



Federal Tax Information - Student Aid Internet Gateway (FTI-SAIG) TDClient Host Communication Guide

Version 1.0

November 2023

DOCUMENT CONTROL

DOCUMENT INFORMATION

Title:	FTI-SAIG TDClient Host Communication Guide
Revision:	4.5
Issue Date:	11/06/2023

DOCUMENT HISTORY

Version Number	Date	Summary of Change
3.0	1/12/2010	Updated for SAIG 3.2 release
3.1	1/19/2012	Updated Windows Installation section for SAIG 3.2 release
3.2	5/4/2012	Updated TDClient 3.2 Supported Platforms
3.3	5/23/2012	Updated TDClient 3.2 supported platforms for AIX
3.4	8/13/2012	Updated TDClient 3.2 supported platforms OS/400, HP-UNIX, and Linux
3.5	2/21/2013	Updated guide with new TDClient test environment information and instructions.
3.6	9/29/2014	Updated documentation to inform readers that an IPv6 compatible version of TDClient (3.3.2) is now available.
3.6	10/29/2014	Corrected information regarding the Windows installation of the tdclient.ini file.
3.7	11/7/2014	Added information about command files, command file usage and example of transfer statements.
3.8	2/5/2015	Accepted changes, applied 508 accessibility, and finalized for posting.
3.9	1/8/2016	Updated document to reflect that TDClient now supports Linux 64-bit version, as well as Windows Server 2012. Updates were also made to remove the "www" from SAIG and FSA web portals. New content was added under section 2.5 to clarify the 2 types of TDClient installs and adding information on how to conduct an installation for upgrades made to the TDClient.
4.0	1/20/2016	Finalized edits made to document.
4.1	8/31/2020	Updated document to remove references to the FSA Download website and replaced with references to the IFAP website.
4.2	10/26/2020	Finalized edits to document.
4.3	8/18/2021	Updates made to capture changes with upgraded versions of TDClient 3.3.1 for AIX, HP-UX, Linux, OS400 and Solaris, as-well-as, TDClient 3.4.0 for Windows and MVs/zOS. TDClient 3.3+ provides improved security to comply with Federal Mandate BOD 18-01.
4.5	11/06/2023	Updates made to capture changes with upgraded versions of TDClient 3.5.0 and transmission with the New FTI-SAIG system.

Table of Contents

DOCUMENT CONTROL DOCUMENT INFORMATION.....	2
TABLE OF CONTENTS.....	2
1 OVERVIEW.....	3
1.1 PREFACE.....	3
1.1.1 Supported TDClient Platforms.....	5
1.1.2 FTI-TDCommunityManager.....	6
1.2 CONFIGURING AND INSTALLING TDCLIENT.....	7
1.2.1 Helpful Hints.....	7
1.2.2 TDClient INI and Runtime Files.....	7
1.3 INSTALLATION ON Z/OS SYSTEMS.....	9
1.3.1 Materials required prior to installation:.....	9
1.3.2 To install TDClient.....	9
1.3.3 Install Runtime Module.....	12
1.4 INSTALLATION ON HP/SUN/AIX UNIX AND LINUX SYSTEMS.....	14
1.4.1 Materials required for installation:.....	14
1.4.2 To install TDClient.....	14
1.5 INSTALLATION ON WINDOWS.....	17
1.5.1 Materials required prior to installation:.....	17
1.5.2 TDClient Clean Installation Process:.....	18
1.5.2.1 tdclient.ini File.....	29
1.5.3 TDClient RUNTIME File.....	30
1.5.4 TDClient Certificate Import.....	31
1.5.5 TDClient Uninstall Process.....	34
2 SECURITY.....	36
2.1 PASSWORD UPDATE PROCEDURE.....	36
2.1.1 General Information.....	36
2.2 BATCH PROCEDURE.....	37
3 COMMUNICATION PROCEDURES.....	40
3.1 INTRODUCTION.....	40
3.2 COMMAND LINE KEYWORDS.....	40
3.3 QUERY LIST & AUDIT LOG.....	52
3.4 FILE AND TRANSMISSION HEADER & TRAILER RECORD LAYOUTS.....	57
3.4.1 Transmission Header (O*N05) & Trailer Record (O*N95) Layouts (Required).....	58
4 APPENDIX A: COMMAND LINES FOR DIFFERENT METHODS OF SENDING DATA.....	1
5 APPENDIX B: COMMAND LINES FOR DIFFERENT METHODS OF RECEIVING DATA.....	6
6 APPENDIX C: COMMAND FILE USAGE AND TRANSFER STATEMENTS.....	13
7 APPENDIX D: UNIX/LINUX SCRIPTS FOR DIFFERENT METHODS OF RECEIVING DATA.....	15
8 APPENDIX E: WINDOWS COMMAND LINES FOR DIFFERENT METHODS OF RECEIVING DATA.....	16
9 APPENDIX F: TROUBLESHOOTING.....	17
10 APPENDIX G: CONFIGURATION CHANGES NEEDED FOR CONNECTING TO SAIG SYSTEM.....	21
11 APPENDIX H: TDCLIENT.INI FILE DEFAULTS.....	23

1 Overview

1.1 Preface

The Federal Tax Information-Student Aid Internet Gateway (FTI-SAIG) system is a store-and-forward mailbox application which supports the secure electronic exchange of Title IV data that contains Federal Tax Information (FTI) over the internet. It provides telecommunications support for the transmission of data between FSA systems, partners, and other federal entities.

This guide is designed to meet the reference needs of programmers and data processing staff who transmit Title IV Data via a windows, mainframe, or midrange computer. Additionally, this guide serves as a working document that we will periodically update and revise so that you have access to the most current information possible.

The FTI-SAIG are designed around FSA's vision and target architecture to provide an Internet solution for data transmissions. FSA to the Internet offers an integrated solution for FSA's constituents by implementing a Commercial Off-the-Shelf (COTS) application that supports multiple hardware and operating system platforms.

Note: To all third-party software providers:

1. Use the appropriate technical reference when creating output. The application system receiving the data will dictate use of low values and null values.
2. Provide a Carriage Return/Line Feed (CR/LF) in the final position of the data file to be transmitted.

The diagram in **Figure 1** represents the flow of data between FTI-SAIG destination points and Application Systems.

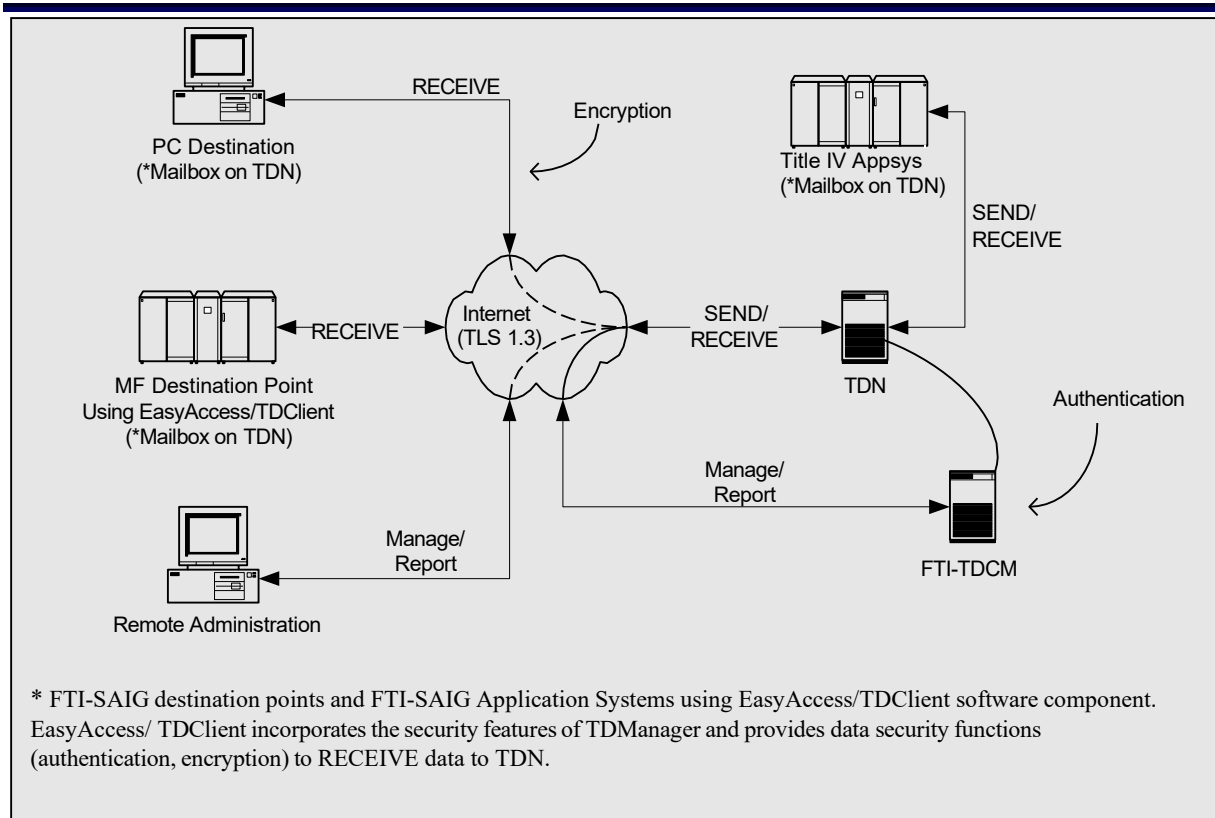


Figure 1: System Diagram

The integrated solution consists of TDClient, TDManger, TDNgine (TDN), and Federal Tax Information TDCommunity Manager components. The following sections provide more detail on each of these products.

Note: *TDClient 3.5.0 User Guide* provides guidance on usage with FTI-SAIG. While the new release can be used for transmitting with both SAIG or FTI-SAIG, this guide is tailored for FTI-SAIG system. The information for TDClient usage with SAIG is provided in [TDClient 3.4.0 Host Communications Guide dated 2021-09-10](#). Additionally, [Appendix G](#) of this guide provides a summary of configurations needed to connect to SAIG or FTI-SAIG using TDClient 3.5.0.

1.1.1 Supported TDClient Platforms

TDClient is the client software used to send and receive File Transfer Protocol (FTP) Title IV data transmissions securely over the Internet using TLSv1.3 and Advanced Encryption Standard (AES). Port 26586 needs to be open in your firewall to allow outbound TCP/IP traffic. You can download the TDClient software by going to the [Software and Other Tools - TDClient](#) page of the FSA Knowledge Center website.

TDClient 3.5.0 is supported under the following operating systems for use by destination points:

Table 1: Supported Platforms for TDClient 3.5.0

Windows	Linux	Solaris	AIX	MVS (OS/400)
<ul style="list-style-type: none">• Win10 and 11• Windows Server 2016 or higher	<ul style="list-style-type: none">• RedHat ES7.9 or higher	<ul style="list-style-type: none">• 11.3 Sparc or higher	<ul style="list-style-type: none">• 7.1 TL2 or higher	<ul style="list-style-type: none">• z/OS 2.4 or higher

1.1.2 FTI-TDCommunityManager

The FTI-SAIG TDCommunity Manager (FTI-TDCM) is the bTrade product that is used to manage FTI-SAIG destination points. The product runs as a Web application. Users of the system are system administrators, customer service/technical support staff, and FTI-SAIG destination points. Destination points can use this system to manage their mailbox and view network traffic via the Internet.

The FTI-TDCM User Guide, containing instructions on how to query your FTI-SAIG mailbox, is available on the FTI-TDCM page of the [Software and Other Tools](#) on the FSA Knowledge Center website.

1.2 Configuring and Installing TDCClient

The following sections will help guide you through installing the TDCClient 3.5.0 software with one of the supported operating system platforms listed in 1.1.1.

1.2.1 Helpful Hints

- TDCClient requires a physical connection to the Internet.
- The TDCClient software and the accompanying documentation are available for download from the [Software and Other Tools](#) page of the FSA Knowledge Center website.

1.2.2 TDCClient INI and Runtime Files

To successfully transmit with the FTI-SAIG, TDCClient users must first download the TDCClient INI (tdclient.ini) and Runtime (RTM) files from the **FTI-TDCM Security Subtab** located under the **Participant Tab**. The “Export-” buttons shown below allows for a successful download.

The screenshot displays the FTI-SAIG Mailbox interface. The top navigation bar includes links for Welcome, FTBZ571, Logout, Participants, Relationships, Messages, Archive, and Help. The main content area is divided into a left sidebar with a search function and a main panel for the 'PARTICIPANTS' subtab. The 'Security' subtab is active, showing configuration details for participant FTBZ571. A table at the bottom lists the participant's X509 certificate status, and two buttons, 'Export Runtime' and 'Export INI', are highlighted for downloading the necessary files.

Type	Status	Start Time	End Time	Issuer
X509	Current	Oct 19, 2023 06:20:59 PM	Oct 19, 2025 06:20:59 PM	

Figure 2: FTI-TDCM Security Subtab

Once you have downloaded the tdclient.ini file, you **must** open it by double clicking on the file, insert the specified additional parameters listed below under the [FTISAIGPORTAL] section and save the file after making the modifications:

DATA_OVER_COMMAND=Y
DHONLY=N

The section below displays the location in the tdclient.ini file where you must place the additional parameters:

[FTISAIGPORTAL]

HOSTIPNAME=ftisaigmailbox.ed.gov
HOSTIPNAME2=
NETWORKSTYLE=EAFTP
DIAL_ENTRY=
CASE=L
PASSIVE=N
SSL=Y
AUTO_DIAL=N
AUTO_DISCONNECT=N
SECURITYMENU=N
SUNIQUE=0
CONTROL_PORT=26586
MAX_AUTO_DIAL_DELAY=180
USERID=FT00000
PASSWORD=DFEF041B49DBA893972A7FA35C4FA0349C2F84DA01C8A9C61C08B
C869515E06D
DATA_OVER_COMMAND=Y
DHONLY=N

Note: The tdclient.ini file defaults are captured in [Appendix H](#).

1.3 Installation on z/OS Systems

To use TDClient, you must have z/OS 2.4 or above, with the MVS feature of Language Environment Version 1, Release 9 with POSIX functionality. Higher versions of z/OS must include the appropriate C++ language support feature. In order to use the file transmission feature, you must also have installed and configured TCP/IP for MVS Version 3, Release 1 or higher.

