file with the message that is being sent. The nodeName can be anything, but it must be defined in the FTF configuration table at every node that wishes to participate with it.
- **-ttype** *targetType* – The distribution type for the target files. Applies to the 4690 Client only. See “Specifying File Distribution Types on the Command Line” for more information. **Valid values:** 1 - 5 **Default value:** 1

Configuring the Client for 4690

The following tables illustrate how to configure the MQSeries Client as a background process, how to set up the background processes for the TDE components (Manager, Sender, and Receiver), and how to start the components in the background.

Setting Up the MQSeries Client

To connect the 4690 Client to a server, the MQSERVER variable must be defined. Follow the steps below from the SYSTEM MAIN MENU (displayed at startup or by pressing <Alt> + <PrtScn/SysRq> then selecting option **s** **Start new application**):

Screen	Action	Result
SYSTEM MAIN MENU	Select option 4 Installation and Update Aids	INSTALLATION AND UPDATE AIDS screen appears.
INSTALLATION AND UPDATE AIDS	Select option 1 Change Configuration Data	CONFIGURATION screen appears.
CONFIGURATION	Select option 2 Controller Configuration	LAN CONFIGURATION screen appears.
LAN CONFIGURATION	Press <Enter>	CONTROLLER CONFIGURATION screen appears.

Installing Tivoli Data Exchange on 4690

Configuring the Client for 4690

CONTROLLER CONFIGURATION	Press <Enter>	CONTROLLER CONFIGURATION STORE CONTROLLER screen appears.
CONTROLLER CONFIGURATION STORE CONTROLLER	Press <Tab> until the cursor is on the line User Logical File Names . Place an X in front of the line and press <Enter>.	USER LOGICAL FILE NAMES STORE CONTROLLER screen appears.
USER LOGICAL FILE NAMES STORE CONTROLLER	Press <Enter>	Cursor is on the line Type the logical filename being processed.
USER LOGICAL FILE NAMES STORE CONTROLLER	Type MQSERVER and press <Enter>	DEFINE LOGICAL FILE NAMES STORE CONTROLLER screen appears.
DEFINE LOGICAL FILE NAMES STORE CONTROLLER	Type QMGRNAME/TCP/XXX. XXX.XXX.XXX(NNNN) where QMGRNAME is the name of the MQSeries Server queue manager that the TDE client will connect to, XXX.XXX.XXX.XXX is the IP address of the server, and NNNN is the port number of the server (not required if default port 1414). Press <Enter>.	USER LOGICAL FILE NAMES STORE CONTROLLER screen appears.
USER LOGICAL FILE NAMES STORE CONTROLLER	Select option 4 Display Existing Logical File Names	The newly defined MQSERVER variable is displayed.
DISPLAY USER LOGICAL FILE NAMES STORE CONTROLLER	Press <F3> three times	CONFIGURATION screen appears.
CONFIGURATION	Select option 4 Activate Configuration	ACTIVATE CONFIGURATION screen appears.

ACTIVATE CONFIGURATION	Select option 2 Controller Con- figuration	The message Configuration changes are being verified will appear for a few minutes. When completed the message Additional messages available. Press F10 will appear.
ACTIVATE CONFIGURATION	Press <F10>	MESSAGE screen appears. Verify there are no error messages.
MESSAGE	Press <F10>	ACTIVATE CONFIGURATION screen appears.
ACTIVATE CONFIGURATION	Press <F3> until SYSTEM MAIN MENU appears	Completed.

Setting Up Background Processes for Components

To setup the TDE components (Manager, Sender, and Receiver) to execute in the background, follow the steps below from the SYSTEM MAIN MENU (displayed at startup or by pressing <Alt> + <PrtScn/SysRq> then selecting option **s Start new application**):

Screen/Description	Action	Result
SYSTEM MAIN MENU	Select option 4 Installation and Update Aids	INSTALLATION AND UPDATE AIDS screen appears.
INSTALLATION AND UPDATE AIDS	Select option 1 Change Configuration Data	CONFIGURATION screen appears.
CONFIGURATION	Select option 2 Controller Configuration	LAN CONFIGURATION screen appears.
LAN CONFIGURATION	Press <Enter>	CONTROLLER CONFIGURATION screen appears.

Installing Tivoli Data Exchange on 4690

Configuring the Client for 4690

CONTROLLER CONFIGURATION	Press <Enter>	CONTROLLER CONFIGURATION STORE CONTROLLER screen appears.
CONTROLLER CONFIGURATION STORE CONTROLLER	Press <Tab> until the cursor is on the line Background Application . Place an X in front of the line and press <Enter>.	BACKGROUND APPLICATION STORE CONTROLLER screen appears.
Repeat the following steps for each component (ftfmgr, ftfsdr, and ftfcv)		
BACKGROUND APPLICATION STORE CONTROLLER	Select option 1 Define a Background Application	DEFINE BACKGROUND APPLICATION STORE CONTROLLER screen appears.
DEFINE BACKGROUND APPLICATION STORE CONTROLLER	In the INITIAL MESSAGE field, type a meaningful description for the component, such as FTFMGR . Press <Tab>.	Cursor is in the PROGRAM NAME field.
DEFINE BACKGROUND APPLICATION STORE CONTROLLER	In the PROGRAM NAME field, type a component executable, such as "c:\ftfmq\ftfmgr.286" . Assuming ftfmq is the directory where TDE has been installed. The valid executable names are ftfmgr.286, ftfsdr.286, and ftfcv.286. Press <Tab>.	Cursor is in the PARAMETER LIST field.
DEFINE BACKGROUND APPLICATION STORE CONTROLLER	In the PARAMETER LIST field, type the ofile filename (to be created later), such as "-ofile c:\ftfmq\mgrofile.txt" . Press <Page Down>	Cursor is in the IPL START field.

Installing Tivoli Data Exchange on 4690 Configuring the Client for 4690

DEFINE BACKGROUND APPLICATION STORE CONTROLLER	If you want the component to be started at IPL, then type Y ; otherwise, type N ; press <Enter>	BACKGROUND APPLICATION STORE CONTROLLER screen appears.
End of Repetition		
BACKGROUND APPLICATION STORE CONTROLLER	Press <F3> until CONFIGURATION screen appears	CONFIGURATION screen appears.
CONFIGURATION	Select option 4 Activate Configuration	ACTIVATE CONFIGURATION screen appears.
ACTIVATE CONFIGURATION	Select option 2 Controller Con-figuration	The message Configuration changes are being verified will appear for a few minutes. When completed the message Additional messages available. Press F10 will appear.
ACTIVATE CONFIGURATION	Press <F10>	MESSAGE screen appears. Verify there are no error messages.
MESSAGE	Press <F10>	ACTIVATE CONFIGURATION screen appears.
ACTIVATE CONFIGURATION	Press <F3> until SYSTEM MAIN MENU screen appears	SYSTEM MAIN MENU screen appears.
Create the ofile.	Create the mgrofile.txt with the following line: -lqm QMGR -nodename NODE -cfile c:\ftfmq\ftfconfi.ini -lfile c:\ftfmq\ftfmgr.log where QMGR is the Queue Manager name on the server and NODE is the name of the 4690 Client.	File has been created. Create an ofile for the other components if needed.
IPL machine.	IPL machine	Machine is IPL'd.

Starting Background Processes for Components

To start the TDE components (Manager, Sender, and Receiver) created in the previous section, follow the steps below from the BACKGROUND APPLICATION CONTROL (displayed by pressing <Alt> + <PrtScn/SysRq> then selecting option **b Access the Background Application Control screen**):

Screen	Action	Result
BACKGROUND APPLICATION CONTROL	Press <Tab> to the background process for the component you want to start. You may have to press <Page Down> if you do not see your process on the current screen. Press <F7> to start the process.	STATUS=ACTIVE will be displayed. Repeat step for each desired component.
BACKGROUND APPLICATION CONTROL	Press <F3>	Completed.

Setting Up Clients on Another Platform

The startup parameters for the TDE Sender, Manager, and Receiver can be set up for client deliveries only by using the -nodename argument. The -nodename argument accepts an identifier that will be used in the respective source queue manager (-sqm) or destination queue manager (-dqm) arguments of the FTF request. For example, if the FTF request looks as follows when using MQSeries servers:

```
FTF -sqm MQM1 -dqm MQM2 -spath C:\FILE.1
      -dpath C:\FILE.2
```

Then the command when working with clients would look like the following:

```
FTF -sqm CLIENTA -dqm CLIENTB -spath C:\FILE.1
      -dpath C:\FILE.2
```

However, the TDE components on CLIENTA and CLIENTB need to be started with the `-nodename CLIENTA` and `-nodename CLIENTB`, respectively. This name must be specified in the `FTFNodeAlias` section, or defined in the `FTFNodeOverride` section of the configuration file at every node that wishes to participate with this client.

The client queues must be defined on the server queue manager. The queues used by the `FTFNodeAlias` section use the Alias's name and the default section of the configuration file to define the queues. For example, a queue named `CLIENTA.FTFMGR.CONTROL` must be defined on queue manager `MQM1`. All of the working queues must be defined in this manner.