1.3.1 Materials required prior to installation:

The following items will need to be downloaded from the [Software and Other Tools](#) TDClient page located on the FSA Knowledge Center:

- z/OS TDClient from the website
- secfile

Note: You will need the *tdclient.ini* and *RTM* files referenced in [Section 1.2.2](#).

1.3.2 To install TDClient

1. FTP the TDACCESS.Vxx.XMIT file in BINARY mode to an MVS dataset with the following attributes: RECFM=FB, LRECL=80, and BLKSIZE=3120. The TDACCESS.vxx.XMIT file contains the TDClient load library, example JCL and configuration files. You can FTP the file in a variety of ways, such as from a DOS ftp prompt (see Figure 2).

```
C:\> ftp
ftp> open your.ip.address           <= connect to
                                     MVS/OS390
220 User (none)): userid           <= enter USERID
331 Enter password:xxxxxxx         <= enter PASSWORD
230 USERID logged on.
ftp> bin                           <= binary mode
200 Representation type is binary IMAGE.
ftp> quote site recfm=fb lrecl=80 blksize=3120 <= file attributes
200 SITE COMMAND WAS ACCEPTED
ftp> put c:\ TDACCESS.vxx.XMIT 'your.xmit.dataset' <=ftp the file to the
                                     mainframe
200 PORT subcommand request successful
125 Storing data set user.ealib.file
250 Transfer completed successfully
ftp> quit                           <= disconnect
```

Figure 3: Example of FTP from a DOS FTP Prompt

2. Upload the compressed file from step 4 into a Partitioned Data Set (PDS). To do this:
 - a. Go to a **TSO READY** prompt.
 - b. Type **RECEIVE INDA ('your.xmit.dataset')**. Replace “your.xmit.dataset” with the dataset you created in Step 4. See **Figure 2**.
 - c. When prompted to “enter restore parameters”, type **DA ('your.install.dataset')**. See example TSO RECEIVE below.

```
READY
RECEIVE INDA ('your.xmit.dataset')
Dataset SP01.DDNAME.INSTALL from SP)! On NODENAME
Enter restore parameters or 'DELETE' or END' +
DA ('your.install.dataset')
```

- d. Replace “your.install.dataset” with a dataset name appropriate for your installation. The install.dataset must be a different name than the dataset name used in Step 5b.

The steps above will create an Installation Library containing the files required to complete the installation of TDClient.

3. Edit the MEMBER \$INSTALL of the **your.install.dataset** and make the changes described in Steps 1-6 of the example below. Run the \$INSTALL JCL.

```
//jobname JOB (acct),pgmr,MSGLEVEL=1,REGION=7M,CLASS=A,
// MSGCLASS=X,NOTIFY=user
//*
/* MEMBER $INSTALL
/*
/* TDClient/MVS Installation JCL.
/*
/* Make the following changes:
/*
/* 1) Provide the appropriate fields on the JOBCARD, above.
/* 2) Change all occurrences of your.install.dataset to the name you created for this dataset.
/* 3) Change all occurrences of your.user.tdload to a valid destination dataset name.
/* 4) Change all occurrences of your.user.tdsamp to a valid destination dataset name.
/* 5) Change all occurrences of your.user.tdssamp to a valid destination dataset name.
/* 6) Change all occurrences of your.user.cpsamp to a valid destination dataset name.
/*
/******
/*TSO Receive for DISTLIB and SAMPLIB Datasets.
/******
/*
//RECEIVE EXEC PGM=IKJEFT01,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//EALOAD DD DSN=your.install.dataset(TDLOAD),DISP=SHR
//EASAMP DD DSN=your.install.dataset(TDSAMP),DISP=SHR
//CPDBRM DD DSN=your.install.dataset(TDSSAMP),DISP=SHR
//CPSAMP DD DSN=your.install.dataset(CPSAMP),DISP=SHR
//SYSTSIN DD *
RECEIVE INFILE(TDLOAD)
DATASET('your.user.eaload')
RECEIVE INFILE(TDSAMP)
DATASET('your.user.easamp')
RECEIVE INFILE(TDSSAMP)
DATASET('your.user.cpdbrm')
RECEIVE INFILE(CPSAMP)
DATASET('your.user.cpsamp')
/*
```

4. Allocate a new file with attributes LRECL=80, RECFM=FB, BLKSIZE=23440 and name the file '**your.dataset.prefix.TDCLIENT.EXFER.INI**'. This file will remain blank until you send your first file using TDClient, at which time it will populate with parameters contained in the TRANSFER command line of your JCL (described in Section 3, [Communication Procedures](#)).
5. Upload the **tdclient.ini** file ([Section 1.2.2](#)) as BINARY with the attributes RECFM=FB, LRECL=80, CRLF, and name the file '**your.dataset.prefix.TDCLIENT.INI**'.

Note: This file holds network configuration information and is described in *Section 3, [Communication Procedures](#)*. **Do not alter this file.**

1.3.3 Install Runtime Module

1. Upload Runtime module to the system in binary format.
2. Modify the JCL below for the following parameters.
3. KEY=XXXXXXXXXXXXXXXXXX, change to reflect the approval code for the Runtime.
4. Modify parameter **RTMFILE** to reflect the name of the uploaded runtime file.
5. Modify parameter **HIGHQUAL** (high-level qualifiers for the certs that are created during the import)
 - a. Modify parameter **STEPLIB** to point to installed TDCClient libraries.

```
//CGIMPORT JOB , 'IMPORT', CLASS=A, MSGCLASS=X, NOTIFY=&SYSUID
//*
//IMPORT PROC RTMFILE=, <== DO NOT CHANGE HERE.
//          HIGHQUAL=,          PROVIDE VALUES ON THE 'EXEC'
//          STEPLIB=
//*
//DELETE EXEC PGM=IEFBR14
//CERT DD DSN=&HIGHQUAL..CERT.FIL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//PRIVATE DD DSN=&HIGHQUAL..PRIVATE.FIL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//SYMKEY DD DSN=&HIGHQUAL..SYMKEY.FIL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//ALIAS DD DSN=&HIGHQUAL..ALIAS.TBL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//CPLOOKUP DD DSN=&HIGHQUAL..CPLOOKUP.TBL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//PARTIC DD DSN=&HIGHQUAL..PARTIC.TBL, DISP=(MOD, DELETE, DELETE),
//        SPACE=(TRK, (1))
//*
//DECOMP EXEC PGM=DECOMP, REGION=2M,
//        PARM='NOUNCOMP NOVERIFY KEEPSIGS KEY=XXXXXXXXXXXXXXXXXX'
//STEPLIB DD DSN=&STEPLIB, DISP=SHR
//DCMPLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DATAIN DD DSN=&RTMFILE, DISP=SHR
//*
//        PEND
/*
//STEP010 EXEC IMPORT,
//          RTMFILE=BTRADE.RTMFILE,
//          HIGHQUAL=BTRADE.RTM.DATA,
//          STEPLIB=BTRADE.V350.TDLOAD
/*
//DECOMP.DYNPARMS DD *
//          DSN=BTRADE.RTM.DATA
//          LRECL=84          # <=== RECORD LENGTH -- DO NOT CHANGE!
//          BLKSIZE=4204      # <=== BLOCK SIZE
//          RECFM=VB          # <=== RECORD FORMAT -- DO NOT CHANGE!
//          TRACKS            # <=== ALLOCATION UNITS
//          PRIMARY=5         # <=== PRIMARY ALLOCATION
//          SECONDARY=5       # <=== SECONDARY ALLOCATION
/*
```

6. Modify the existing JCL for transfers to include certificates by adding DD statements like

```
//PUBLKEYS DD DSN=BTRADE.RTM.DATA.CERT.FIL,DISP=SHR
//PRIVKEYS DD DSN=BTRADE.RTM.DATA.PRIVATE.FIL,DISP=SHR
```

7. If you need to view the certificates and verify the certificates, use the following JCL:

```
//***** SET VARIABLES *****/
//*
// SET TDCLNTLB=BTRADE.FIPS.V32.TDLOAD
// SET RTMQUAL=ZZCZ5K.RTM.TDACCESS
//*
//*
//*
//LISTRTM EXEC PGM=LISTRTM,REGION=0M
//*
//STEPLIB DD DSN=&TDCLNTLB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
//*****
//* MODIFY THE HIGH-LEVEL QUALIFIER OF THE FOLLOWING DATASETS TO *
//* REFER TO YOUR RUNTIME FILE NAMES CREATED BY THE 'IMPORT' JOB *
//*****
//*
//PUBLKEYS DD DSN=&RTMQUAL..CERT.FIL,DISP=SHR
//PRIVKEYS DD DSN=&RTMQUAL..PRIVATE.FIL,DISP=SHR
//*SYMKEY DD DSN=&RTMQUAL..SYMKEY.FIL,DISP=SHR
//EXTCERT DD DSN=&RTMQUAL..EXTCERT.FIL,DISP=SHR
//*
```

1.4 Installation on HP/SUN/AIX UNIX and LINUX Systems

TDClient provides file transfer capabilities with compression and encryption for UNIX and LINUX platforms.

1.1.1 Materials required for installation:

The following items will need to be downloaded from the [Software and Other Tools](#) TDClient page located on the FSA Knowledge Center website):

- Applicable TDClient executable file
- secfile

Note: You will need the INI file referenced in [Section 1.2.2](#).

1.1.2 To install TDClient

1. Make a directory called **TDC** on the UNIX or LINUX box you are using by typing **mkdir TDC** from the command prompt. You will choose the location to create the TDC directory.

Note: To verify your folder or directory location, enter the command **pwd**.

2. Type **cd TDC** from the command prompt and press **Enter**.
3. FTP the file that you downloaded to the **TDC** folder.
 - Ensure that the TDClient install file has executed, read, and write permissions.

Note: You can change the permissions with many GUI FTP programs or by using the **chmod** command at the command prompt by typing **chmod [+x +r +w] tdaccess_vxx_FSA_HPUX11.SFX**.

4. Run the TDClient self-extracting file from the command line to expand its components by typing the appropriate file name and then press **Enter**.

Note: You will see the file decompressing at this time.

5. You will be prompted with a default directory location of TDAccessx.x to install TDClient.
 - The next prompt will say: “Directory does not exist. Create Directory?”
 - Type “Y” for yes and press the **enter** key
 - Next you will be prompted to enter an e-mail address. You can enter any value here because the FTI-SAIG system does not use this e-mail address anywhere.

6. If you are installing the UNIX/LINUX version of the TDClient you will be asked: "Install TDServe? <Y or N>". Enter "**N**" and press the **Enter** key. *Do not install TDServe* The program will begin installing and display this message:

Installing TDAccess in directory /home/test/TDAccess 3.x.x

Decompressing TDAccess x.x.x Installation file (this may take a minute or two ...)

Note: *you will see the files decompressing at this time.*

7. After installing you will see the text:

Creating TDAccess subdirectories...

Resetting file permissions for TDAccess files...

cp -p exfer.ini /home/jsteapp/TDAccessx.x/exfer.ini

cp -p tpaddrss.ini /home/jsteapp/TDAccessx.x/tpaddrss.ini Installation completed!

8. After installation is completed you will see the following text:

IMPORTANT SYSTEM CONFIGURATION INFORMATION:

You must update your SHLIB_PATH environment variable to include the directory into which the TDAccess product was installed.*

An example of how to do this is given in the file .profile_example created in this directory.

Since you can control the value of SHLIB_PATH from your .profile, or from some other point, this installs program will not attempt to make this change for you.

However, the TDClient program will not be able to find the required shared libraries unless you update your environment.

* SHLIB_PATH is the shared library definition for the HP OS

LD_LIBRARY_PATH is the shared library definition for the Solaris OS

LIBPATH is the shared library definition for the AIX OS.

```

$ pwd
/home/jtest
$ mkdir TDClient
$ cd TDClient
$ ll
total 12874
-rwxrw--w- 1 jtest users 6590934 May 20 11:46
tdaccess_vxx_FSA_HPUX11.SFX
$ ^C
$ -rwxrw--w- 1 jtest users 6590934 May 20 11:46
tdaccess_vxx_FSA_HPUX11.SFX
ksh: -rwxrw--w-: not found
$ tdaccess_vxx_FSA_HPUX11.SFX
TDCompress Build 0465 (master, triple DES) (c) Copyright 1990-2003
DECOMPRESS STARTED - Tue May 20 11:51:00 2011

Decomping tdaccess_vxx_FSA_HPUX11.SFX
Decomped install.dat
Decomped tdsetup.ksh

Installing TDAccess x.x, containing TDClient x.x and TDServer x.x ...

Please enter the directory in which TDAccess
is to be installed [/home/jtest/TDAccessx.x]

Directory /home/jtest/TDAccessx.x does not exist

Decomped ./libtdclients.sl
Decomped ./libtdnmib.sl
Decomped ./license.txt
Decomped ./listrtm
Decomped ./outmsgp
Decomped ./pfxhelp.txt
Decomped ./readme.inv
Decomped ./readme.txt
Decomped ./release.txt
Decomped ./run_in_background.ksh
Decomped ./tdclient.cmd
Decomped ./tdclient_example_read.me
Decomped ./tdclientc
Decomped ./tdserver
Decomped ./tdserver.cfg
Decomped ./tdserver_install_guide.doc
Decomped ./tpaddrss.ini
*.ini not found
Creating TDAccess subdirectories...
Resetting file permissions for TDAccess files...
cp -p exfer.ini /home/jtest/TDAccessx.x/exfer.ini
cp -p tpaddrss.ini /home/jtest/TDAccessx.x/tpaddrss.ini
Installation completed!

```

Figure 4: Example of install text displayed on screen

Note: You may need to modify the .profile record of the individual(s) who will be running TDClient. Add a variable called “LD_LIBRARY_PATH=”, “SHLIB_PATH=”, or “LIBPATH=”, depending on your UNIX/LINUX OS, to define the path to the TDC directory. There are library files contained in the TDC directory that are needed by the TDClient executable.

1.5 Installation on Windows

TDClient provides file transfer capabilities with compression and encryption for the supported platforms previously mentioned in 1.1.1.