See the “`FTFNodeAlias` Section” in the Configuration chapter of the *enableNet Tivoli Data Exchange Installation Guide* and the “Additional Information” section for “`FTFMGR`” in the *enableNet Tivoli Data Exchange Technical Reference* for more information on `FTFNodeAlias`. The entry in the `FTFNodeAlias` section of the configuration file would look as follows:

```
FTFNodeAlias:  
AliasQueueManager=MQM1  
Aliases=CLIENTA, CLIENTB
```

Installing Tivoli Data Exchange on 4690

Setting Up Clients on Another Platform

All of the client queues specified in the FTFNodeOverride section must be defined on the queue manager specified by the Queue Manager stanza. For example, all of the A.* queues need to be defined on the MQM1 queue manager. The entry in the FTFNodeOverride section of the configuration file would look as follows:

```
FTFNodeOverride:
name=CLIENTA, CLIENTB

CLIENTA:
QueueManager=MQM1
ManagerControlQueue=A.FTFMGR.CONTROL
ManagerSyncQueue=A.FTFMGR.SYNC

SenderNumInstances=1
SenderStageControlQueue=A.FTFSDR.STAGE.CONTROL
SenderControlQueue=A.FTFSDR.CONTROL
SenderSyncQueue=A.FTFSDR.SYNC
SenderStageQueue=A.FTFSDR.STAGE
SenderMaxStageQueues=1
SenderSystemQueue=A.FTFSDR.SYSTEM
SenderCancel=NO

ReceiverNumInstances=1
ReceiverControlQueue=A.FTFRCV.CONTROL
ReceiverSyncQueue=A.FTFRCV.SYNC
ReceiverStageQueue=A.FTFRCV.STAGE
ReceiverSystemQueue=A.FTFRCV.SYSTEM
ReceiverCancel=NO

CLIENTB:
QueueManager=MQM1
ReceiverNumInstances=1
ReceiverControlQueue=B.FTFRCV.CONTROL
ReceiverSyncQueue=B.FTFRCV.SYNC
ReceiverStageQueue=B.FTFRCV.STAGE
ReceiverSystemQueue=B.FTFRCV.SYSTEM
ReceiverCancel=NO
```

Using Exits on the 4690 Client

The processing of TDE user exits is slightly different on the 4690 platform.

Note:

For a general overview on how to use the TDE user exits, refer to the “Tivoli Data Exchange User Exits” chapter in the *Tivoli Data Exchange User's Guide*.

All of the user exit points are supported on the 4690 but instead of invoking a dynamic load library (DLL), you use the `-exitdata` option to submit a command-line request. You submit the command-line request in a similar manner to that of the enableNet TDE sample exits on other platforms. You can invoke system commands, user-written batch files and executables.

The following sample request invokes the Manager Post-Process exit (4) to move the source file to an archive directory. Since the Manager process being used in this transaction is running on the same machine as the source file, it can access the source file via its Post-Process exit. Because there are no DLLs involved in the exit processing on the 4690, TDE requires that you enter the text “4690” for both the `-exitdll` and `-exitentry` options on the request. The sample request is as follows:

```
FTF -lqm UNIXQMGR -oqm 4690QMGR -sqm 4690QMGR  
-spath C:/store1/prices.txt -dqm UNIXQMGR  
-dpath /region2/store1/prices.txt -exit 4  
-exitdll 4690 -exitentry 4690 -exitdata "move  
C:/store1/prices.txt C:/archive/store1.prices.txt"
```

Tivoli Data Exchange Commands

Overview

The commands in this section allow you to start specified TDE Client for 4690 components or the entire TDE Client for 4690 environment. These commands can be used from any command line on the specified operating systems.

You can use the following commands to manipulate TDE Client for 4690 components.

FTF

The FTF command allows you to perform data transfers from one TDE component to another. You can find an extensive explanation of this command in the “Tivoli Data Exchange Interface Commands” chapter of the *Tivoli Data Exchange Technical Reference*.

FTFMGR

The FTFMGR command allows you to start ana TDE Manager according to the conditions set in the command-line parameters. You can find an extensive explanation of this command in the “Component Configuration Commands” chapter of the *Tivoli Data Exchange Technical Reference*.

FTFRCV

The FTFRCV command allows you to start ana TDE Receiver. You can find an extensive explanation of this command in the “Component Configuration Commands” chapter of the *Tivoli Data Exchange Technical Reference*.

FTFSDR

The FTFSDR command allows you to start ana TDE Sender. You can find an extensive explanation of this command in the “Component Configuration Commands” chapter of the *Tivoli Data Exchange Technical Reference*.

FTFPING

FTFPING command allows you to send a test data transfer to TDE components that are currently running. You can find an extensive explanation of this command in the *Tivoli Data Exchange Technical Reference* manual in the “Component Configuration Commands” chapter.

FTFEND

The FTFEND command allows you to stop TDE components that are currently running. You can find an extensive explanation of this command in the “Component Configuration Commands” chapter of the *Tivoli Data Exchange Technical Reference*.

Tivoli Data Exchange Configuration

This chapter describes how to configure Tivoli Data Exchange (TDE). It describes the settings in the TDE configuration file, the corresponding settings in the MQSeries configuration file, and addresses TDE security.

This chapter contains the following sections.

Section	Page
Tivoli Data Exchange Configuration File	130
Working with the Tivoli Data Exchange Configuration File	131
FTFNodeDefault Section	132
Status Queue Default Properties	139
MVS Default Properties	142
AS/400 Default Property	144
Data Pool Section	156
FTFNodeAlias Section	159
FTFNodeOverride Defaults	161
Security	162

Assumptions

This chapter makes the following assumptions:

- MQSeries has been successfully installed.
- TDE has been installed according to the directions for the appropriate operating systems.

Tivoli Data Exchange Configuration File

The TDE configuration file allows you to set properties that govern how TDE runs. Each property is set in a stanza that consists of the property name and the value to which the property is set.

One configuration file exists per TDE installation. You can edit the configuration file using any text editor appropriate for the operating system on which the configuration file resides.

The TDE configuration file resides in the location specified during installation and, with the exception of the MVS and AS/400 sections, its contents should be the same for each TDE queue manager.

The configuration file contains the following sections, each with a related set of properties:

- Node defaults
- Data pools
- Node overrides

Using Node Default Properties and Node Override Properties

Node default properties set default values for all nodes in an TDE instance. Node override properties specify different values for specific properties of specific queue managers. The ability to override node defaults is useful when you maintain a large TDE environment and the default values are not the same for each node.

For example, if your TDE environment is distributed over five physical locations and one location has a different queue-naming convention from the other four, you would use the node default properties for the four locations and the node override properties for the fifth.

Tivoli Data Exchange Configuration File and MQSeries Scripts

When you specify a queue in the TDE configuration file, you can use the default version of the TDE script file provided in TDE as a template for creating your own MQSeries script file to create required queues. The script file `ftf.mqs` is located in the TDE installation directory for all platforms except Win 32, where it is in the TDE bin directory.

The following TDE configuration properties require that queue definitions be provided for MQSeries:

- Sender default properties
- Receiver default properties
- Manager default properties
- Status queue default properties
- Status daemon default properties
- Intercomponent communication queue properties
- Data pool properties

The documentation for each of these sets of properties contains a sample of the queue definitions required to support these properties.

Working with the Tivoli Data Exchange Configuration File

To set configuration file properties that correspond to your TDE environment, perform the following steps:

1. Stop any TDE components that are running.
2. Make the appropriate changes to the properties in each TDE configuration file. Each set of properties is encompassed by a stanza and each stanza is described in this chapter.
3. If new queues are defined, make the corresponding changes to the script that defines all TDE queues in MQSeries.
4. Run the following command to define the queues in the script:

```
runmqsc localQMgr < scriptFile
```

Where:

- *localQMgr* is the name of the queue manager running MQSeries.
 - *scriptFile* is the name of the script file containing the queue definitions required to support the TDE configuration file.
5. Restart TDE.

FTFNodeDefault Section

The FTFNodeDefault section contains properties that determine default settings for each queue manager running in TDE. When a property has a command-line equivalent, the command line overrides the default property.

The FTFNodeDefault section can contain the following types of node default properties:

- Receiver properties
- Sender properties
- Manager properties
- Status properties
- MVS properties
- AS/400 properties
- Log queue property
- File types property
- Translation property
- Notification message property
- Miscellaneous properties

Tivoli Data Exchange Receiver Default Properties

The TDE Receiver default properties specify the number of TDE Receiver instances to process data-transfer requests and identify the control, system, stage, and sync queues for the TDE Receivers.

You can specify the following properties for TDE Receivers:

- **ReceiverNumInstance** – Determines the number of TDE Receiver instances for the current TDE instance. This value must be a positive number. The operating system determines the maximum number of TDE Receivers.
- **ReceiverControlQueue** – Determines the name of the TDE Receiver's control queue. One control queue is required for all TDE Receiver instances.
- **ReceiverSyncQueue** – Determines the name of the TDE Receiver's sync queues. One sync queue is required for each TDE Receiver instance.

- **ReceiverStageQueue** – Determines the name of the TDE Receiver's stage queues. One stage queue is required for each TDE Receiver's instance.
- **ReceiverSystemQueue** – Determines the name of the TDE Receiver system queue. One system queue is required for all TDE Receiver instances.
- **ReceiverCancel** – Determines whether the TDE Receiver supports preemptive cancellation. Using this feature slows the performance of the TDE Receiver. **Valid values:** Yes, No **Default value:** No

Queue Name Expansion

Although only one name is specified for each type of queue for each CQDI component, some queue types require more than one queue per component type. TDE uses the *queue name expansion facility* to determine name assignments for multiple queues.