There are two types of Installation that can perform using the instructions below:

- **Clean Installation** - This type of installation is used when the TDClient is not currently installed on the target computer.
- **Upgrade Installation** - If the TDClient is already installed on the target computer, this type of installation upgrades the software, keeping customizations unchanged.

1.1.3 Materials required prior to installation:

The following materials must be downloaded from the [Software and Other Tools](#) TDClient page located on the FSA Knowledge Center website):

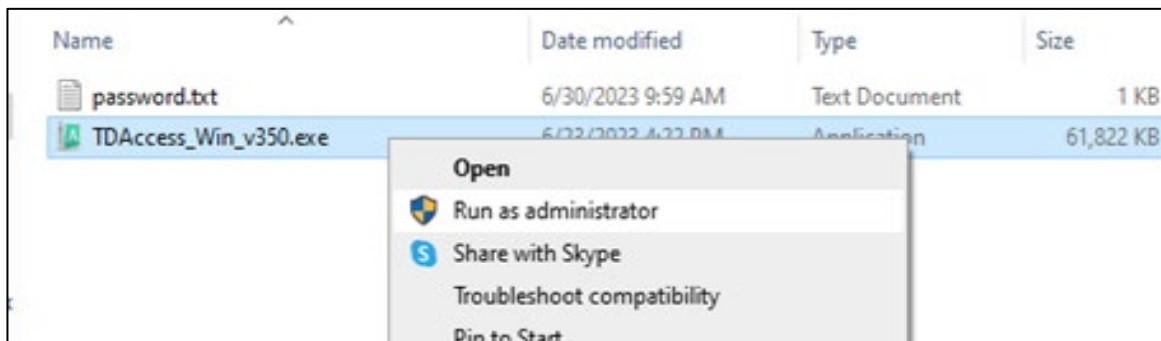
- Under TDClient click on the option for **TDClient Windows 64-bit** to download the TDClient software. TDClient 64-bit installation files are available for users based on their applicable operating system. Clicking or saving the link will download the software into your default download directory. This file is necessary for both the Clean Installation and the Upgrade Installation.
- If you are performing a Clean Install (it is not necessary for an Upgrade Installation), download the **tdclient.ini** file from FTI-TDCM. See section 1.2.2 below for instructions.

Note: You will need the *tdclient.ini* file referenced in [Section 1.2.2](#). The *tdclient.ini* file that is installed in the TDClient directory is nonfunctional. The file is strictly a blank place holder and users will need to replace the file with the downloaded *tdclient.ini* file.

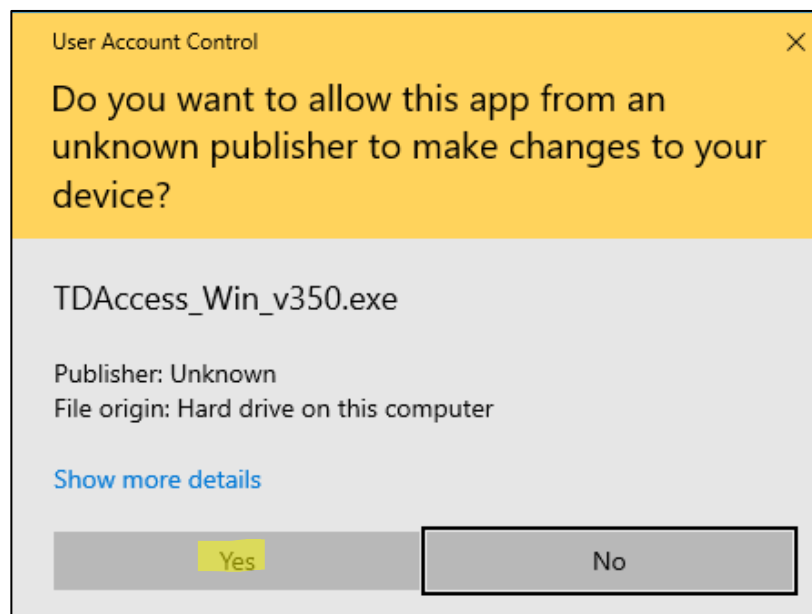
1.1.4 TDClient Clean Installation Process:

1. Move the TDClient install file to a local directory where you can perform the installation.
2. To start the Install Wizard, right click on the applicable 64-bit version of the TDAccess_Win_v350.exe file and select “**Run as administrator.**”

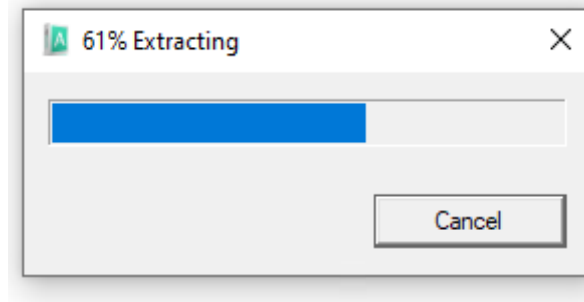
Note: TDClient user must install the software as an administrator for the software to install correctly and avoid permission errors.



3. Select **Yes** to allow the app to make changes to your device.



- Next, the software extraction process starts, and the extracting status bar will display.

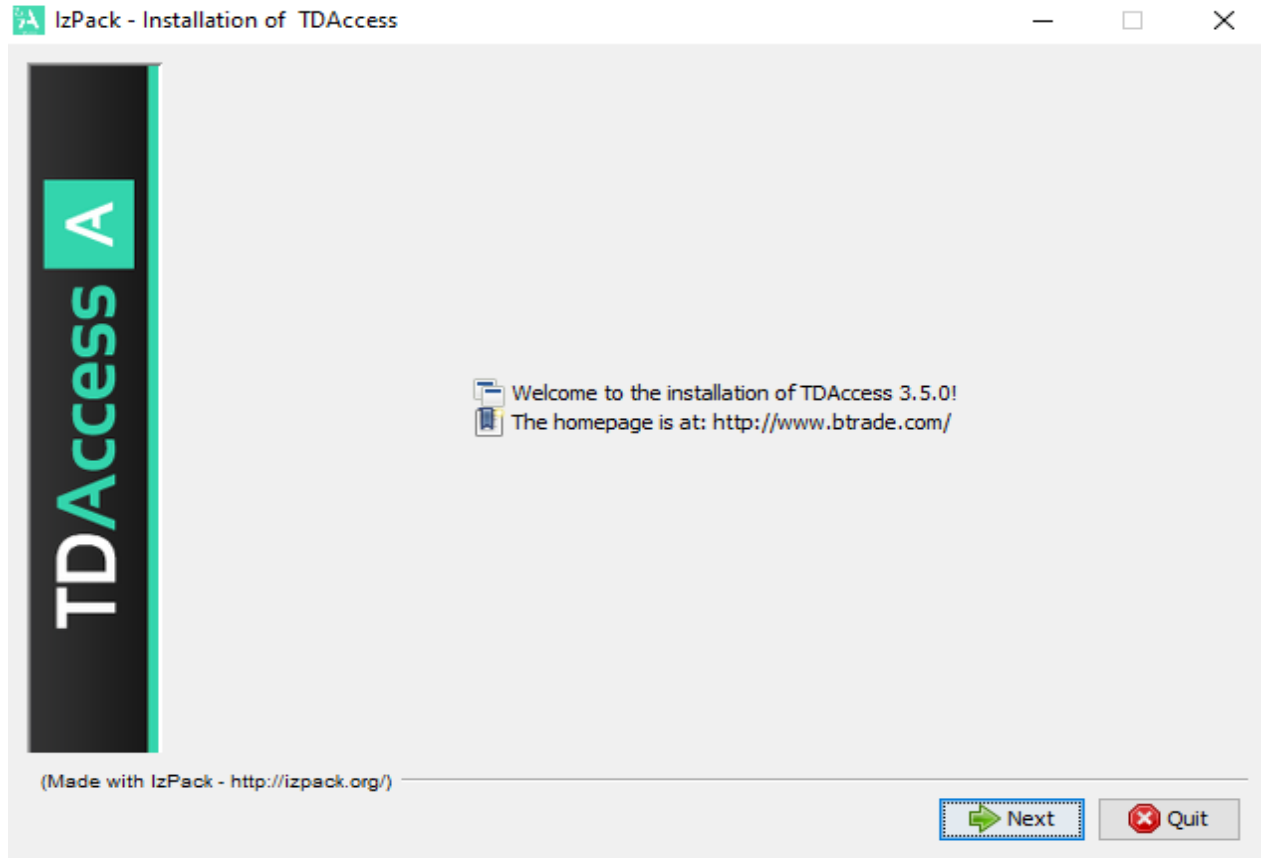


After the software extraction has completed the installer may see a command-line terminal open as displayed below. The terminal is a result of the Java installer running in command prompt and the information can be ignored.

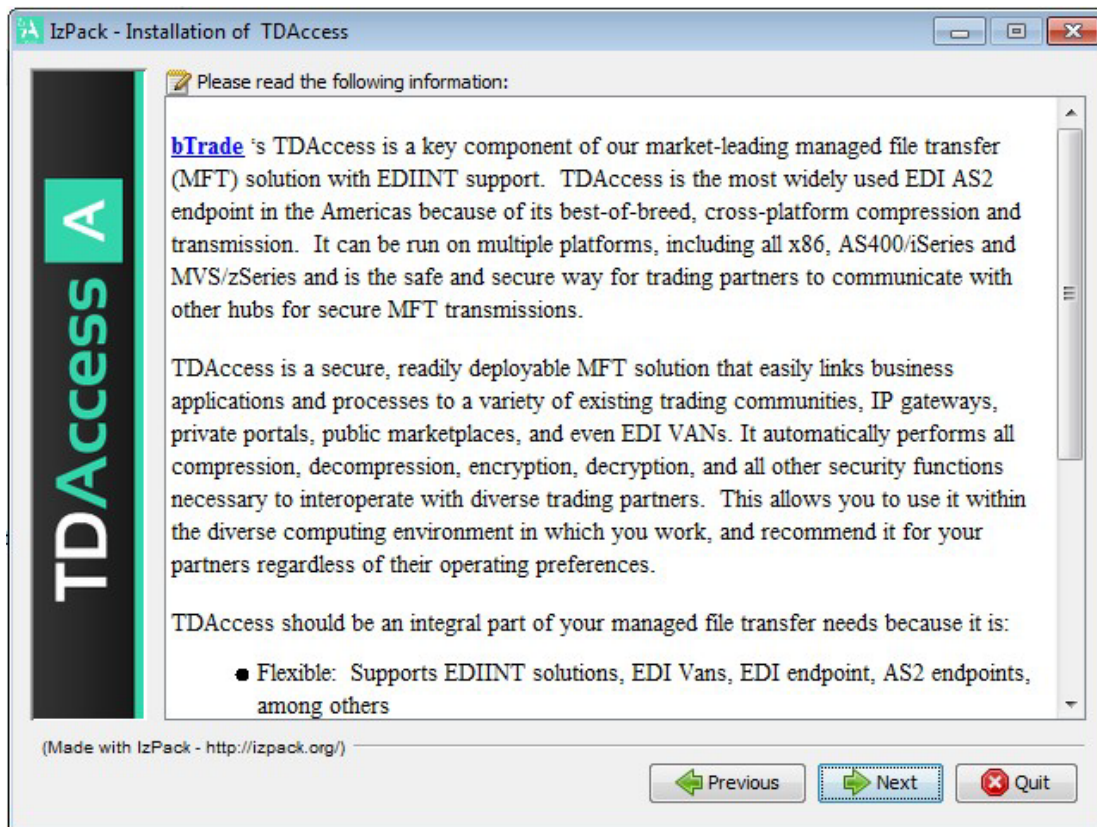
```
C:\Windows\system32\cmd.exe

C:\Users\ppanah\AppData\Local\Temp\7zSCEEA554>set "JRE_HOME=.\jre"
C:\Users\ppanah\AppData\Local\Temp\7zSCEEA554>set "JAVA=.\jre\bin\java"
C:\Users\ppanah\AppData\Local\Temp\7zSCEEA554>rem .\jre\bin\java -DDEBUG=TRUE -Xdebug -Xrunjdw:transport=dt_socket,server=y,suspend=y,address=5005 -jar tdaccess-installer.jar
C:\Users\ppanah\AppData\Local\Temp\7zSCEEA554>.\jre\bin\java -jar tdaccess-installer.jar
Nov 3, 2023 3:40:53 PM INFO: Logging initialized at level 'INFO'
Nov 3, 2023 3:40:53 PM INFO: Commandline arguments:
Nov 3, 2023 3:40:54 PM INFO: Detected platform: windows,version=6.3,arch=x64,symbolicName=null,javaVersion=1.7.0_80
Nov 3, 2023 3:40:56 PM INFO: Cannot find named resource: 'packsLang.xml' AND 'packsLang.xml_eng'
```

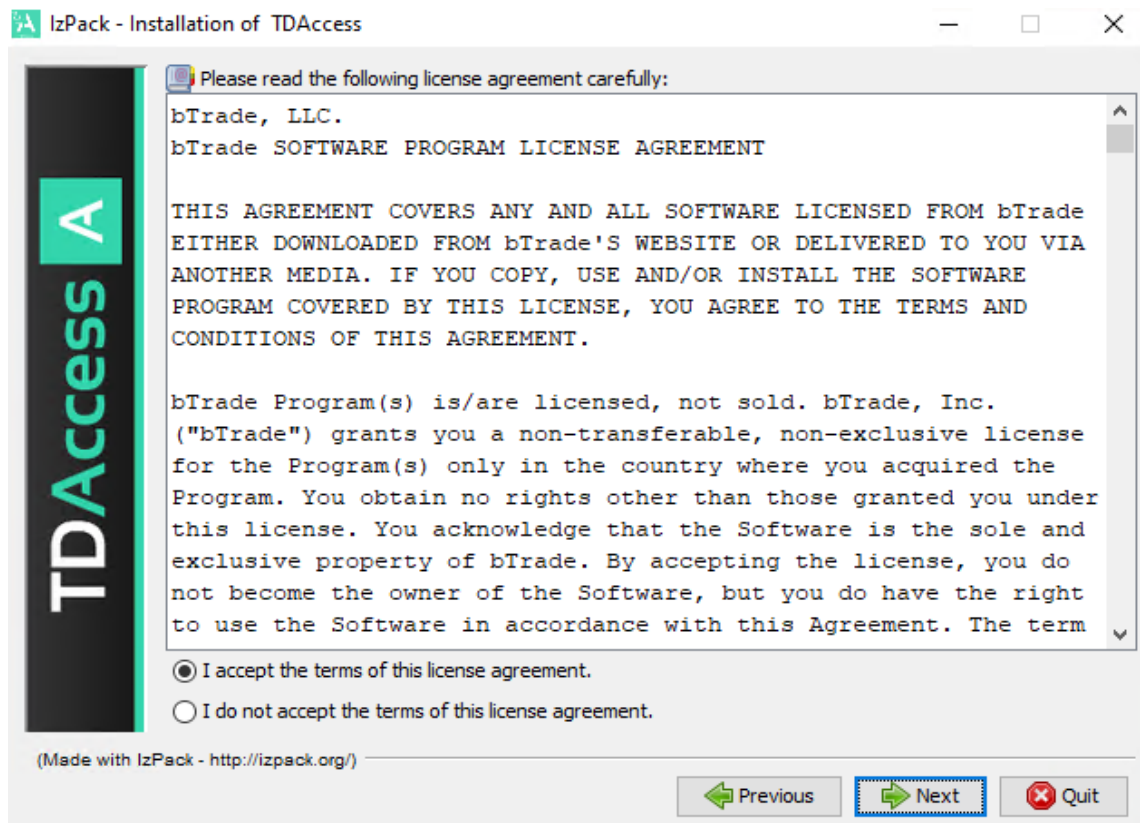
5. After the extraction has completed the installation wizard welcome screen will have the installer select the **Next** button to continue the installation.



6. Next, a description of the bTrade TDAccess software shall be displayed to the installer. Select **Next** to proceed.

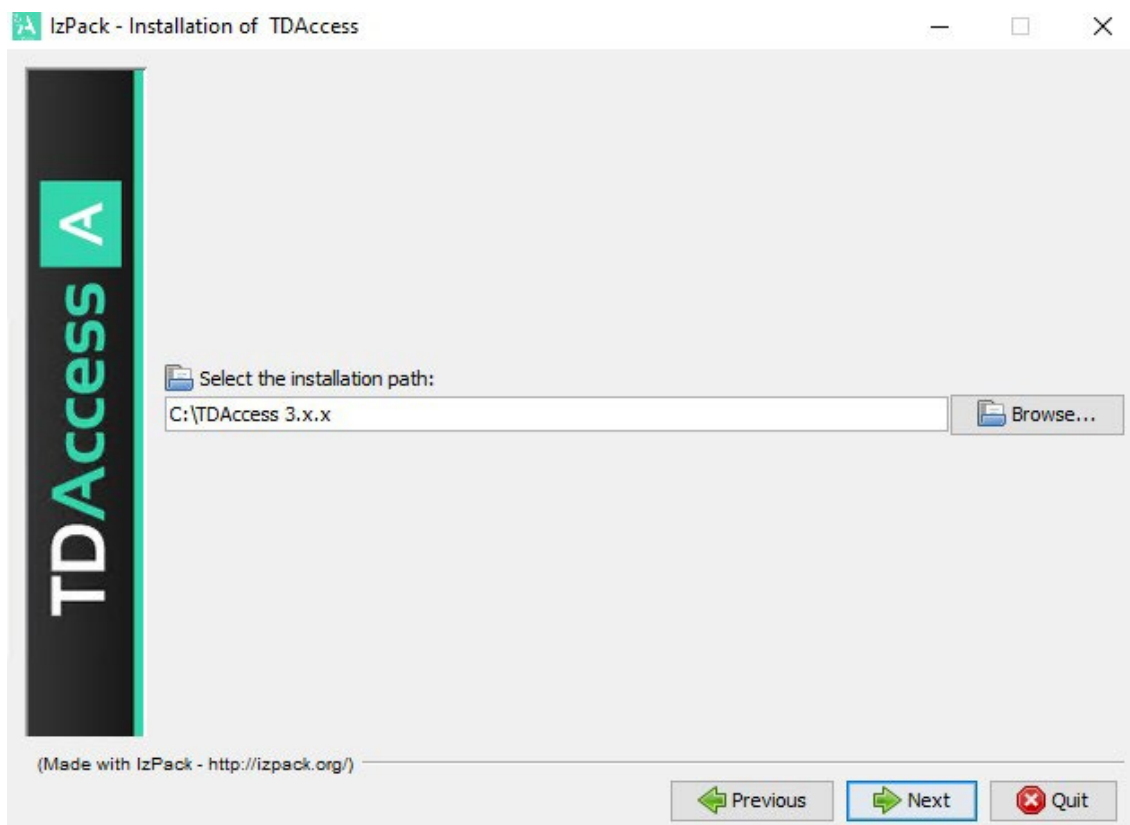


7. The installer will then be prompted to read the bTrade Software Program License Agreement and asked to select the **I accept the terms of the license agreement** radio button to continue the installation; or the **I do not accept the terms of the license agreement** radio button to cancel the installation.



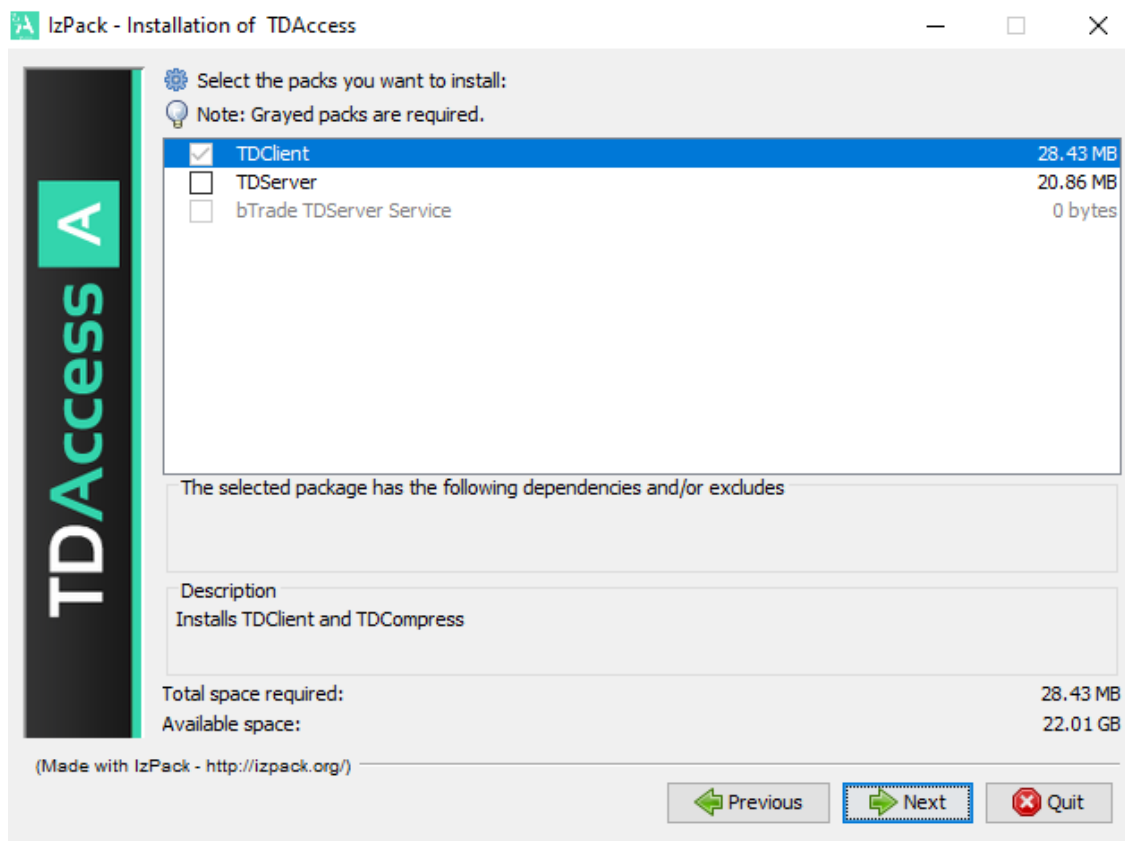
8. Once the terms of the license agreement are selected the installation path will need to be selected for the TDAccess software. Click on the **Browse** button to select the installation path and then click the **Next** button.

Note: The install wizard defaults to the Program Files directory however, it is strongly recommended that the TDClient for Windows version be installed in the **root directory**. The recommended destination folder is displayed below.



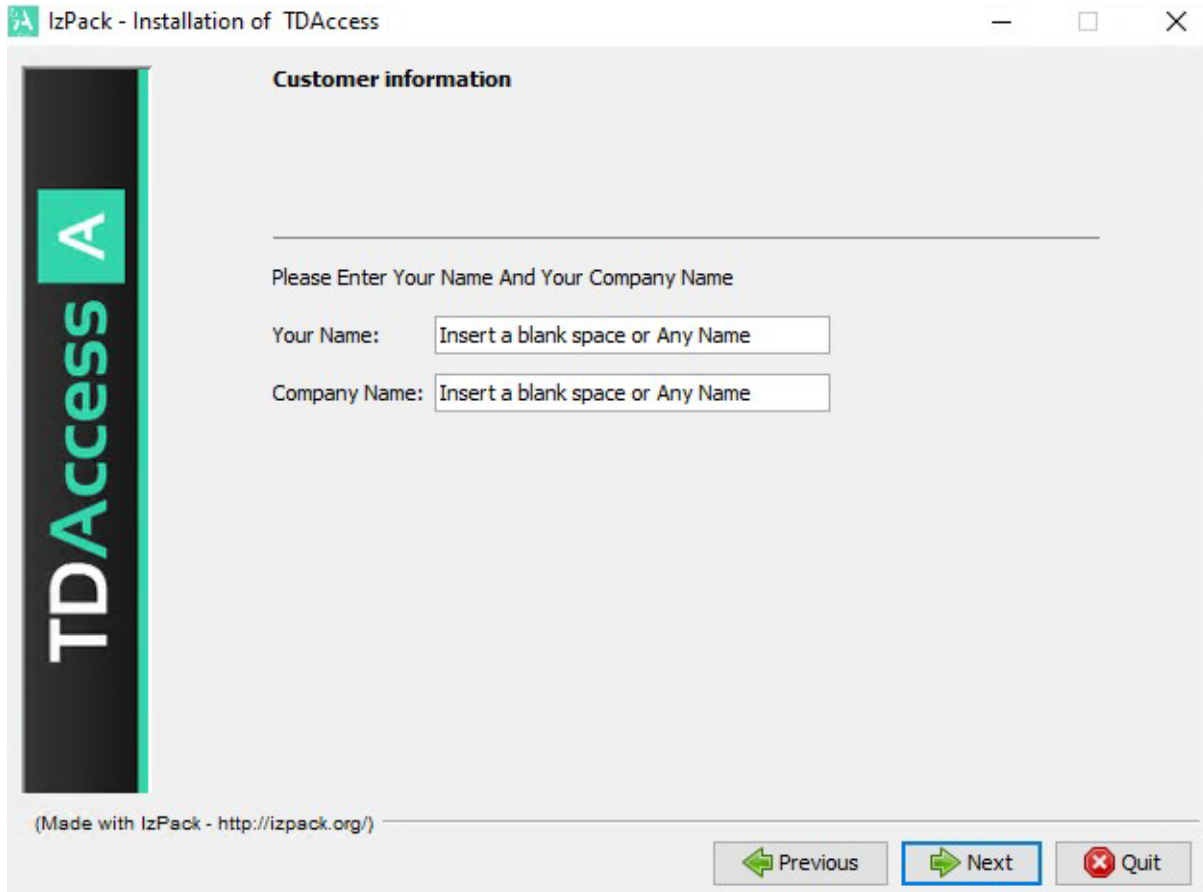
9. Next, the TDClient install packs are displayed to the installer. The **TDClient** check box is pre-selected and no further action is required for this screen. Select the **Next** button to continue the installation.

Note: The **TDServer** and **bTrade TDServer Service** checkboxes should remain unchecked.



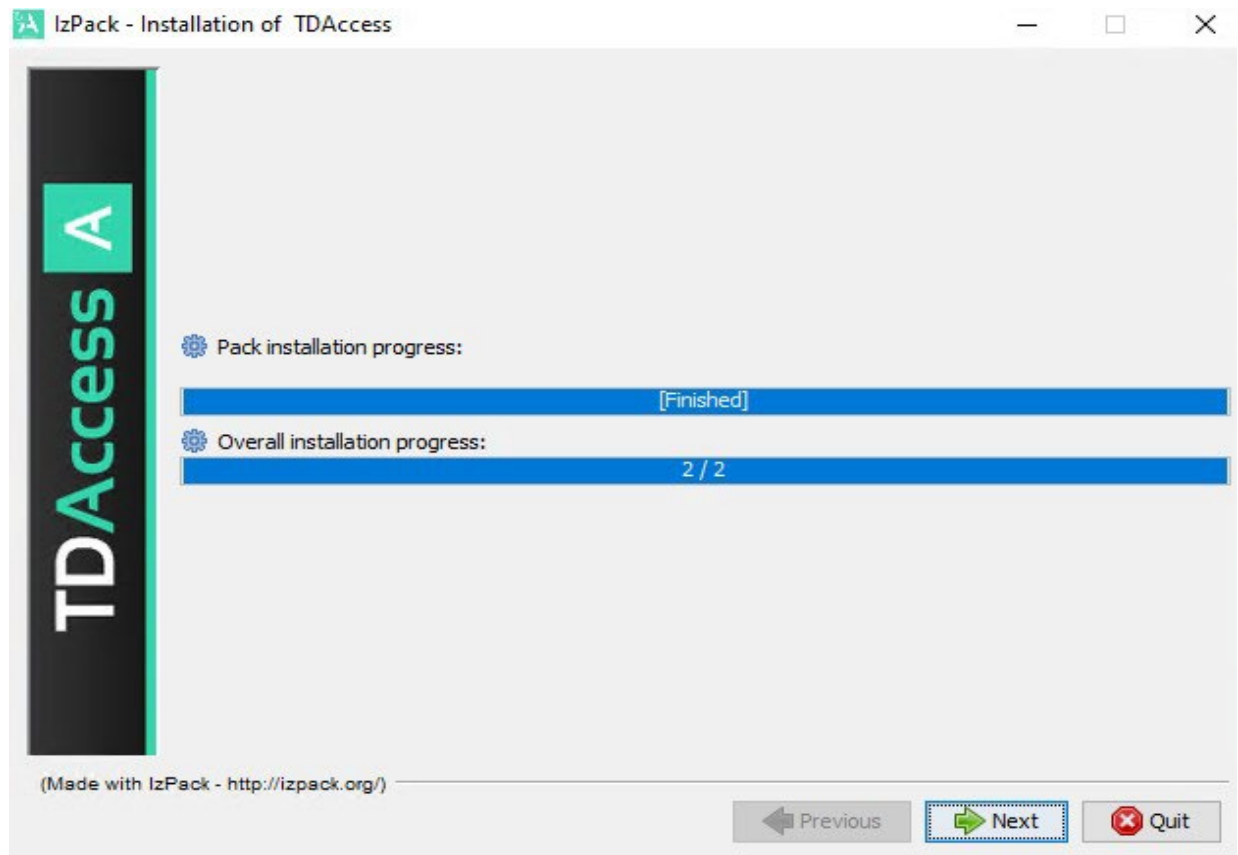
10. The **Customer information** dialog screen requires information inputted to the **Your Name** and **Company Name** fields in order to continue with the installation. The installer shall input their information or populate the fields with dummy data. Select **Next** to continue with the installation process.