For example, if more than one TDE Receiver instance is defined, a stage queue must be created for each TDE Receiver instance. The TDE configuration file allows for only one stage queue name to be specified. The queue name expansion facility creates the queue names and adds a counter to each queue after the first.

If four TDE Receiver instances are specified and the specified stage queue name is FTFRVCV.STAGE, the following stage queues are created:

- FTFRVCV.STAGE
- FTFRVCV.STAGE.1
- FTFRVCV.STAGE.2
- FTFRVCV.STAGE.3

You can override the automatic dynamic queue naming by specifying the desired queue names in a comma-delimited list in the appropriate property. The following example is a queue name list using letters, rather than numbers, as the queue name discriminator:

```
QueueName=FTFRVCV.STAGE, FTFRVCV.STAGE.A,  
FTFRVCV.STAGE.B, FTFRVCV.STAGE.C
```

Example

In the following example, four TDE Receiver instances are defined and queue names are specified for all TDE Receiver queues. The cancel value is set to *No* to prevent the TDE Receiver from supporting a preemptive cancel.

```
ReceiverNumInstances=4
ReceiverControlQueue=FTFRCV.CONTROL
ReceiverSyncQueue=FTFRCV.SYNC
ReceiverStageQueue=FTFRCV.STAGE
ReceiverSystemQueue=FTFRCV.SYSTEM
ReceiverCancel=No
```

Because four receiver instances are defined, the following queues are established:

- Control queues – FTFRCV.CONTROL
- Sync queues – FTFRCV.SYNC,FTFRCV.SYNC.1,FTFRCV.SYNC.2,FTFRCV.SYNC.3
- Stage queues – FTFRCV.STAGE,FTFRCV.STAGE.1,FTFRCV.STAGE.2,FTFRCV.STAGE.3
- System queues – FTFRCV.SYSTEM

Setting the Corresponding MQSeries Script

After you have set the TDE Receiver default properties in the TDE configuration file, you must update MQSeries to correspond with the defaults (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following example lists the MQSeries script that corresponds with the properties listed in the TDE Receiver defaults example.

```
* *****
* ***** Receiver component *****
* *****
DEFINE QLOCAL(FTFRCV.CONTROL) REPLACE
* Instances=4, 4 sync queues
DEFINE QLOCAL(FTFRCV.SYNC) REPLACE
DEFINE QLOCAL(FTFRCV.SYNC.1) REPLACE
DEFINE QLOCAL(FTFRCV.SYNC.2) REPLACE
DEFINE QLOCAL(FTFRCV.SYNC.3) REPLACE
* Instances=4, 4 stage queues
DEFINE QLOCAL(FTFRCV.STAGE) REPLACE
DEFINE QLOCAL(FTFRCV.STAGE.1) REPLACE
DEFINE QLOCAL(FTFRCV.STAGE.2) REPLACE
DEFINE QLOCAL(FTFRCV.STAGE.3) REPLACE
DEFINE QLOCAL(FTFRCV.SYSTEM) REPLACE
```

Tivoli Data Exchange Sender Default Properties

The TDE Sender default properties specify the number of TDE Sender instances in the current data transfer and identify the control, system, stage, and sync queues for the TDE Senders. They also specify the maximum possible number of TDE Sender stage queues.

You can specify the following properties for TDE Senders:

- **SenderNumInstances** – Determines the number of TDE Sender instances in the current TDE instance. This value must be a positive integer. The operating system determines the maximum number of TDE Senders.
- **SenderControlQueue** – Determines the name of the TDE Sender’s control queue. One control queue is required for all sender instances.
- **SenderSyncQueue** – Determines the name of the TDE Sender’s sync queues. One sync queue is required for each TDE Sender instance.

- **SenderMaxStageQueues** – Determines the maximum number of stage queues that can be established on the TDE Sender.
- **SenderStageQueue** – Determines the name of the TDE Sender's stage queues. One stage queue is required for each TDE Receiver instance.
- **SenderSystemQueue** – Determines the name of the TDE Sender's system queue. One system queue is required for all TDE Sender instances.
- **SenderStageControlQueue** – Determines the name of the queue that holds records about files sent from staging queues.
- **SenderCancel** – Determines if the TDE Sender supports preemptive cancel. **Valid values:** Yes, No **Default value:** No
- **SenderRecordPad** – Determines whether record padding is enabled for data transfers from operating systems that support record-structured files (such as OS/390) to operating systems that do not (such as Win 32). **Valid values:** Yes, No
- **SenderPadChar** – Determines the character to be used for record padding if record padding is enabled. If you specify a literal character for this value, you must enclose it in single quotes; a hexadecimal pad character does not need to be enclosed in quotes. **Valid values:** A-Z, a-z, 0-9, space, 0x00-0xFF (where the last two positions are a hexadecimal value mapped to a character). **Default value:** space

Note:

The hexadecimal character specified as a padding character is subject to MQSeries data translation rules. The hexadecimal character is converted according to the specification of the target system's code page.

- **SenderAlwaysCheckStage** – Determines whether the stage queues are checked for the file or FTF ID in a data-transfer request. If this property is turned on, the staging queues are checked for the specified file or FTF ID regardless of the options entered on the command line. **Valid values:** On (staging queues are checked), Off (staging queues are not checked)

Example

In the following example, four TDE Sender instances are defined. Queue names are established for all queues. The *SenderMaxStageQueues* value limits the number of stage queues established to three. The TDE Sender cancel value is set to *No* to prevent the TDE Sender from supporting preemptive cancellation. Padding is enabled and the padding character is set to Hex 0 (NULL character).

```
SenderNumInstances=4
SenderControlQueue=FTFSDR.CONTROL
SenderSyncQueue=FTFSDR.SYNC
SenderStageQueue=FTFSDR.STAGE
SenderMaxStageQueues=3
SenderSystemQueue=FTFSDR.SYSTEM
SenderCancel=No
SenderRecordPad=Yes
SenderPadChar=0x00
```

Because four TDE Sender instances are defined and a maximum number of stage queues is specified, the following queues are established:

- Control queues – FTFSDR.CONTROL
- Sync queues – FTFSDR.SYNC,FTFSDR.SYNC.1,FTFSDR.SYNC.2,FTFSDR.SYNC.3
- Stage queues – FTFSDR.STAGE,FTFSDR.STAGE.1,FTFSDR.STAGE.2
- System queues – FTFSDR.SYSTEM

Setting the Corresponding MQSeries Script

After you have set the TDE Sender default properties in the TDE configuration file, you must update MQSeries to correspond with the defaults (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following example lists the MQSeries script that corresponds with the properties listed in the TDE Sender defaults example.

Tivoli Data Exchange Configuration

FTFNodeDefault Section

The number of sync queues is determined by the SenderNumInstances property. The number of stage queues is determined by the SenderMaxStageQueues property. In this example, the number of stage queues (three) is less than the number of TDE Senders (four) because the maximum stage queue value is set to three.

```
* *****
* ***** Sender component *****
* *****
DEFINE QLOCAL(FTFSDR.CONTROL) REPLACE

* Instances=4, 4 sync queues
DEFINE QLOCAL(FTFSDR.SYNC) REPLACE
DEFINE QLOCAL(FTFSDR.SYNC.1) REPLACE
DEFINE QLOCAL(FTFSDR.SYNC.2) REPLACE
DEFINE QLOCAL(FTFSDR.SYNC.3) REPLACE

* 3 stage queues
DEFINE QLOCAL(FTFSDR.STAGE) REPLACE
DEFINE QLOCAL(FTFSDR.STAGE.1) REPLACE
DEFINE QLOCAL(FTFSDR.STAGE.2) REPLACE
DEFINE QLOCAL(FTFSDR.SYSTEM) REPLACE
```

Tivoli Data Exchange Manager Default Properties

The TDE Manager default properties specify the names of the manager control queue and sync queue.

You can specify the following properties for TDE Managers:

- **ManagerControlQueue** – Determines the name of the TDE Manager's control queue.
- **ManagerSyncQueue** – Determines the name of the TDE Manager's sync queue.

Example

In the following example, a control queue and a sync queue are established for the TDE Manager.

```
ManagerControlQueue=FTFMGR.CONTROL
ManagerSyncQueue=FTFMGR.SYNC
```

Setting the Corresponding MQSeries Script

After you have set the TDE Manager default properties in the TDE configuration file, you must update MQSeries to correspond with the defaults (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following example lists the MQSeries script that corresponds with the properties listed in the Manager defaults example.

```
* *****
* Manager component *****
* *****
DEFINE QLOCAL(FTFMGR.CONTROL) REPLACE
DEFINE QLOCAL(FTFMGR.SYNC) REPLACE
```

Status Queue Default Properties

Status queue default properties determine the names of the queues used to store status information. You can specify multiple control and detail queues for the status system. Each status queue must be specified individually, however, because the queue name expansion facility does not apply to these queues. The default values are used if you do not provide them in an options file. The following values are assigned in the status default settings:

- **StatusControlQueue** – Determines the name of the status queue that receives control messages about each data transfer.
- **StatusDetailQueue** – Determines the name of the status queue that receives detailed messages from each TDE component.