Tip! Installers can enter a space into each field to proceed to the next step.

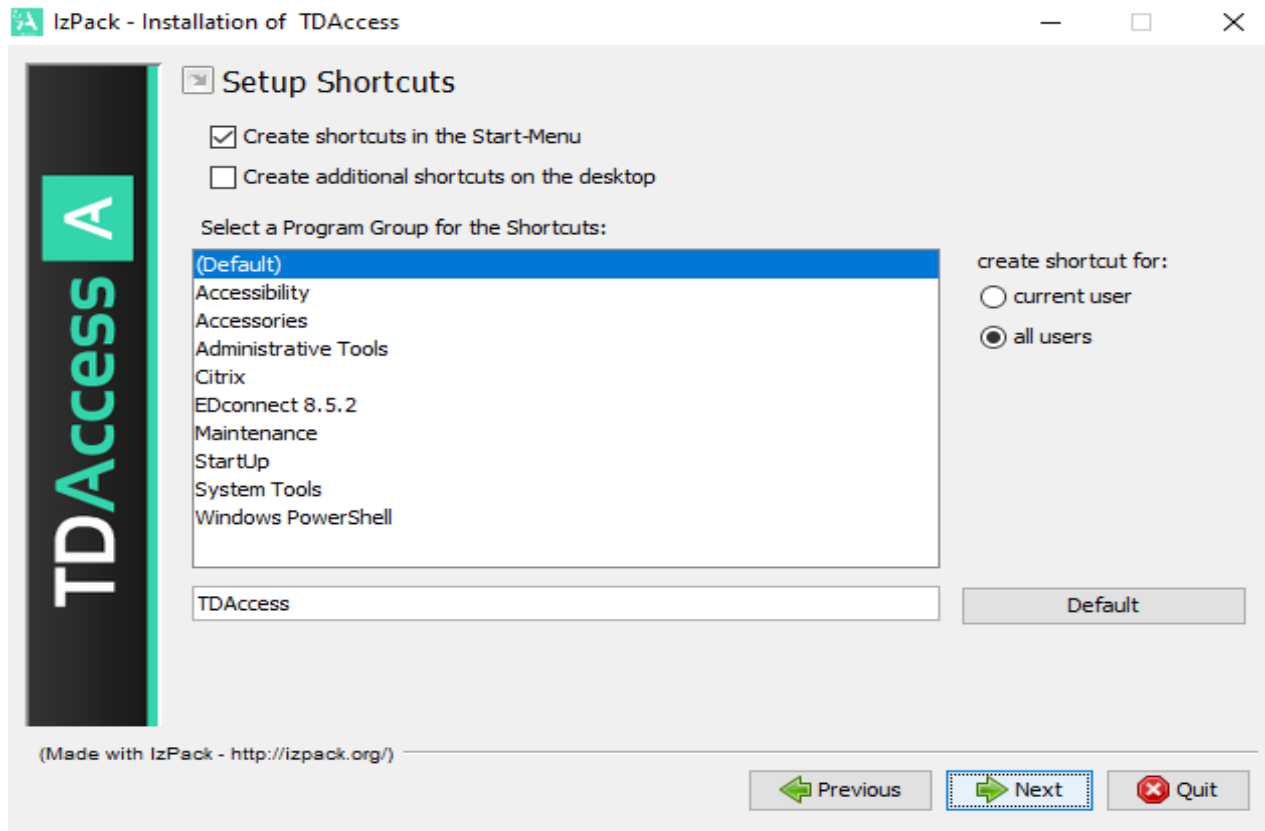


The screenshot shows a window titled "IzPack - Installation of TDAccess". On the left is a vertical banner with the "TDAccess A" logo. The main area is titled "Customer information" and contains the text "Please Enter Your Name And Your Company Name". Below this are two input fields: "Your Name:" and "Company Name:", each with a placeholder text "Insert a blank space or Any Name". At the bottom, there are three buttons: "Previous" (disabled), "Next" (active/highlighted with a blue border), and "Quit" (disabled). A footer note reads "(Made with IzPack - http://izpack.org/)".

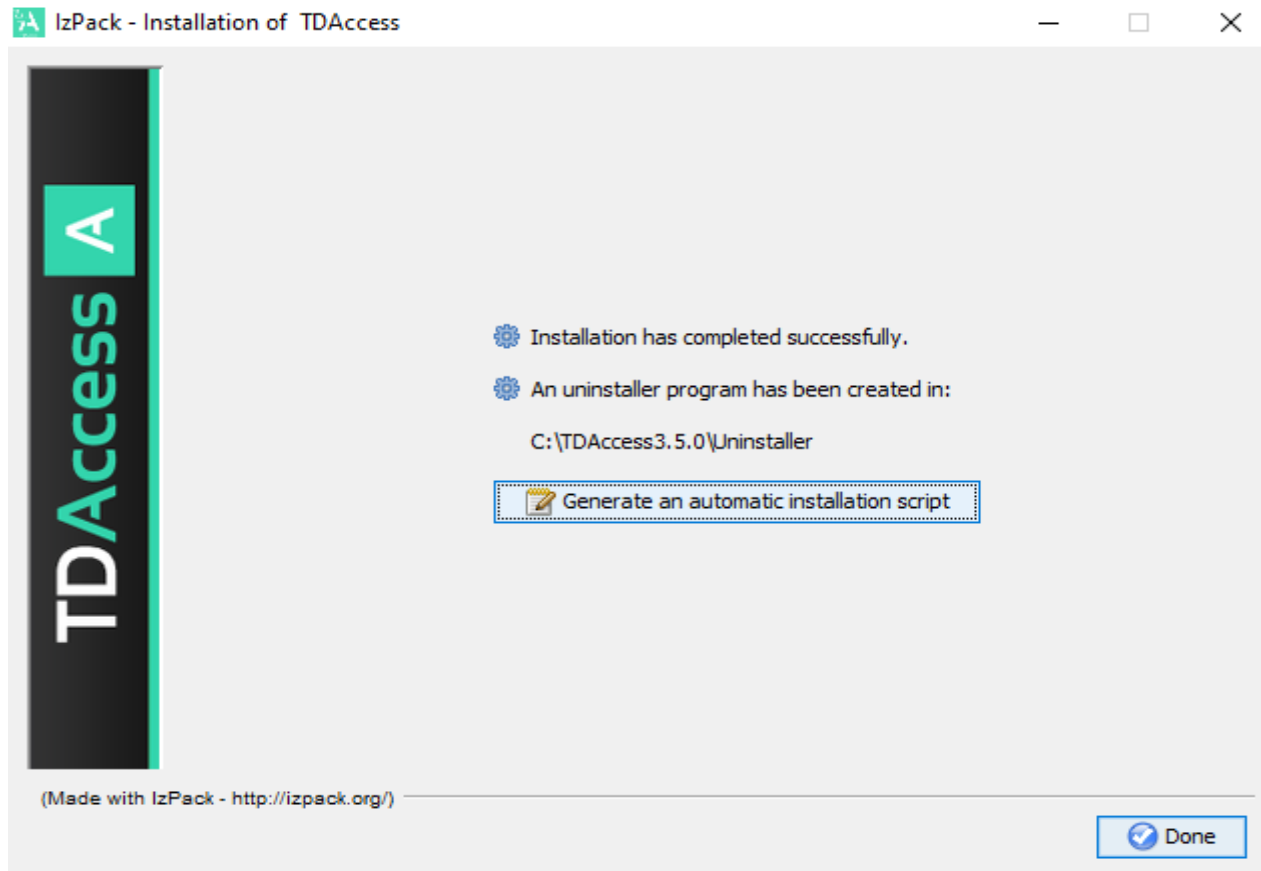
11. Next the pack installation status will display. Select the **Next** button to continue with the installation.



12. The Setup Shortcuts page displays options to set up shortcuts. Click on the **Next** button to continue with the installation.

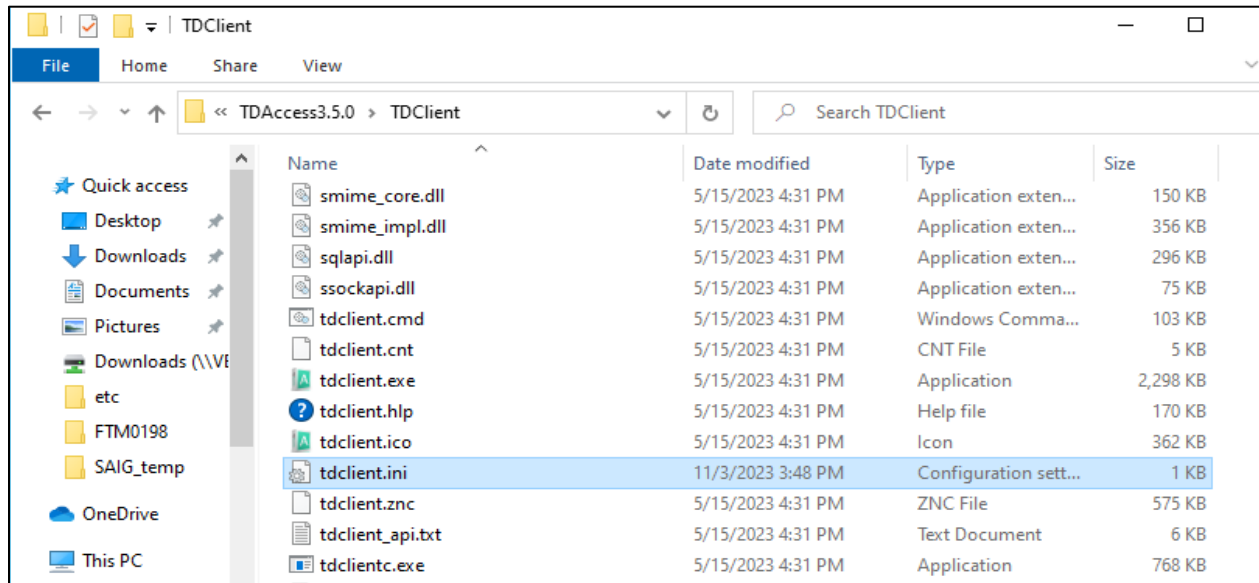


13. The TDAccess installation has completed and the TDClient is ready to use.
Select the **Done** button to exit the installation wizard



1.2.2.1 tdclient.ini File

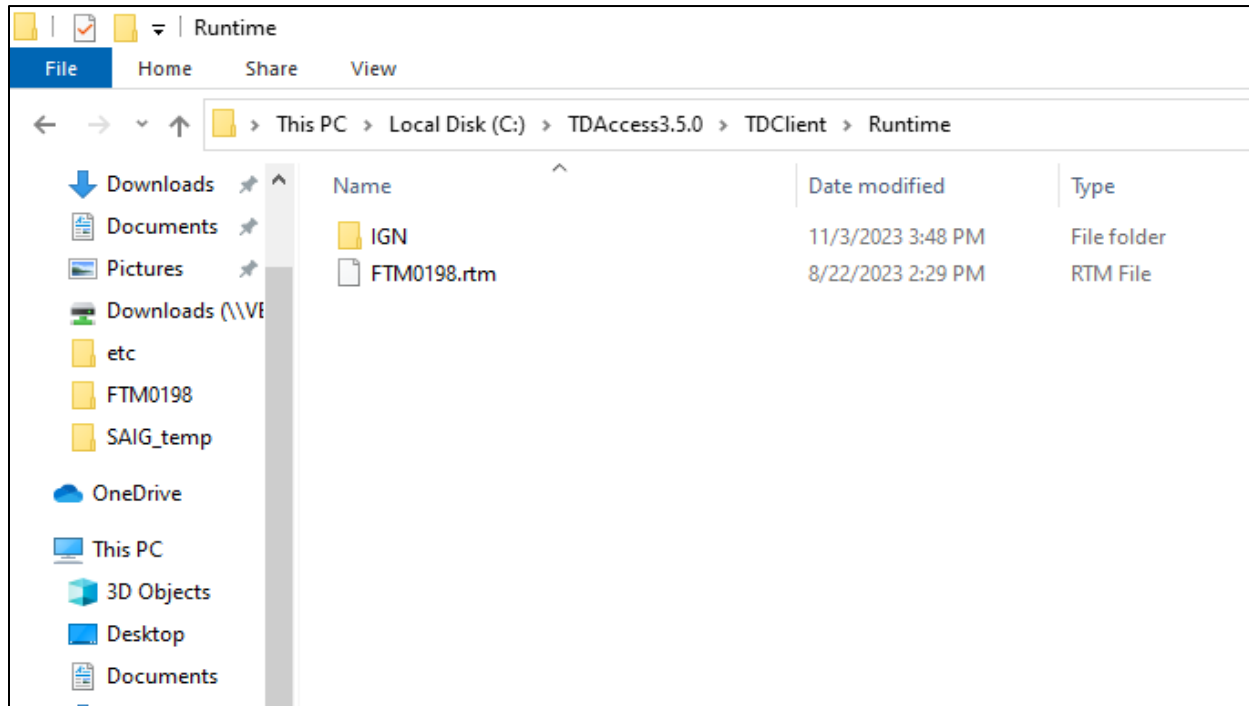
The tdclient.ini file from [Section 1.2.2](#) must be copied/moved to the TDAccess destination folder specified during the installation process. The illustration below displays the correct directory where the tdclient.ini file should be located.