- **Status** – Determines status message persistence. Persistent status messages remain in their status queues in the event of a queue manager restart. **Valid Values:**
 - PERSIST - Status messages survive a system restart.
 - NOTPERSIST - Status messages do not survive a system restart.
 - QDEF - The status message's persistence is determined by the status queue's setting in MQSeries.
 - OFF - No status messages are generated.

Note:

For information on configuring the status offload daemon component, see “TDE Status Offload Daemon” in the *Tivoli Data Exchange User's Guide*.

Status Message Routing

Status messages can be routed to any queue manager or a selected TDE component. This can provide a central repository for all status messages within an entire TDE topology.

StatusServer=*qmgrName*

Where:

- *qmgrName* – Names the queue manager to which status messages are to be routed.

StatusNodes=*nodeValues*

Where:

- *nodeValues* – Names the TDE component(s) to which status messages should be sent in addition to the queue manager specified in the StatusServer argument. **Valid Values:** REQUESTOR, ORIGINATOR, SOURCE, TARGET.

Defining a Reply Destination

An additional message can be routed to a reply queue/reply queue manager when the Receiver is started in reply mode. The message is persistent and contains the FTFCa data structure and the filename.

ReplyQueue=*queueName*

Where:

- *queueName* – Names the queue to which reply messages are to be routed.

ReplyQMgr=*qmgrName*

Where:

- *qmgrName* – Names the queue manager to which reply messages are to be routed.

Example

In the following example, settings are established for the status control queue, the status detail queue, and the reply queue.

```
StatusControlQueue=FTFSTAT.CONTROL
StatusDetailQueue=FTFSTAT.DETAIL
ReplyQueue=FTFSTAT.REPLY
Status=NOTPERSIST
```

Setting the Corresponding MQSeries Script

After you have set the TDE status default properties in the TDE configuration file, you must update MQSeries to correspond with the defaults (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following example lists the MQSeries values that correspond with the properties listed in the status defaults example.

```
* *****
* ***** Status subsystem component *****
* *****
DEFINE QLOCAL(FTFSTAT.CONTROL) REPLACE
DEFINE QLOCAL(FTFSTAT.DETAIL) REPLACE
DEFINE QLOCAL(FTFSTAT.REPLY) REPLACE
```

MVS Default Properties

The MVS default properties are default settings for the dynamic allocation of a target-file data set. The default values are used if you do not specify a value for them when you make a data-transfer request.

MVS default properties can vary from node to node. If the default properties on one node conflict with the default properties on another, they are resolved according to the following order of precedence:

1. MVS default properties set on the command line or in an API call.
2. MVS default properties set in the TDE Sender configuration file.
3. MVS default properties set in the TDE Receiver configuration file.

The properties in the configuration file are used only if no values are specified at the command line or API call. If values are set in the configuration files for both the TDE Sender and the TDE Receiver, the TDE Sender values take precedence.

Specifying an Esoteric Unit Name

To specify an esoteric name for the MVS/ESA UNIT value, follow these steps:

- Do not set a value in the MVSVOLUME stanza.
- Specify the *unit* value in the appropriate interface (FTF command, FTFReq API, ISPF interface, 5250 interface, TDE GUI), or in the MVSUNITNAME stanza in the TDE configuration file on either the TDE Sender or the TDE Receiver.

The following MVS/ESA default properties are set in the TDE configuration file:

- **MVSDIRBLOCKS** – Determines the number of directory blocks set aside for the target file. This value must be a positive integer. **Default value:** 10
- **MVSPRIMALLOC** – Determines the default number of primary allocation units set aside for OS/390 data transfers. This value must be a positive integer. **Default value:** 5

- **MVSSECALLOC** – Determines the default number of secondary allocation units set aside for OS/390 data transfers. This value must be a positive integer. **Default value:** 2
- **MVSSPACETYPE** – Determines the allocation unit used for the target file. **Valid values:** BLK (block), CYL (cylinder), TRK (track) **Default value:** TRK
- **MVSUNITNAME** – Determines the target's unit name.
- **MVSVOLUME** – Determines the target's volume serial number.
- **MVSBLKSIZE** – Determines the target file's block size. **Valid values:** 0–32760 **Default value:** 3120
- **MVSLRECL** – Determines the target file's logical record length. **Valid values:** 1–32760 **Default value:** 80
- **MVSTEXTWRAP** – Determines the treatment of the source data record if it is longer than the target data set. **Valid values:** Wrap, Truncate, Fail. **Default value:** Fail
- **MVSRECFM** – Determines the record format for MVS/ESA data transfers. **Valid values:** F (fixed), V (variable), FB (fixed block), VB (variable block). **Default value:** FB
- **MVSMODELGDG** – Defines a base generation group that describes the characteristics of the GDG. This contains definitions of the attribute's group such as record length, block size, the maximum number of revisions that will be supported, and many others. These definitions form the basis for creating a GDG during processing.
- **MVSCONSOLEOUTPUT** – Determines whether error messages are processed and displayed to the MVS operator console. The messages contain the token of "TDE" preceded by a timestamp and followed by the error text. **Default value:** Yes

No corresponding section exists in the MQSeries configuration script for MVS default properties.

Example

In the following example, all MVS default values are set.

```
MVSDIRBLOCKS=10
MVSPRIMALLOC=5
MVSSECALLOC=2
MVSSPACETYPE=TRK
MVSUNITNAME=++UNIT++
MVSVOLUME=++VOLSER++
MVSBLKSIZE=3120
MVSLRECL=80
MVSTEXTWRAP=FAIL
MVSRECFM=FB
MVSRECORDPAD=PAD
MVSMODELGDG=SYSTEM.MODEL
MVSCONSOLEOUTPUT=YES
```

AS/400 Default Property

The AS/400 default property determines the default file type for target files on AS/400 systems. The default value is used if you do not specify an AS/400 file type when you make a data-transfer request.

The following AS/400 default property is set in the TDE configuration file:

- **AS400FT** – Determines the default file type for AS/400 data transfers.
Valid values: SRCPF, SAVE **Default value:** SRCPF

No corresponding section exists in the MQSeries configuration script for AS/400 default properties.

Example

In this example, the AS/400 file type is set to SRCPF.

```
AS400FT=SRCPF
```


Setting Up Clients

In addition to setting up servers in the TDE environment, you can set up clients. MQSeries versions 2 and 5 require separate TDE installations. For more information, see “Configuring Clients” in this manual. If you are installing a client on TDE for Windows, read the “Installing TDE for Windows” chapter in this guide.

Client Overview

The TDE Client allows users to transfer data to and from machines that use the MQSeries Client, rather than the full MQSeries Server. All of the TDE components (except the status daemon) can still be used on the Client, just like they are on the Server. The difference is that no Queue Manager, nor Queues, reside on the Client machine. The Client actually uses a Queue Manager and a set of Queues that reside on a machine containing an installation of the MQSeries Server.

There is almost no loss of functionality when using the TDE Client. In fact, the only restrictions when using the Client are that there has to be a synchronous connection between that Client and the Server it is using, and compression cannot be used from or to a Client. Features such as assured delivery, restart and recovery, and centralized status are still functional when using the TDE Client.

Tivoli Data Exchange Configuration

FTFNodeDefault Section

To configure the TDE Client, some changes must be made to the `ftfconfig.ini` file. Also, since the client uses its own set of queues on the Server machine(s) it communicates with, these queues need to be created. Below is a stanza that can be used to implement an TDE Client in the `ftfconfig.ini` file:

```
FTFNodeOverride:
name=CLIENT1

CLIENT1:
QueueManager=SERVERQM
ManagerControlQueue=CLIENT1.FTFMGR.CONTROL
ManagerSyncQueue=CLIENT1.FTFMGR.SYNC

SenderNumInstances=1
SenderStageControlQueue=CLIENT1.FTFSDR.STAGE.CONTROL
SenderControlQueue=CLIENT1.FTFSDR.CONTROL
SenderSyncQueue=CLIENT1.FTFSDR.SYNC
SenderStageQueue=CLIENT1.FTFSDR.STAGE
SenderMaxStageQueues=1
SenderSystemQueue=CLIENT1.FTFSDR.SYSTEM
SenderCancel=NO

ReceiverNumInstances=1
ReceiverControlQueue=CLIENT1.FTFRCV.CONTROL
ReceiverSyncQueue=CLIENT1.FTFRCV.SYNC
ReceiverStageQueue=CLIENT1.FTFRCV.STAGE
ReceiverSystemQueue=CLIENT1.FTFRCV.SYSTEM
```

Starting the Tivoli Data Exchange Client

When starting a component on the Client machine, specify the `-lqm` (local queue manager) argument to correspond to the Queue Manager defined for that particular node. In this case, `SERVERQM` is the correct `lqm` to specify. Also, the `-nodename` argument needs to be specified to tell the component which stanza to use inside of the `ftfconfig.ini` file. The Client `nodename` is `CLIENT1`. All of the queues under this stanza are available for the Client to use, even though they reside on the Server.

Note:

It is very important that the `ftfconfig.ini` files are identical on both the Client and the Server. Mismatched configuration files can cause unexpected behavior.

To start the TDE Sender on the Client machine, type:

```
ftfsdr -lqm SERVERQM -nodename CLIENT1
```

If you want to change any configuration of the client, you must make any `ftfconfig` file modifications on both the TDE Client host machine and the TDE Server host machine. Modifications to the `ftfconfig` file require that TDE components be stopped and restarted. TDE components read the `ftfconfig` file once at startup. Changes are not recognized until the TDE component is restarted.

MQSeries configuration modifications may be required; specifically, a `SVRCONN` channel on the MQSeries Server. A `CLNTCONN` channel is also required from Client to Server. One method of creating the channel is to create an environment variable called `MQSERVER` with a value equal to `ChannelName/TransportType/ConnectionName`. Test the `SVRCONN/CLNTCONN` connection using MQSeries commands of `amqsputc` and `amqsgetc` before testing TDE FTF commands.

Using the Tivoli Data Exchange Client

When the TDE client is operational, the nodename is an eligible value for the FTF command arguments -oqm, -sqm, -dqm. Using the above NodeOverride section, an FTF command might look like this:

```
ftf -lqm SERVERQM -dqm CLIENT1 -spath  
c:\send.txt -dpath c:\receive.txt
```

When submitting an FTF command on the Client node, you should specify each of the queue manager arguments. The following example illustrates:

```
ftf -oqm SERVERQM -lqm SERVERQM -sqm CLIENT1  
-dqm SERVERQM -spath c:\send.txt -dpath  
c:\receive.txt
```

Log Queue Property

The log queue property determines whether log queues are used. It also determines the queue or queues that receive the log information.

You can specify the following log queue properties:

- **FTFLogQueue** – Determines the log queue(s) that receive log information from the TDE components governed by the configuration file.

Example

In the following example, the FTFLogQueue stanza is set to write log messages to the FTFLOG queue.

```
FTFLogQueue=FTFLOG
```

Setting the Corresponding MQSeries Script

After you have set the Log Queue property in the TDE configuration file, you must update MQSeries to correspond with the log queue setting. The following example lists the MQSeries script that corresponds with the property listed in the log queue property example.

```
*****
* Log Queue
*****
DEFINE QLOCAL(FTFLOG) REPLACE
```

File Types Property

The FileTypes property allows you to enable processing for data that does not reside in files, based on the file type specified in the FTF command line, the FTFReq CAPI, the FTFREQ COBOL API, the ISPF interface, the 5250 interface, and the TDE GUI on Win 32.

To use data that does not exist in files, you must specify the following property:

- **FileTypes** – Determines the processing invoked for each valid file type.

Unlike other property values, which take a single value or comma-delimited set of values, the FileTypes property uses a more complex syntax. The following model displays the FileTypes property's syntax:

```
FileTypes = sdrFileTypeSpec SDR sdrConnectorDLL sdrConnectorEntryPoint,  
            rcvFileTypeSpec RCVR rcvConnectorDLL  
            rcvConnectorEntryPoint
```

Where:

- *sdrFileTypeSpec* determines the file type value to be used in the calling interface to call the appropriate I/O processing on the TDE Sender.
- *sdrConnectorDLL* is the DLL invoked to work with the specified data type on the TDE Sender.
- *sdrConnectorEntryPoint* is the entry point for the DLL on the TDE Sender.

Tivoli Data Exchange Configuration

FTFNodeDefault Section

- *rcvConnectorEntryPoint* determines the file type value to be used in the calling interface to call the appropriate I/O processing on the TDE Receiver.
- *rcvConnectorDLL* is the DLL invoked to work with the specified data type on the TDE Receiver.
- *rcvConnectorEntryPoint* is the entry point for the DLL on the TDE Receiver.

Example

In the following example, sender and receiver processing is specified for two file types: AS400PF (AS/400 physical files) and ORACLE (Oracle tables).

```
FileTypes = AS400PF SDR E:\ftfbin\as400.dll AS400SND,  
AS400PF RCVR E:\ftfbin\as400.dll AS400RCV,  
ORACLE SDR E:\ftfbin\oradb.dll ORASND,  
ORACLE RCVR E:\ftfbin\oradb.dll ORARCV
```

Translation Property

The translation property allows you to translate any character or a set of characters. This translation is intended to be used to translate single-byte characters in order to correct any characters that were not properly translated between platform code pages. The translation is not meant to be used to correct any multiple-byte situations.

The property adds a new DataMap stanza to the ftfconfig.ini file. The stanza names a datamap and contains a pairing of characters to be translated. Each pair lists the “from” character first followed by the “to” character. Each paired set is separated by a comma. Multiple datamaps may be specified in the same stanza. You invoke the datamap by naming it in the -dtype argument of the FTF statement (see an explanation in the *Tivoli Data Exchange Technical Reference*). All mappings are applied AFTER the data has been converted to the local code page by MQSeries. When using the DataMap stanza, place a comma after the last character sets of the map to ensure the entire map is loaded and usable for translation.

Example

The following is an example of the DataMap stanza in the ftfconfig.ini file:

```
DataMap=MyMap 0x0d 0x0f, 0xff 0x00,  
AnotherMap 0x0e 0x0a, 0xc1 0xa9,
```

In the example above, the datamap MyMap contains “0x0d 0x0f” which is a “from and to” character pair. During translation, all hexadecimal characters of “0d” are translated to hexadecimal “0f”. The second “from and to” pair is “0xff 0x00”. Using this pair, all hexadecimal characters of “ff” are translated to hexadecimal “00”.

The datamap stanza above describes a second translation map named AnotherMap. This datamap also translates but contains different “from and to” characters. In this case “0e” becomes “0a” and “c1” becomes “a9”.

Notification Message Property

The customized notification process allows the user to deliver a notification message that is based on a transaction's status. You define the notification status, type, and data in the FTF configuration file. If the transaction's status matches the status specified, then TDE sends the information to a specified notification queue/queue manager.

Note:

TDE functions only deliver the notification messages; any further processing is the responsibility of the user.

The notification message is text only and contains the transaction's FTFID and status, notify status, notify type, and notify data. The TDE Manager builds the notification message and delivers it to the specified destination. If a notification queue has been specified, the Manager opens the queue during initialization. If no notification queue has been specified, the Manager attempts to use the local queue manager to deliver the notification message.

You can also specify the notification values on the transaction request. These values will override the values specified in the TDE Manager configuration. All three options – notification status, type, and data – must be entered in order to override the Manager values.

The property adds five new stanzas to the `ftfconfig.ini` file. `NotifyQueue` and `NotifyQMgr` define where the notification message will be delivered. `NotifyStatus` defines when a notification message will be sent (Success, Failure, Nonsuccess). `NotifyType` can be any user-defined method of notification such as e-mail, pager, fax, or WTO. `NotifyData` can be any user data that provides information related to notification processing, such as e-mail address or pager number. You can specify the following notification message properties:

- **NotifyData** – Specifies user-defined data to aid in notification, such as e-mail, pager, or fax information, that will be used to deliver a notification message based on a transaction's status. If you specify this property, you must also specify `NotifyStatus` and `NotifyType` properties.
- **NotifyQMgr** – Defines the queue manager where the notification message will be delivered.
- **NotifyQueue** – Defines the queue where the notification message will be delivered.

- **NotifyStatus** – Defines when a notification message will be sent to the NotifyQueue. The notification is sent if the transaction's status matches the status specified in this argument. If you specify this property, you must also specify NotifyData and NotifyType properties. **Valid values:** Success, Failure, Nonsuccess (includes failed, cancelled, expired)
- **NotifyType** – Specifies the user-defined method that will be used to deliver a notification message based on a transaction's status. Examples of user-defined values are: EMAIL, PAGER, FAX, WTO. If you specify this property, you must also specify NotifyData and NotifyStatus properties.

Example

The following is an example of the notification message stanzas in the `ftfconfig.ini` file:

```
*NotifyStatus=FAILURE
*NotifyType=EMAIL
*NotifyData=support@sample.com
*NotifyQueue=PROD19B
*NotifyQMgr=FTF
```

In the example above, the queue manager is FTF and the queue is PROD19B. The notification message is set up to be delivered when the transaction fails, via e-mail to *support@sample.com*. Note that these definitions are for the purpose of setup only; for the message actually to be sent, the user must provide a method of processing the message.

Miscellaneous Properties

Miscellaneous properties determine the name of the Configuration Queue, Intercomponent Communication (ICC) queue, whether OAM security is used, the default expiration time, the conversion format, the name of the node for that instance, the number of times to retry a failed component in NT TDE Service, and the wait time between tries.

The following miscellaneous properties are set in the TDE configuration file:

- **FTFConfigQueue** – Determines the name of the queue where the configuration information is stored by the FTFCFG interface.
- **FTFICCQueue** – Determines the name of the ICC queue. One ICC queue exists for all TDE components in the current instance.
- **OAMSecurity** – Determines whether OAM security is enabled. OAM security allows MQSeries authorization security to be used with TDE. **Default value:** Off
- **RequesterDefaultExpiryTime** – Determines the default expiration time after which a data-transfer request expires. This value is set in clock time in any combination of hours, minutes, or seconds. **Default value:** 24 hours
- **ConversionFormat** – Determines the conversion format to be used in processing. The stanza accepts either FTFFORMAT or MQFORMAT argument. The following table explains the values of the ConversionFormat stanza:

Value	Meaning
FTFFORMAT	Configures TDE to use FTFCVRT1 for data conversion.
MQFORMAT	Disables the use of FTFCVRT1 for data conversion.