1.1.5 TDClient RUNTIME File

Once the Runtime file has been downloaded from FTI-TDCM, the file must be copied/moved to the Runtime destination folder. The illustration below displays the correct directory where the tdclient.ini file should be located.

Note: The Runtime file is needed before any transmission.



1.1.6 TDClient Certificate Import

For the certificate import process, you will need to ensure that you have completed the steps in the previous section. Once you have downloaded the files and placed them in the correct destination folders, open the command-line terminal and use import command prompt shown below:

```
C:\Windows\System32\cmd.exe - import
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\TDAccess\TDClient>import

Enter full pathname of 'rtm' file from KeyManager:
```

Next, use the `.\Runtime\FTXXXXXX.rtm` command prompt. You are required to enter the full pathname of the 'rtm' file. See the example shown below:

```
Enter full pathname of 'rtm' file from KeyManager:
.\Runtime\FTM0198.rtm
```

Next enter the output path for the runtime files using the `.\Runtime` command as shown below:

```
Enter output path for runtime files (128 char max):
.\Runtime
```

Next enter the 16 characters approval code as shown below:

```
Enter approval code if required (16 characters):
D89C711DA8560583
```

Note: This code is taken from FTI-TDCM Participant tab. Please reference the FTI-TDCM guide for additional information.

You should see the following message after entering the approval code in the command prompt.

```
X509v3 Key Usage: critical
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
X509v3 Subject Key Identifier:
    C8:EB:FA:2C:6D:9D:75:D3:AC:F0:5C:34:5D:57:2B:60:72:03:3D:B4
X509v3 Authority Key Identifier:
    C8:EB:FA:2C:6D:9D:75:D3:AC:F0:5C:34:5D:57:2B:60:72:03:3D:B4
Signature Algorithm: sha512WithRSAEncryption
Signature Value:
    3c:51:f1:26:01:c1:26:9c:f6:d8:bf:a1:c6:d1:d3:c4:ee:a5:
    2f:bf:20:12:5a:bf:f3:a6:81:39:64:d5:9b:38:6a:ed:3e:70:
    61:0d:30:20:e5:0e:eb:7d:fa:71:f0:df:6b:eb:c3:55:c7:4f:
    a9:e4:86:2d:7f:b5:3d:7a:00:ae:44:82:34:ae:48:0c:f9:66:
    bc:42:58:02:47:c4:58:22:1c:80:c4:4c:9c:54:d5:bb:3b:10:
    2b:ca:16:4e:c6:d8:4b:f7:ff:1e:44:2b:8a:c0:d0:e2:3c:e0:
    a2:77:43:83:00:f8:36:82:56:48:66:74:49:7b:3a:39:bf:b9:
    ed:cf:e1:67:fe:3b:95:a2:88:77:e9:9f:ce:b1:a7:35:4b:86:
    05:e3:c6:aa:68:81:e8:a4:59:e2:0b:68:f3:60:2a:27:0c:02:
    95:31:83:b0:1b:b4:26:e0:7e:a6:6f:44:6e:29:bd:6d:08:05:
    1e:59:f4:3d:0c:09:26:b4:51:64:c1:e3:7c:c0:9d:36:a5:84:
    e8:a5:eb:97:d0:4c:3b:04:82:2d:65:bd:2b:34:22:9b:07:22:
    3d:04:fc:ce:99:58:37:31:bf:c7:0f:27:4c:c7:7f:eb:9f:55:
    04:f9:d5:bb:f7:7c:b5:48:20:fa:d4:72:1c:9c:68:e8:d0:5b:
    0e:66:7b:12
```

3.)

Migrating .\Runtime\extcert.fil

Next, use the **tdclientc query_list** command prompt to login. If successful, you will see the highlighted message.

```
C:\Users\vboxuser\Documents\TD322\TDClient>tdclientc query_list

bTrade TDAccess Client Command Line          Version V3.5.0  May 15 2023

Present working directory is [.\]
Using ini-file               [.\tdclient.ini]
Using secondary ini-file     [.\exfer.ini]
Using address book           [.\tpaddrss.ini]
WARNING: Usage information

Your software is not configured with your Email Address
which is required.
Please specify it using the EMAIL_ADDRESS= command-line keyword
or by running the GUI version of the TDClient client.

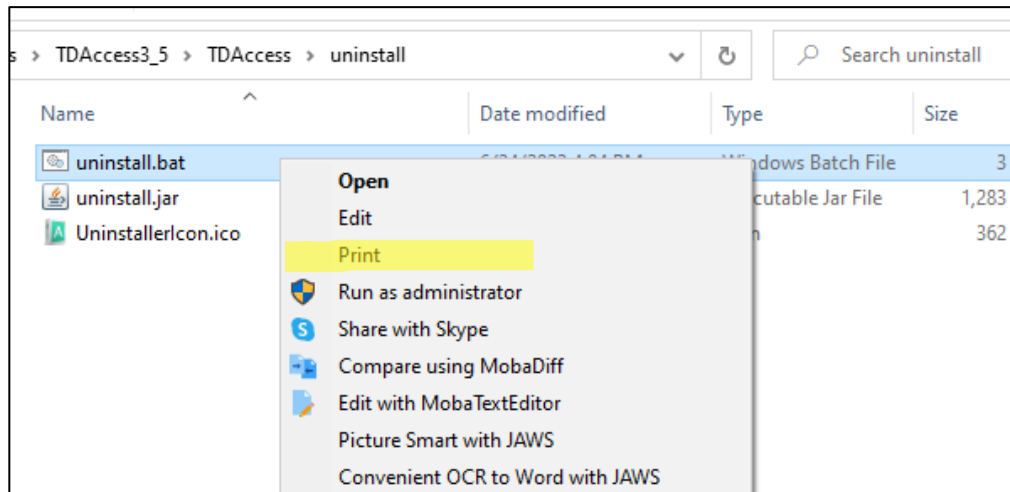
For this program run, will use 'missing-address@my-company.com'

***** Start Communications Session
Connecting to server FTISAIGPORTAL...
200 Command OK.
Connected.
200 Command OK.
200 Command OK.
Logged into FTP server as user TG7701
Obtaining file list from server
226 Transfer complete.
Termination started...
Disconnecting...
221 Goodbye.
```

1.1.7 TDClient Uninstall Process

Under the TDClient directory click open the uninstall folder to access the **uninstall.bat** file. To start the uninstall process right click on the **uninstall.bat** file and **Run as administrator** and select **Yes** to proceed with the uninstall.

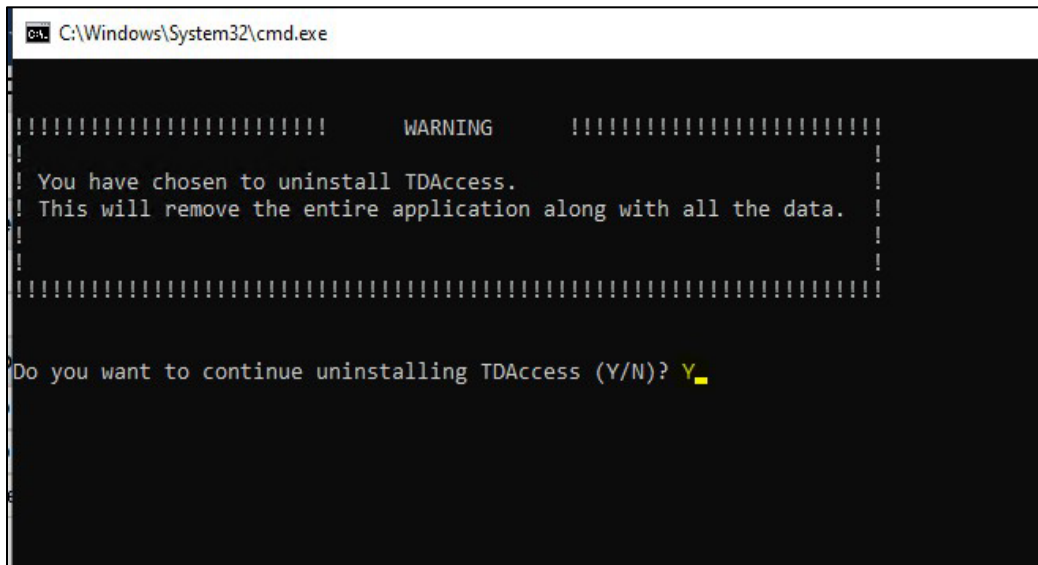
Note: The *uninstall.bat* file must be Run as administrator to uninstall the TDClient.



To continue with the uninstall process, select **Yes** on the pop-up window:



Next, a command line terminal window will display asking whether you would like to continue with the uninstall process. If yes, type Y and then hit enter.

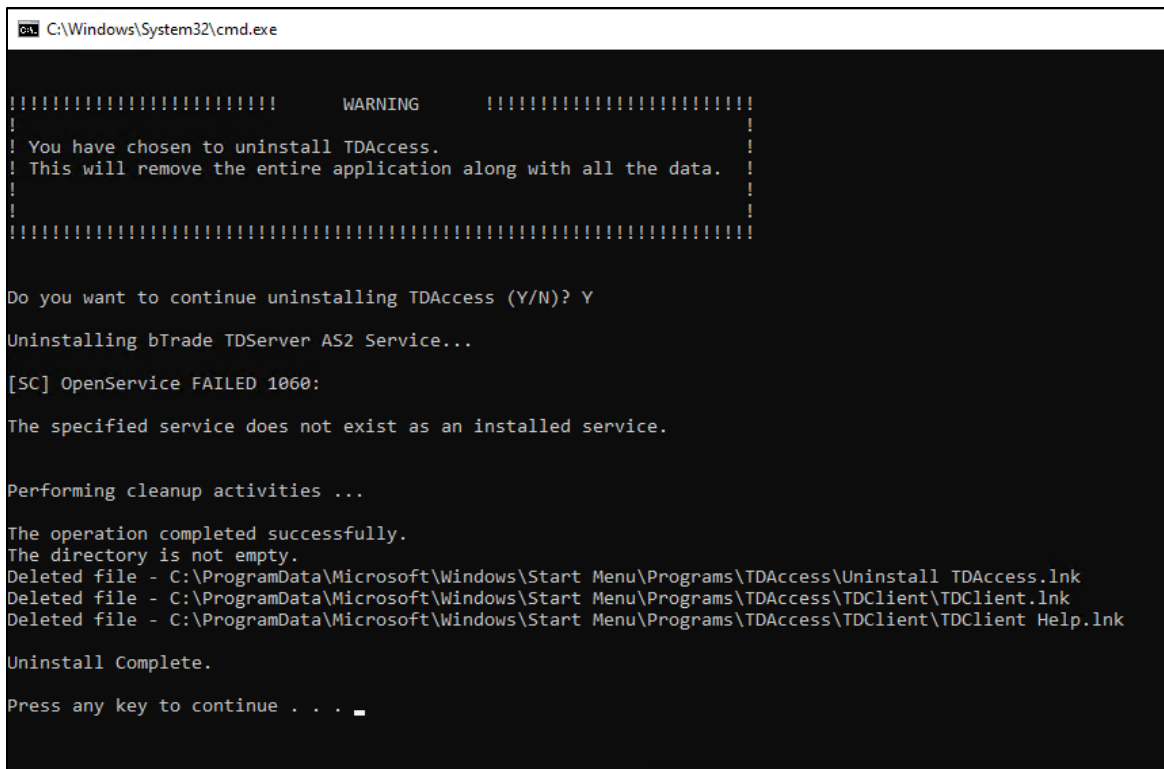


```
C:\Windows\System32\cmd.exe

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! You have chosen to uninstall TDAccess.
! This will remove the entire application along with all the data.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Do you want to continue uninstalling TDAccess (Y/N)? Y_
```

Once the TDClient software is uninstalled the command line output will confirm the operation was completed successfully.



```
C:\Windows\System32\cmd.exe

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! You have chosen to uninstall TDAccess.
! This will remove the entire application along with all the data.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Do you want to continue uninstalling TDAccess (Y/N)? Y

Uninstalling bTrade TDSer AS2 Service...

[SC] OpenService FAILED 1060:
The specified service does not exist as an installed service.

Performing cleanup activities ...

The operation completed successfully.
The directory is not empty.
Deleted file - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TDAccess\Uninstall TDAccess.lnk
Deleted file - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TDAccess\TDClient\TDClient.lnk
Deleted file - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TDAccess\TDClient\TDClient Help.lnk

Uninstall Complete.

Press any key to continue . . .
```

2 Security

2.1 Password Update Procedure

2.1.1 General Information

If this is a newly enrolled mailbox the initial password for access to your Student Aid Internet Gateway (FTI-SAIG) mailbox is set as follows: “ccyynnnn” where “ccyy” is your birth year and “nnnn” represents the last four digits of your social security number. You will be required to change your password the first time you access your mailbox and prior to performing any other activity.

The following rules apply when creating a new FTI-SAIG network password:

- Password must begin with an alpha character.
- Password must be a minimum length of 15 characters.
- Password must contain at least one upper case, one lower case alpha character and one numeric character.
- Password cannot be any of the last five passwords used.
- Password cannot match the User Name or FT number (ignore case).
- Password will be locked out after three failures. (Password will be unlocked after 15 minutes or you can call CPS/SAIG Technical Support to have the password reset.)
- Passwords expire every 90 days, but you can change your password more frequently.

If you have any difficulty establishing your first password, contact CPS/SAIG Technical Support at **800-330-5947** and request to have your FTI-SAIG mailbox password reset.

Note: Network passwords can be changed by either of two methods: via batch job with the TDClient software, or with the FTI-TDCM. This document addresses only the batch process. The FTI-TDCM User Manual containing instructions on the alternate method of updating your password is available on the [Software and Other Tools](#) page of the FSA Knowledge Center website.

2.2 Batch Procedure

We recommend the following procedure for changing passwords every 90 days:

1. Create a separate job for password changes only.
2. We recommend using the **NEW_PASSWD** parameter. Use of the **NEW_PASSWD=password** causes the TDClient to change the network password on the server and will update the encrypted password which is stored in an encrypted form in the TDClient.INI file. Multiple transmissions will not be necessary.
3. Since TDClient will store any password supplied on the command line, we suggest that you maintain a separate tdclient.ini file for testing purposes.
4. Remove the FTPPASSWD= parm from all existing programs and JCL. Removal of this parameter will force TDClient to use the encrypted password that is stored in the tdclient.ini file.

If your FTI-SAIG password expires you will receive the following error in your SYSOUT file or logs:

- WARNING: Logon to server failed
- Login for UserID: FT71504 failed
- (530) FTP login failed. 530 Change password required

Note: Use with MVS zOS TDClient version 3.x or higher to change the network password and to update the *tdclient.ini* file with the new password.

```
//CMDSEND DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx
NEW_PASSWD=newpassword RESET
QUERY_LIST QUERY_FILE=DD:QUERYfilename
```

Figure 5: Example MVS z/OS Password Change

TDClient command line options for UNIX/Linux are NOT case sensitive. Backslashes are being used at the end of each line for line continuation. Double quotes or no quotes can be used in command lines that use the backslashes for line continuation. Do not use single quotes. Change the executable name in the example to the appropriate UNIX client that you have installed.

Note: Use with TDClient version 3.x or higher to change the network password and to update the *tdclient.ini* file with the new password.

```
tdclientc "network=ftisaigportal" ftpuserid=FTxxxxx \
NEW_PASSWD=NEWPASSWORD reset query_list
```

Figure 6: Example UNIX/LINUX Script for Password Change

```
tdclientc RESET "network=ftisaigportal" ftpuserid=ftxxxxx \
ftppasswd=PASSWORD query_list save_only
```

Figure 7: Example UNIX/LINUX Script to Synchronize INI File Password with Current Network password

```
tdclientc RESET "network=ftisaigportal" ftpuserid=FTxxxxx
ftppasswd=PASSWORD query_list save_only
```

Figure 8: Example Windows Command Line to Synchronize INI File Password with Current Network Password

```
tdclientc RESET "network=ftisaigportal" ftpuserid=FTxxxxx
NEW_PASSWD=PASSWORD query_list
```

Figure 9: Example Windows Command Line to Change Network Password and Update the INI File

Note: See sample batch files in the *TDClientWin_bat_examples.zip* file, in the *MAINT* folder.

```
@ECHO OFF
REM This batch file is used to set the TDCLIENT.INI id and password
w/o a network change
REM it can also be used to sync a pwd when a TDCM change was done.
REM enter ID and Password on command line

:: Set default ID, PASSWORD
SET ID=FTXXXXXX
SET PSWD=PASSWORD

:: Use command-line settings if given
IF NOT (%1)==() SET ID=%1
IF NOT (%2)==() SET PSWD=%2

cd ..
@ECHO ON
tdclientc RESET "network=ftisaigportal" ftpuserid=%ID% ftppasswd=%PSWD%
query_list save_only
cd maint
```

**Figure 10: Example Windows Batch File to Synchronize INI File
Password with Current Network password**

Note: See sample batch files in the *TDClientWin_bat_examples.zip* file, in the *MAINT* folder.

```
@ECHO OFF
REM network password change
REM Assumes tdclient.ini and mailbox password are in sync
REM If not use TDCset.bat to sync tdclient.ini and mailbox password
REM
REM Can enter id & pwd on command line or it will use defaults set below

:: Set default ID, PASSWORD
SET ID=FTXXXXXX
SET PSWD=PASSWORD

:: Use command-line settings if given
IF NOT (%1)==() SET ID=%1
IF NOT (%2)==() SET PSWD=%2

cd ..
@ECHO ON
tdclientc RESET "network=ftisaigportal" ftpuserid=%ID% NEW_PASSWD=%PSWD%
query_list
cd maint
```

**Figure 11: Example Windows Batch File to Change Network
Password and Update the INI File**

3 Communication Procedures

3.1 Introduction

This section describes basic procedures for sending and receiving data over the FTI-SAIG.

This section:

- Contains a list of keywords
- Provides examples for sending and receiving data
- Describes the Query List function used to manage the contents of your mailbox(es)
- Documents the record layouts for network headers and trailers

3.2 Command Line Keywords

Command line keywords control the login process as well as what files are to be sent or received. The same commands (keywords) are used on all platforms. The appropriate transfer command lines are combined with the network command line to perform the desired actions as shown in Figure 12.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET  
TRANSFER=(NAME=yourname SENDUSERID=FTxxxxx  
SEND=:inputfilename  
OTHER_COMP_PARS='SECFILE=secfilename')
```

**Figure 12: Network and Transfer Command Lines
for Sending a Single File**

Table 2 lists keywords used in the NETWORK command line to sign on to TDPortal and perform any network activity: sending, receiving, query list or password changes.

Table 2: Network Command Line Keywords

Keyword	Definition
NETWORK	NETWORK=FTISAIGPORTAL is required. This parameter defines the secure network being used for the communications session. This will always be FTISAIGPORTAL to match the network defined in your tdclient.ini file.
FTPUSERID	FTPUSERID=FTxxxxx is required (where “xxxxx” is your five-digit FT number.) This is your FTI-SAIG mailbox user ID. This will not change from what you currently use.
FTPPASSWD	This is the password associated with the FTPUSERID. Note: Once you have reset your password it is stored in the tdclient.ini file in encrypted format and you will no longer need to use the FTPPASSWD parameter in your programs and JCL. See Section 2 for changing and maintaining passwords.
RESET	Instructs TDClient to ignore any previously failed Transfers which would otherwise attempt to restart.
CMDFILE=	Specifies the location of a TDClient command-file to be used. The command-file will completely control the work done during the program run. See section 3.2.2 for a summary of supported transfer keywords and an example. When running the command-line version of TDClient, if no command-file is specified via the CMDFILE= keyword, then the program will assume the work to be done has been previously set up via an editor and is specified in the tdclient.ini & exfer.ini files. See Appendix C for command file usage guidelines and an example of a command file.
SAVE	Instructs TDClient to save the data you specify on the command-line or in the command-file in the tdclient.ini file, causing the data to be permanently in effect until you change it.
SAVE_ONLY	Acts like the SAVE keyword, except the program exits after saving the specified data in the tdclient.ini file. Note: only for versions 3.x and above.

Table 3 lists keywords used in the TRANSFER command line for sending data.

Table 3: Transfer Command Line Keywords for Sending Data

Keyword	Definition
TRANSFER=	This defines the transfer parameters of data being sent. Up to 200 transfers can be CREATED within the command-file. Note: A TRANSFER=NAME= can be saved to the TDCLIENT.EXFER.INI file by using SAVE at the end of the command line. This is helpful when data of the same message class is sent or received on a routine basis. Once a NAME= is saved, on subsequent job submissions you will only need to specify the saved NAME=, and not any of the other TRANSFER commands.
NAME=	Names the transfer being created. The definition will be saved in the TDCLIENT.EXFER.INI file, provided you use the SAVE command. If the name currently exists it will overwrite the current definition.
SENDUSERID=	SENDUSERID is a required parm in the Transfer command line. You must use a valid FT ID as a place holder in this field. Using your own FT ID is highly recommended. Note: Using a FT ID other than your own will cause your job to fail if the FT ID is deleted or inactivated. The O*N05 header record in your data file is the default for the FT ID where the data is being sent.
SEND=	This is the location that the data is being sent from (a UNIX filename or MVS DD name).
SENDCLASS=	SENDCLASS is optional; however, we highly recommend removing this parm from your Transfer command line. The O*N05 header record is the default for the message class. If you use this parm, it will override the default and not use the O*N05 header.
OTHER_COMP_PARMS=	These are compression parameters used only during the compression step for sending data.
SECFILE=	This is a parameter used in OTHER_COMP_PARMS during sending. It contains the location of the secfile definition. This would be a UNIX filename or MVS DD name in JCL. This required parameter provides two functions: The SECFILE defines the position of each parameter in the network headers and trailers. This information is used by TDPortal to separate files and place files in the correct mailboxes; and It forces TDClient to use the SECFILE parameters to send and receive data properly.

SAVE (Optional)	<p>A TRANSFER NAME can be saved to the TDCLIENT.EXFER.INI file by using this keyword at the end of your command line. This is helpful when data of the same message class is sent or received on a routine basis.</p> <p>Example: TRANSFER=(NAME=yourname RECEIVE=outputfilename RECEIVEUSERID=FTxxxxx RECEIVECLASS=messageclass) SAVE</p> <p>Once a NAME= is saved, on subsequent job submissions you will only need to specify the saved NAME=, and not any of the other TRANSFER commands.</p>
SAVE_ONLY (Optional)	<p>Acts like the SAVE keyword, except the program exits after saving the specified data in the tdclient.ini file. Note: only for versions 3.x and above.</p>

Note: See Appendices C for more information on use of the TRANSFER command to control the data you send or receive.

- Appendix A, Example 2, is an example of how to compress a file separately from TDClient prior to sending the file.
- Appendix A, Example 3, is an example of how to send an already compressed file with compression turned off in TDClient.

Example 1: The following displays an example of MVS JCL to SEND Data.

```
//STEP0020 EXEC PGM=EA2KMVSC,REGION=4M,TIME=1000,
//          PARM='CMDFILE=DD:CMDSEND'
//STEPLIB DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//*
//EASYACC DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//*
//EXFER DD DSN=your.dataset.prefix.TDCLIENT.EXFER.INI,DISP=SHR
//*
//CMDSEND DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=yourname SENDUSERID=FTxxxxxx SEND=DD:SENDFILE
```

```

OTHER_COMP_PARMS='SECFILE=DD:SECFILEX')
/*
//SENDFILE      DD DSN=your.send.file,DISP=SHR
/*
//SYSUT1        DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK01        DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02        DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03        DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04        DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS      DD DSN=your.EASTATUS,
//              DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//SECFILEX      DD *
SENDER(FTxxxx);
HEADERLITERAL(O*N05) HEADERSTART(1) RECEIVERSTART(6) RECEIVERLENGTH(14)
CLASSSTART(25) CLASSLENGTH(8);
TRAILERLITERAL(O*N95) TRAILERSTART(1);
LITERAL(O*N01) LITERALSTART(1) DROP(Y);
LITERAL(O*N99) LITERALSTART(1) DROP(Y);
/*
//OUTMSG        DD SYSOUT=*
//SYSPRINT      DD SYSOUT=*
//EAFTPLOG      DD SYSOUT=*
//EALOG         DD SYSOUT=*
//EXFERLOG      DD SYSOUT=*
//COMPLOG       DD SYSOUT=*
//CPFTPLOG      DD SYSOUT=*

```

Note: Insert your own dataset names, FT numbers, and time parameters as appropriate.

Example 2: The following displays an example of UNIX/LINUX Script to SEND Data.

```

tdclientc network=ftisaigportal ftpuserid=FTxxxxx
reset \ "transfer=(name=yourname
senduserid=FTxxxxx
send=/your/send/file.txt
other_comp_parms=secfile=./path/to/your/secfile.txt)"

```

IMPORTANT: Replace the *tdclientc* with the appropriate client name. Backslashes are being used at the end of each line for line continuation. Double quotes or no quotes can be used in command lines that use the backslashes for line continuation. Do not use

single quotes. You can have more than one “transfer=(DATA)” line to transmit multiple files.

Note: See sample batch files in the *TDCClientWin_bat_examples.zip* file, in the *MAINT* folder.

Example 3: The following is an example of Windows Command Line to SEND Data.

```
tdclientc network=ftisaigportal ftpuserid=FTxxxxxx reset  
"transfer=(name=yourname senduserid=FTxxxxxx send=\your\send\file.txt  
other_comp_parms=secfile=.\path\to\your\secfile.txt)"
```

Note: See sample batch files in the *tda3.xx_windows.zip* file, in the *MAINT* folder.

Example 4: The following is an example of Windows Batch File to SEND Data.

```
TRANSFER=(  
    NAME=TDCSend  
    senduserid=FTXXXXXX  
    SEND=outgoing\goodsend.txt  
    OTHER_COMP_PARMS='SECFILE=maint\secfile'  
) SAVE
```

Note: See sample batch files in the *tda3.xx_windows.zip* file, in the *MAINT* folder.

Example 5: The following is an example of Windows Batch File to Transmit Data.

```
@ECHO OFF  
REM Executes transfer statements in the .c files  
  
:: Set default command file.  
SET ID=send.c  
  
:: Use command-line settings if given  
IF NOT (%1)==() SET FILE=%1  
  
cd ..  
@ECHO ON  
TDCCLIENTc.exe "network=ftisaigportal" RESPLOG=TEMP\RESPLOG.TXT  
CMDFILE=MAINT\TRANS\%FILE% RESET  
cd maint
```

Example 6: The following is an example of Send File with Network Headers.