- **NodeName** – Determines the name by which this instance is known.
- **FTFDataMessageTimeout** – Determines the number of minutes the Receiver will wait for one data message to arrive in immediate mode before reporting an error. **Default value:** 5
- **FTFMessageAckCnt** – Determines the number of messages passed between the Sender and Receiver in immediate mode before acknowledgment is sent. **Default value:** 100

Example

In the following example, settings are established for the configuration queue, intercomponent communication queue, OAM security, and the expiry value. The expiration time is set to 24 hours, the conversion format uses FTFCVRT1, the number of retries for a send is 5, and the timeout value of a retry is 30 seconds.

```
FTFConfigQueue=FTFCFG.QUEUE
FTFICCQueue=FTFICC
OAMSecurity=Off
RequesterDefaultExpiryTime=24 hours
ConversionFormat=FTFFORMAT
RetryCount=5
RetryTimeout=30
```

Setting the Corresponding MQSeries Script

After you have set the TDE miscellaneous properties in the TDE configuration file, you must update MQSeries to correspond with the defaults (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following examples lists the MQSeries scripts that correspond with the properties listed in the configuration queue and ICC examples.

```
* *****
* Configuration Queue ***
* *****
DEFINE QLOCAL(FTFCFG) REPLACE
```

```
* *****
* ICC Queue *****
* *****
DEFINE QLOCAL(FTFICC) REPLACE
```

Data Pool Section

Data pools allow TDE to go beyond the 350MB transfer queue size limit specified in MQSeries. Queue pools take several physical queues and create a single logical queue. In a transfer that uses queue pools, you reference the pool as the queue rather than the individual queues.

The TDE configuration file's Data Pool section allows you to specify the number of pools that exist, name each pool, and assign properties to them. It also allows you to specify one of the pools as the default pool. You can use the Queue Name Expansion Facility for pool queues.

No restriction exists for the quantity of data or number of messages within an TDE pool except the physical limits of the disk on which the pools exist.

The following data pool settings are specified in the Data Pool section:

- **name** – Determines the names of the queue pools set up with TDE. One name must appear for each queue pool.
- **QueueName** – Determines the name of the queue. One queue name is set once for each pool and each queue that is created after the first one is comma delimited.
- **MaxQueues** – Determines the maximum number of queues in a pool. The MaxQueues value is set once for each pool.
- **Overflow** – Determines the number of messages to be processed by the current queue before the next queue is used. The overflow value is set once for each pool.
- **DefaultPool** – Determines the default pool for data transfers. This default value is used if no pool is specified with the request.

For more information, see “Using Pools” in the *Tivoli Data Exchange User's Guide*.

Dynamic Queue Naming

When TDE establishes the queues specified in the pools, it uses the *dynamic queue name expansion facility* to handle queue naming.

This process uses the queue name specified in the TDE configuration file and appends a counter to each queue created after the first one.

In the PROD1 pool displayed in the example, the following queues are created:

- FTFDATA
- FTFDATA.1
- FTFDATA.2
- FTFDATA.3

You can override the automatic dynamic queue naming by specifying the desired queue names in a comma-delimited list in the *QueueName* property. The following example is a queue name list using letters, rather than numbers, as the queue name discriminator:

```
QueueName=FTFDATA,FTFDATA.A,FTFDATA.B,FTFDATA.C
```

Tivoli Data Exchange Configuration

Data Pool Section

Example

In the following example three pools are defined: POOL1, POOL2, and POOL3. In POOL1, four queues are established. They are used sequentially after each 50 messages are processed. In POOL2, five queues are established, to be used sequentially after each 15 messages. In POOL3, three queues are established, to be used sequentially after each 45 messages. POOL1 is designated as the default pool.

```
POOLS:  
name=POOL1, POOL2, POOL3  
POOL1:  
QueueName=FTFDATA  
MaxQueues=4  
Overflow=50  
POOL2:  
QueueName=FTFDATA2  
MaxQueues=5  
Overflow=15  
POOL3:  
QueueName=FTFDATA3  
MaxQueues=3  
Overflow=45  
DefaultPool:  
name=POOL1
```

Setting the Corresponding MQSeries Script

After you have set the pool default properties, you must update MQSeries to correspond with the pool definitions (see “Tivoli Data Exchange Configuration File and MQSeries Scripts” on page 130). The following example lists the MQSeries script that corresponds with the properties listed in the queue pools example.

```
*****
* Data pool queues
*****
DEFINE QLOCAL (FTFDATA) REPLACE
DEFINE QLOCAL (FTFDATA.1) REPLACE
DEFINE QLOCAL (FTFDATA.2) REPLACE
DEFINE QLOCAL (FTFDATA.3) REPLACE
DEFINE QLOCAL (FTFDATA2) REPLACE
DEFINE QLOCAL (FTFDATA2.1) REPLACE
DEFINE QLOCAL (FTFDATA2.2) REPLACE
DEFINE QLOCAL (FTFDATA2.3) REPLACE
DEFINE QLOCAL (FTFDATA2.4) REPLACE
DEFINE QLOCAL (FTFDATA3) REPLACE
DEFINE QLOCAL (FTFDATA3.1) REPLACE
DEFINE QLOCAL (FTFDATA3.2) REPLACE
```

FTFNodeAlias Section

The FTFNodeAlias section of the TDE 4.0 configuration file allows you to define the client nodes and their associated queue managers. These client definitions use the default client queue naming convention when resolving queue names.

Note:

Versions of TDE prior to V4.0 do not support this configuration. Use FTFNodeOverride to define queue managers and client nodes in previous TDE versions. In FTFNodeOverride, you must define each client separately.

Tivoli Data Exchange Configuration

FTFNodeAlias Section

The following node alias settings are specified in the FTFNodeAlias section:

- **AliasQueueManager** – Determines the name of the queue manager to which the defined aliases connect.
- **Aliases** – Defines the clients that connect to the defined queue manager. Values are comma-delimited.

For more information, see “Using the Default Client Queue Naming Convention” in the *Tivoli Data Exchange User's Guide*.

Example

To define all of the clients (CLIENTA through CLIENTE in this example) that connect to a queue manager (QMGRA), list them in the configuration file as follows:

```
FTFNodeAlias:
AliasQueueManager=QMGRA
Aliases=CLIENTA, CLIENTB, CLIENTC, CLIENTD, CLIENTE
```

To start the TDE Manager component with a node name of CLIENTA connecting to QMGRA, you must define CLIENTA in the FTFNodeAlias section of the configuration file as follows:

```
FTFNodeAlias:
AliasQueueManager=QMGRA
Aliases=CLIENTA
```

Note:

You must define at least one client in the configuration file.

FTFNodeOverride Defaults

The FTFNodeOverride section allows you to override settings for specific queue managers. The header part of this section lists the queue managers for which the override settings apply. The values for which override settings are specified are listed in each queue manager's detail section.

Note:

If you are specifying the same node override values for multiple queue managers, you must specify each queue manager's overrides in a separate section.

Example

In the following example, override settings are specified for queue managers PROD17A and PROD12B.

```
FTFNodeOverride:
name=PROD17A, PROD12B
PROD17A:
ReceiverControlQueue=FTF2.RECEIVER.CONTROL
PROD12B:
ReceiverControlQueue=FTF.RCV.CTRL
```

Tivoli Data Exchange Configuration

Security

If you are using a version of TDE earlier than 4.0, you must use the `FTFNodeOverride` section of the configuration file to define the name(s) of the client node(s) and corresponding queue manager(s). In the following example, override settings are specified for starting the TDE Manager component with a node name of `CLIENTA` connecting to `QMGR`.

```
FTFNodeOverride:
name=CLIENTA

CLIENTA:
QueueManager=QMGR
ManagerControlQueue=CLIENTA.FTFMGR.CONTROL
ManagerSyncQueue=CLIENTA.FTFMGR.SYNC

SenderNumInstances=3
SenderStageControlQueue=CLIENTA.FTFSDR.STAGE.CONTROL
SenderControlQueue=CLIENTA.FTFSDR.CONTROL
SenderSyncQueue=CLIENTA.FTFSDR.SYNC
SenderStageQueue=CLIENTA.FTFSDR.STAGE
SenderMaxStageQueues=1
SenderSystemQueue=CLIENTA.FTFSDR.SYSTEM
SenderCancel=NO

ReceiverNumInstances=1
ReceiverControlQueue=CLIENTA.FTFRCV.CONTROL
ReceiverSyncQueue=CLIENTA.FTFRCV.SYNC
ReceiverStageQueue=CLIENTA.FTFRCV.STAGE
ReceiverSystemQueue=CLIENTA.FTFRCV.SYSTEM
```

Security

TDE currently supports MQSeries object-level security. If appropriate access is enabled for the various queue objects to which TDE requires access, TDE abides by this security provided that `OAMSecurity` is enabled as specified in the TDE configuration file.

The Security Enabling Interface (SEI) is comprised of the following:

- Authorization Service

- OAM (Object Authority Manager), an installable component that provides authorization services. This ensures that a given user has access control facilities to the queue manager when an application issues an MQI call.

In the TDE configuration file, there is a stanza for OAMSecurity:
OAMSecurity=ON|OFF. During the execution of a data transfer, this value is checked by various TDE components.

Warning

You should employ appropriate security measures to protect each node's TDE configuration file.

If OAMSecurity is enabled before the submission of a request to TDE, the user name of the requester is retrieved by TDE. The user name is passed along with other user identity information and used by all of the MQSeries operations to read and write to TDE queues. Each of the TDE components in the subsystem (TDE Manager, TDE Sender, and TDE Receiver), upon receiving a request, read their TDE configuration file to determine whether OAMSecurity is enabled.

You may also implement one or more of the following security measures:

- If it is enabled, TDE ensures that a valid user name has been provided. If the user name is not valid, the request is rejected.
- If a file requires more than one queue for storing the information, TDE provides the ability for you to direct which queues a file may span.
- Various security schemes can be generated based on a data transfer request. For example, a user may be granted certain rights to a given TDE pool.
- Various exits can be implemented to provide additional authority verification and authentication.

Verifying the Tivoli Data Exchange Installation

This chapter describes how to verify that Tivoli Data Exchange (TDE) has been properly installed and configured. It contains the following sections:

Section	Page
Overview	166
The Verification Procedure	166
Expected Results	167

Assumptions

This chapter makes the following assumptions:

- MQSeries has been successfully installed.
- TDE has been successfully installed and configured.
- The queues specified in the TDE configuration file have been defined in MQSeries.

Overview

After TDE has been installed and configured and its queues are defined in MQSeries, you can verify that the entire environment is working by using the FTFPING command.

FTFPING ensures that all TDE and required MQSeries components are active and operational to perform data transfers. The FTFPING command accepts parameters that are similar to the FTF command, which initiates an actual data-transfer request.

The Verification Procedure

FTFPING submits a virtual data transfer that exercises all components required to perform the data transfer. The TDE Manager, TDE Sender, and TDE Receiver components are exercised. In addition, the response times are displayed to give an indication of the performance that can be expected based on the message size chosen or the specific network exercised.

To verify the local components on a node, the FTFPING command can be submitted with no arguments (on MVS and AS/400 the `-lqm` and `-cfile` parameters are mandatory). With no arguments, the default queue manager is assumed. In either case, this is the queue manager on which the TDE components operate (various TDE instances can be started on a single queue manager and TDE can operate across multiple queue managers).

Each FTFPING request exercises a specific data transfer path. To verify distributed data transfers, use the FTFPING command with the `-sqm` and `-dqm` parameters. For example, if you want to ensure that a file can be sent from NODE A to NODE B, you would issue the following from NODE A:

```
ftfping -dqm NODEB
```

This is the source node/queue manager and the destination node/queue manager. These parameters are the same used in the TDE request command line. For example, to verify two systems, SYS1 and SYS2, with two queue managers, QM1 and QM2 respectively, enter the following command:

```
ftfping -lqm QM1 -dqm QM2
```

Note:

-lqm and -cfile are required on MVS and AS/400 platforms.

The default timeout for an FTFPING request is 5 seconds. Because of network or system delays, you may need to extend the timeout to verify component functionality. For example, enter the following command to specify a 30-second timeout period for the above command:

```
ftfping -lqm QM1 -dqm QM2 -timeout 30
```

Expected Results

Successful FTFPING

A successful FTFPING has the following characteristics:

```
LQM->OQM->SQM->DQM bytes=310 time<0 secs
```

Unsuccessful FTFPING

If the FTFPING was unsuccessful, the following message is displayed:

```
FTF Ping timed out
```

If the FTFPING fails, be aware of the following:

Verifying the Tivoli Data Exchange Installation

Expected Results

1. FTFPING connects to the lqm (either the specified queue manager or the default queue manager).
2. A request is submitted to the originating TDE Manager (where this request is managed and tracked). The originating queue manager is the -oqm parameter. If oqm is not specified, it is assumed that the originating manager is operating on the same queue manager as the FTFPING command (the lqm). Therefore, the -oqm parameter is optional.
3. The TDE Manager processes the ping by submitting a send request, specified within the -sqm parameter, to the TDE Sender. The -sqm parameter is where the source file resides. The -sqm parameter is optional. If it is not specified, it is assumed that the sending node to be exercised is on the -lqm parameter.
4. The TDE Sender accepts and processes the ping by submitting a receive request, specified within the -dqm parameter, to the receiving node. The -dqm parameter is where the target file is deposited. The -dqm parameter is optional. If it is not specified, it is assumed that the TDE Receiver to be exercised is on the -lqm parameter.
5. The command `FTFPING -lqm QM1 -dqm QM2` exercises the TDE components across SYS1 and SYS2. Note that a TDE Manager, TDE Sender, and TDE Receiver would be executing at each system. The above FTFPING command sends a virtual transfer to the TDE Manager and TDE Sender on SYS1 using queue manager QM1, and to the TDE Receiver on SYS2 using queue manager QM2.

Index

Symbols

++DASD++ replacement variable 16
++DPATH++ replacement variable 16
++DQMGR++ replacement variable 16
++FTFHLQ++ replacement variable 16
++FTFINI++ replacement variable 16
++LEHLQ++ replacement variable 16
++LOADLIB++ replacement variable 16
++LQMGR++ replacement variable 17
++MEMBER++ replacement variable 17
++MODE++ replacement variable 17
++MQMHLQ++ replacement variable 17
++QMGR++ replacement variable 17
++SPATH++ replacement variable 17
++SQMGR++ replacement variable 17
++TAPE++ replacement variable 17
++TYPE++ replacement variable 17
++UNIT++ replacement variable 17
++VOLSER++ replacement variable 17
++VOLUME++ replacement variable 17

Numerics

3480 tape cartridge 23
4690 126
 file distribution types 113, 114
 FTF command 126
 FTFEND command 127

FTFMGR command 126
FTFPING command 127
FTFRCV command 126
 logical filenames 113
 user exit 125
4690 installation
 initialization 114

A

Add the environment variables
 86
Adding Logical Filenames 113
Aliases stanza 160
AliasQueueManager stanza 160
arguments
 FTFPING 44, 45, 46, 47
 FTFSTAT 46
AS/400
 default properties 144
 installation 102
 installation requirements 102
 installing on 102
 starting components 104
 starting session 104
 system requirements 102
 using the menu 105
AS400FT stanza 144

C

- cfile argument 38
 - FTFPING 44, 46
- Checking Data Set Size Allocations 19
- CLEARALL JCL 49
- CLEARALL Procedure 48
- clearing queues 48
- client
 - connecting to a UNIX server 86
 - set up 145
- client installation 66
- clientname
 - example 72
- COBOL programs 50
- command-line arguments
 - setting in initialization file 76
- compile COBOL programs 50
- Compiling and Link Editing COBOL programs 50
- configuration file 38
 - AS/400 default properties 144
 - data pool property 156
 - datamap stanza 151
 - FTFNodeAlias section 159
 - FTFNodeDefault section 132
 - FTFPING 44, 46
 - log queue property 148
 - manager default properties 138
 - MVS default properties 142
 - Node Override section 161
 - notification message property 152
 - property types 130
 - setting in initialization file 76
 - setting properties 131
 - status message routing 140
 - status queue default properties 139
 - status reply destination 140
- Configuration File and MQSeries Scripts 130
- Configuration File Values 17
- Configuration for the switch user feature 87
- configuration queue name
 - FTFPING 45, 47

- configuration values 34
- Configuring Clients 72
- Configuring the GUI 89
- ConversionFormat stanza 154
- cq argument
 - FTFPING 45, 47

D

- data pool properties
 - MQSeries script 159
- data pool property 156
- data set size allocations 19
- datamap stanza 151
- default client queue naming 159
- DefaultPool stanza 156
- destination queue manager
 - FTFPING 44
 - FTFSTAT 46
- Determining Successful FTF Commands 98
- dqm argument
 - FTFPING 44
 - FTFSTAT 46
- dynamic queue name expansion facility 157

E

- eNDI Client for 4690
 - 4690 initialization 114
- eNDI Client for 4690 Receiver
 - starting 116
- environment section 75
 - example 77
- environment variable
 - FTF_CONFIG_FILE 81
 - FTF_EXIT_PATH 81
 - FTF_IFILE 81
 - MQSERVER 81
- esoteric unit name, setting 34, 142
- Establish MQSeries Queue Manager Objects 32

Establishing Installation Values 16
 Example 45, 47, 79, 134, 137, 139, 141, 144,
 148, 150, 151, 153, 155, 158, 160,
 161
 examples
 FTFPING 45
 FTFSTAT 47
 Executing the ftfgui Script 90

F

FCOBCOMP JCL 51
 file distribution types
 4690 113, 114
 File Type Support on OpenVMS 99
 File Types Property 149
 FileTypes property 149
 FileTypes stanza 149
 FTF on 4690 126
 ftf.mqs configuration script 130
 FTF/MQ Client for 4690 Sender
 starting 114, 115
 FTF_CLASSPATH
 environment variable 89
 FTF_CONFIG_FILE 81
 environment variable 89
 FTF_CONFIG_FILE environment variable 86
 FTF_EXIT_PATH 81
 FTF_EXIT_PATH environment variable 86, 88
 FTF_IFILE 81
 FTFCFGFI logical file name 113
 FTFCFGQ logical file name 113
 FTFCfgQueue stanza 154
 FTFCVRT1 module 37
 FTFDataMessageTimeout stanza 154
 FTFEND on 4690 127
 FTFICQueue stanza 154
 FTFJDEFQ JCL 33
 FTFLogQueue stanza 148
 FTFMAKEQ JOB 32
 FTFMessageAckCnt stanza 154
 FTFMGR command

 syntax 91, 92, 93, 94
 FTFMGR on 4690 126
 FTFNodeAlias section 159
 FTFNodeDefault section 132
 FTFPING
 cfile 44, 46
 cq 45, 47
 dqm 44
 examples 45
 lqm 44
 oqm 44
 queue manager node arguments 44
 sqm 44
 successful 167
 unsuccessful 167
 FTFPING on 4690 127
 FTFRCV on 4690 126
 FTFRVC command
 syntax 116
 FTFSDR command 126
 syntax 114, 115
 FTFSDR on 4690 126
 FTFSTART
 cfile argument 38
 ofile argument 38
 syntax 38
 sysoutclass 38
 FTFSTART JCL 39
 FTFStart section 74
 example 75
 FTFSTAT
 dqm 46
 examples 47
 lqm 46
 manager node arguments 46
 oqm 46
 rqm 46
 sqm 46
 FTFUPLOD JCL 24
 FTFV2TXF JCL 43

M

G

GUI

- installing on UNIX 88

I

initialization file

- environment sections 75
- example 75, 77, 79
- FTFStart section 74
- setting command-line arguments 76
- setting configuration file value 76
- setting lqm value 75
- setting nodename value 76
- setting RetryCount value 76
- setting RetryTimeout value 76

Initialization on OS/400 104

initialize ISPF user interface 40

Initializing 37

Initializing the ISPF User Interface 40

Install from a 3480 Tape Cartridge 23

INSTALL JCL 26

installation

- OpenVMS 95

installation values 16

Installing Data Sets 25

installing on AS/400 102

Installing on Pre-OS/390 2.10 Systems 15

installing on UNIX 84

Installing Software from a CD 20

Installing the Client 66

Installing the GUI 88

Installing the Server 55

ISPF user interface 40

- initializing 40

J

Java language requirements 54

JCL Replacement Values 16

JDK requirement 54

JRE requirement 54

L

LD_LIBRARY_PATH

- environment variable 89

local queue manager

- FTFPING 44

- FTFSTAT 46

Log Queue Property 148

log queue property 148

logical filenames

- 4690 113

lqm

- setting in initialization file 75

lqm argument

- FTFPING 44

- FTFSTAGE 46

M

Manager

- starting 91, 92, 93, 94, 104

Manager Default Properties 138

manager default properties 138

- MQSeries script 139

ManagerControlQueue stanza 138

ManagerSyncQueue stanza 138

MaxQueues stanza 156

Miscellaneous Properties 154

miscellaneous properties

- MQSeries script 155

Modifying the Initialization File 80

MQSeries channels 31

MQSeries client

- definition 110

MQSeries object-level security 162

MQSeries script 130

- data pool properties 159
- manager default properties 139
- miscellaneous properties 155
- receiver default properties 135
- sender default properties 137
- status default properties 141
- MQSERVER 81, 118
 - client environment variable 89
- MVS default properties 142
- MVSBLKSIZE
 - stanza 143
- MVSBLKSIZE stanza 18, 35
- MVSCONSOLEOUTPUT 143
- MVSDIRBLOCKS
 - stanza 142
- MVSDIRBLOCKS stanza 18, 35
- MVSLRECL
 - stanza 143
- MVSLRECL stanza 18, 35
- MVSMODELGDG
 - stanza 143
- MVSMODELGDG stanza 19, 36
- MVSPRIMALLOC
 - stanza 142
- MVSPRIMALLOC stanza 18, 35
- MVSRECFM
 - stanza 143
- MVSRECFM stanza 19, 36
- MVSSECALLOC
 - stanza 143
- MVSSECALLOC stanza 18, 35
- MVSSPACETYPE
 - stanza 143
- MVSSPACETYPE stanza 18, 35
- MVSTEXTWRAP
 - stanza 143
- MVSTEXTWRAP stanza 19, 35
- MVSUNITNAME
 - stanza 143
- MVSUNITNAME stanza 18, 35
- MVSVOLUME
 - stanza 143
- MVSVOLUME stanza 18, 35

N

- Node Override section 161
- nodename
 - setting in initialization file 76
- NodeName stanza 154
- Notification Message Property 152
- notification message property 152
- NotifyData stanza 152
- NotifyQMgr stanza 152
- NotifyQueue stanza 152
- NotifyStatus stanza 153
- NotifyType stanza 153

O

- OAMSecurity
 - property 163
 - stanza 154
- Object Authority Manager (OAM) 163
- offload 140
- ofile argument 38
- OpenVMS
 - file type support 99
 - installation 95
- options file 38
- oqm argument
 - FTFPING 44
 - FTFSTAT 46
- originating queue manager
 - FTFPING 44
 - FTFSTAT 46
- OS/390 configuration values 34
- OS/390 installation
 - from CD 21
- Overflow stanza 156

S

P

PATH

environment variable 89

Pool name stanza 156

Q

queue name expansion facility 133

QueueName stanza 156

R

Receiver

starting 104

Receiver Default Properties 132

receiver default properties

MQSeries script 135

ReceiverCancel stanza 133

ReceiverControlQueue stanza 132

ReceiverNumInstance stanza 132

ReceiverStageQueue stanza 133

ReceiverSyncQueue stanza 132

ReceiverSystemQueue stanza 133

Reply Destination 140

ReplyQMgr 141

ReplyQueue 141

RequesterDefaultExpiryTime stanza 154

requesting queue manager

FTFSTAT 46

requirements

system and software 15

restarting the Win 32 service 81

RetryCount

setting in initialization file 76

RetryCount stanza 154

RetryTimeout

setting in initialization file 76

Reviewing System and Installation Requirements
15

rqm argument

FTFSTAT 46

running

FTFPING 44

FTFSTAT 46

FTFV2TXF 41

Running CLEARALL Procedure 48

Running Tests using FTFV4TXF 41

S

Security Enabling Interface (SEI) 162

Sender

starting 104

Sender Default Properties 135

sender default properties

MQSeries script 137

SenderAlwaysCheckStage stanza 136

SenderCancel stanza 136

SenderControlQueue stanza 135

SenderMaxStageQueues stanza 136

SenderNumInstance stanza 135

SenderPadRecord stanza 136

SenderStageControlQueue stanza 136

SenderStageQueue stanza 136

SenderSyncQueue stanza 135

SenderSystemQueue stanza 136

Server Installation 55

Setting the Corresponding MQSeries Script 135,
137, 139, 141, 149, 155, 159

Setting the OS/390-Specific Configuration
Values 34

Setting Up Background Processes for
Components 119

Setting Up Clients 145

setting up clients 145

Setting up MQSeries Channels 31

setting up MQSeries channels 31

source queue manager

FTFPING 44

FTFSTAT 46

Specifying File Distribution Types on the

- Command Line 113
- Specifying the Target File Distribution Type in the Configuration 114
- sqm argument
 - FTFPING 44
 - FTFSTAT 46
- starting 106
- starting a Manager 91, 92, 93, 94, 106
- starting an AS/400 session 104
- starting an eNDI Client for 4690 Receiver 116
- starting an eNDI Client for 4690 Sender 114, 115
- Starting an OS/400 Session 104
- Starting Background Processes for Components 122
- starting components
 - in OS/400 104
- Starting the Client 147
- Starting the Logger 107
- Starting the Manager 91, 106
- Starting the Receiver 93, 107
- Starting the Sender 92, 106
- Starting the Status Daemon 107
- Status Daemon
 - starting 104
- status default properties
 - MQSeries script 141
- Status Message Routing 140
- status message routing 140
- status queue default properties 139
- status reply destination 140
- Status stanza 140
- StatusControlQueue stanza 139
- StatusDetailQueue stanza 139
- StatusNodes 140
- StatusServer 140
- SVRCONN 90
- sysoutclass argument 38
- system and software requirements 15
- system output class 38

T

- testing 41
- testing the installation 44
- Testing the Installation with FTFPING 44
- The Environment Sections 75
- The FTFStart Section 74
- Tivoli Customer Support 10
- Translation Property 151
- translation property 151
- Troubleshooting MQSeries on OpenVMS 99

U

- UNIX
 - connecting a client to 86
 - environment variables 89
 - GUI environment variables 88
 - GUI installation 88
 - installation 84, 85
 - installation requirements 84
 - system requirements 84
- Unloading Tape's First Data Set 24
- user exit
 - 4690 125
- Using FTFSTAT To Verify FTFPING 46
- Using the Client 148
- using the menu on AS/400 105
- Using the OS/400 Menu 105

V

- Verifying the Installation 165

W

- What Is an MQSeries Client? 110
- Why Use MQSeries Clients? 111

W

Win 32

- client installation 66

- installation 55

- Server Installation 55

Win 32 service

- restarting 81