```
O*N05FTxxxxx ,CLS=messclas,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
CPS HEADER 0450HFT00000 20000718150719 0001#C10025002000071
10024720300'01002SAM 008002472031
CPS TRAILER 0450H 20000718150719 000100010#C100250020
O*N95FTxxxxx ,CLS=messclas,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
```

See **Example 1** for parameters SENDUSERID and SENDCLASS in the Command Line Input section. See end of section 4, Header, and Trailer Record Layouts, for specifics in creating O*N05 and O*N95 headers and trailers.

The following are tables contain lists of Input, Temporary and Output logs for send and receive JCL.

Table 4: Input Logs

Input DD	Definition
STEPLIB	Dataset name (your.dataset.prefix.TDLOAD) containing the TDClient program libraries you installed.
EASYACC	Dataset name containing the TDCLIENT.INI file, which contains network access information. – <i>Do NOT alter this file.</i>
EXFER	Dataset name containing the TDCLIENT.EXFER.INI file, which stores your saved TRANSFER commands.
CMDSEND*	Command and Transfer statements to Send data.
CMDRCV**	Command and Transfer statement to Receive data.
SENDFILE*	Location of the input file you want to send from your FTI-SAIG mailbox.
RCVFILE**	Location of the pre-allocated files that will receive data pulled from your FTI-SAIG mailbox.

* Used only on Send Transmissions

** Used only on Receive Transmissions

Table 5: Temporary Logs

Temporary DD	Definition
SYSUT1***	Holds directory listings and copies of compressed data files.
WORK01***	Receives compressed data and decompresses into the Receive file. If using COMPRESS=N then this file is not used.
WORK02***	Temporary storage for Query_List.
WORK03***	Not used
WORK04***	Works in conjunction with Eastatus
EASTATUS	Contains any errors during a send or receive session. Must be an allocated file; cannot be a temporary file.
DCMPLOG**	Logs decompression step for each file received. Indicates if any files failed decompression.

Table 6: Output Logs

Output DD	Definition
SECFILE	This name comes from the "SECFILE=secfilename" in the TRANSFER statement. It stores the SECFILE command lines that tell TDCClient how to format the headers and trailers during compression/decompression.
OUTMSG	Confirms successful: a) login, compression, and send of file, or b) decompression and receipt of files.
SYSPRINT	Logs the stored Transfer processing.
EAFTPLOG	Verifies success or failure of logging onto system and send/receive of files as identified by unique filename. Logs all internal and external FTP activities.
EALOG	General log of the session.
EXFERLOG	Verifies all internal FTP, compression, and decompression activities.
COMPLOG*	Verifies successful compression of data.
CPFTPLOG	Log of all commands and responses to and from the FTP server that is normally used for trouble shooting purposes

* Used only on Send Transmissions

** Used only on Receive Transmissions

*** Temporary work files required by the TDCClient software. They can be defined as temporary files with the following parameters: LRECL= 8192, RECFM=VB, BLKSIZE=0.

Table 7: UNIX/LINUX and Windows Output Logs

Filename	Definition
STDOUT	Confirms successful: a) login, compression, and send of file, or b) decompression and receipt of files.
EA2K.LOG	Verifies success or failure of logging onto system and send/receive of files as identified by unique filename. Logs all internal and external FTP activities.
EACOMM.LOG	General log of the session.
EAXFER.LOG	Verifies all internal FTP, compression, and decompression activities.
COMPRESS.LOG	Verifies successful compression of data.
FTPLOG.TXT	Log of all commands and responses to and from the FTP server that is normally used for trouble shooting purposes

Table 8: Transfer Command Line Keywords for Receiving Data

Keyword	Definition
TRANSFER	Defines the transfer parameters of data being received. Up to 200 transfers can be CREATED within the command-file. Note: A TRANSFER=NAME= can be saved to the TDCLIENT.EXFER.INI file by using SAVE at the end of the command line. This is helpful when data of the same message class is sent or received on a routine basis. Once a NAME= is saved, on subsequent job submissions you will only need to specify the saved NAME=, and not any of the other TRANSFER commands.
NAME=	Names the transfer being created. The definition will be saved in the EXFER.INI file, provided you use the SAVE command. If the name currently exists it will overwrite the current definition.
RECEIVE=	Location where the data will be received (a UNIX filename or a MVS DD name in JCL).
RECEIVEUSERID= (Optional)	This field contains the mailbox ID you are receiving from and is optional. If used without RECEIVECLASS, you will receive all data from the specified RECEIVEUSERID.
RECEIVECLASS= (Optional)	This field contains the message class of the data you want to receive and is optional. If used without RECEIVEUSERID, you will receive all data from the specified RECEIVECLASS. If neither RECEIVEUSERID nor RECEIVECLASS are present, you will receive all data in the mailbox.
OTHER_DECOMP_PARAMS= (Optional)	These are decompression parameters used only during the decompression step for receiving data.
SECFILE= (Used to Send files only)	This is a parameter used in OTHER_DECOMP_PARAMS during sending. It contains the location of the secfile definition. This would be a UNIX filename or MVS DD name in JCL. This required parameter provides two functions: The SECFILE parses the network headers and trailers used by TDPortal to separate files and place files in the correct mailboxes; and It forces TDClient to use the SECFILE parameters to send data properly.
SAVE (Optional)	A TRANSFER NAME can be saved to the TDCLIENT.EXFER.INI file by using this keyword at the end of your command line. This is helpful when data of the same message class is sent or received on a routine basis. Once a NAME= is saved, on subsequent job submissions you will only need to specify the saved NAME=, and not any of the other TRANSFER commands. Example: TRANSFER=(NAME=yourname RECEIVE=outputfilename RECEIVEUSERID=FTxxxxx RECEIVECLASS=messageclass) SAVE
SAVE_ONLY (Optional)	Acts like the SAVE keyword, except the program exits after saving the specified data in the TDCLIENT.INI file. Note: only for versions 3.x and above.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET  
TRANSFER=(NAME=yourname RECEIVE=outputfilename  
RECEIVEUSERID=FTxxxxxx RECEIVECLASS=messageclass)
```

Figure 13: Example Network and Transfer Command Lines for Receiving a Specific Message Class from a Specific Sender (RECEIVEUSERID)

You can modify the commands in many different ways, depending on what data you want to receive. To receive all files of a given message class, specify that message class (EAPS23OP, for example) in the RECEIVECLASS command, but do not include the RECEIVEUSERID command.

- To receive all files from a given sender, specify the RECEIVEUSERID, but do not include the RECEIVECLASS command.
- To receive all files in the mailbox, do not include either the RECEIVEUSERID or RECEIVECLASS.
- To receive files of two separate message classes, issue two separate TRANSFER commands, each with separate RECEIVE=DDs or filenames.

Note: See Appendices A and B for more information on use of the TRANSFER command to control the data, you send or receive.

- Remember, when receiving files from your mailbox, files are received in the order of the query list option.
- Appendix B, Example 8, is an example of how to receive a file with decompression turned off during the TDClient Receive process.
- Appendix B, Example 9 is an example of how to decompress a file separately from TDClient if the file was received with decompression turned off.

Example of JCL to Receive Data is shown below:

```
//STEP0020 EXEC
PGM=EA2KMVSC,REGION=4M,TIME=20,PARM='CMDFILE=DD:CMDRECV'
//*
//STEPLIB DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//*
//EASYACC DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//*
//EXFER DD DSN=your.dataset.prefix.EASYACC.EXFER.INI,DISP=SHR
//*
//CMDRECV DD *
NETWORK=FTI-SAIGPORTAL FTPUSERID=FTxxxx RESET
TRANSFER=(NAME=xxxxxxxxx RECEIVE=DD:RECVFLE
RECEIVEUSERID=FTxxxxx RECEIVECLASS=messageclass)
//*
//RECVFLE DD DSN=your.dataset.receive.file,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=nnnn,BLKSIZE=nnnnn,RECFM=FB),
// SPACE=(CYL,(nn,nn))
//*
//SYSUT1 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK01 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATU DD DSN=your.dataset.prefix.EASTATUS,
S
// DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//*
//DCMPLOG DD SYSOUT=*
//*
//OUTMSG DD SYSOUT=*
//*
//SYSPRIN DD SYSOUT=*
T
//*
//EAFTPLO DD SYSOUT=*
G
//*
//EALOG DD SYSOUT=*
//*
//EXFERLO DD SYSOUT=*
G
//*
//CPFTPLO DD SYSOUT=*
G
//*
```

Note: Insert your own dataset names and FT numbers. When defining the receive file dataset, make sure you have sufficient space allocated and that the record length matches the file you are receiving.

```
tdclientc network=FTISAIGPORTAL ftpuserid=FTxxxxx reset \
"transfer=(name=yourname
receive=./path/to/the/file/to/receive.txt \
receiveuserid=FTxxxxx \
receiveclass=xxxxxxxx) "
```

Figure 14: Example UNIX/LINUX Script to Receive Data

Note: When receiving a file, the receive file must already exist.

```
crtpf file(eal48lib/receive) rcdlen(80) filetype(*src)
```

Note: See sample batch files in the tda3.xx_windows.zip file, in the MAINT folder.

```
tdclientc network=ftisaigportal ftpuserid=FTxxxxx reset
"transfer=(name=yourname receive=.\path\to\the\file\to\receive.txt
receiveuserid=FTxxxxx receiveclass=xxxxxxxx) "
```

Figure 15: Example Windows Command Line to Receive Data

Note: See sample batch files in the tda3.xx_windows.zip file, in the MAINT folder.

```
TRANSFER=(
RECEIVECLASS=GOODSEND
receive=.\incoming\GOODSEND.0001
) SAVE
```

Figure 16: Example Windows Batch File to Receive Data

```
@ECHO OFF
REM Executes transfer statements in the .c files

:: Set default command file.
SET ID=send.c

:: Use command-line settings if given
IF NOT (%1)==() SET FILE=%1

cd ..
@ECHO ON
TDCIENTc.exe "network=ftisaigportal" RESPLOG=TEMP\RESPLOG.TXT
CMDFILE=MAINT\TRANS\%FILE% RESET
cd maint
```

Figure 17: Example Windows Batch File to Transmit Data

Note: See sample batch files in the tda3.xx_windows.zip file, in the MAINT folder.

3.3 Query List & Audit Log

The QUERY_LIST command is used outside of the Transfer statement and supersedes any commands used in the Transfer statement. You can use related keywords, QUERY_FILE= and QUERY_STATUS= to control aspects of this query.

Table 9: Query List and Audit Log

QUERY_LIST	QUERY_FILE=	QUERY_STATUS=	RECEIVE_AUDIT_LOGS
Instructs the client to create and execute a transfer to receive a list of available files from your FTI-SAIG mailbox (FTxxxxx).	Specifies the qualified file name of the file to receive the list. If not present, the file list is written to the default file, <i>list.fil</i> , in the temp directory.	Specifies that the QUERY_LIST should return a list of files with the following specified statuses only: <ul style="list-style-type: none"> AVAILABLE – files that have not been received. RECEIVED – files that have been received. DELETED – files that have been deleted. 	Instructs the client to create and execute a transfer to receive a list of available files and a list of files that were sent from your FTI-SAIG mailbox (FTxxxxx). The status of files in this list are the same as QUERY_LIST and show one additional status called ICFAIL. AUDIT_STATUS= can be used with the values: <ul style="list-style-type: none"> ICFAIL - files that have been rejected by the server. AVAILABLE RECEIVED REJECTED – a file whose status has been manually changed on the FTI-TDCM web site to “rejected”. The parm AUDIT_TYPE= can be used in conjunction with AUDIT_STATUS,

			with values of: <ul style="list-style-type: none"> • SENT – files with a sent status on the TDCM • BOTH – files with a sent or received status
--	--	--	--

Example of Query list for RECEIVED status:

```
2K.01.43\FT50000\FT00000\SARA03OP\U\4.42o.01\O*N05FT00000
,CLS=SARA03OP,BAT=#E300000020020315000000,NCT=00000\SENDFILE\2.03
0\ASCCRLFILOTH\29501\FT00000\29501176151633026581\20010625151607\1\1
523\A\RECEIVED\20010627153549\0\50\50\
```

Note: This is one record of data in a sequential file and each field is delimited with a backslash.

Example of Audit list for ICFAIL status:

```
V2.2.026\FT00000\FT50002\U\Win64\MISSINGHEADER.txt\Bld
04\ASCCRLFILOTH\74915\
FT00000\20080731A00007661156\20080731172151\1\299\A\ICFAIL
\00000000000000\0\50\50\
00000000000000\MSGSPILT\20080731172154\Missing group header record\
V2.2.026\FT00000\FT00000\CORR04IN\U\Win64\O*N05FT00000
,CLS=CORR04IN,XXX, BAT=MISMATCHED BAT,NCT=00000\MISSINGTRAILER.txt\Bld
04\ASCCRLFILOTH\74917\
TG51550\20080731A00007661160\20080731172201\0\301\A\ICFAIL
\00000000000000\0\50\50\
00000000000000\MSGSPILT\20080731172204\Missing group trailer record\
```

The following table is an explanation of each field of the LIST.FIL file created by TDClient when performing a QUERY_LIST (mailbox list) on the FTISAIGPORTAL. Each field is separated by a backslash “\”.

Note: Refer to Appendix B for transmitting by unique file name.

Table 10: Description of Fields for a Query or Audit List

#	Field Name	Description
0	VERSION	EA version; e.g., 3.5 (<i>Send only</i>)
1	SENDER	FT number of the Sender (active user)
2	RECVR	FT number of the Receiver (destination)
3	CLASS	Message Class
4	FORMAT	U = Unformatted
5	SYSTYPE	W95 (indicates all Windows systems), AIX, SUN, OS400, compression version of MVS, HPUX. (<i>Send Only</i>).
6	O*N05 HEADER	Transmission header, O*N05, information
7	ORIGFILENAME	Temporary Send file name from EDIPDS file for MVS only and the original file name for Unix. (<i>Send Only</i>) Or Unique File Number (<i>Receive Only</i>)
8 - 9		
10	SESSIONID	EAFTP generated session ID
11	USERID	User ID of user logged in for session
12	UNIQUEFILENAME	EAFTP generated unique filename
13	PUTDATETIME	EAFTP generated (ctime)
14	PUTDURATION	EAFTP generated ?
15	FILESIZE	EAFTP determined
16		
17	CHARFORMAT	A = ASCII, I = BINARY
18	STATUS	Status of file [RECEIVED, AVAILABLE, DELETED, ICFAIL]
19	GETDATETIME	EAFTP generated (ctime)
20	GETDURATION	EAFTP generated
21 - 27		Not used
28	Restore date/time	Used for AUDIT list ONLY
29	Process Failed In	Used for AUDIT list ONLY
30	Fail date/time	Used for AUDIT list ONLY
31	Error message description	Used for AUDIT list ONLY

Example of Query List JCL:

```
//STEP0020 EXEC PGM=EA2KMVSC,REGION=4M,TIME=100,
//          PARM='CMDFILE=DD:CMDRECV'
//STEPLIB DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//EASYACC DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//EXFER DD DSN=your.dataset.prefix.EXFER.INI,DISP=SHR
//CMDRECV DD *
NETWORK=FTI-SAIGPORTAL FTPUSERID=FTxxxxx RESET
QUERY_LIST QUERY_STATUS=AVAILABLE QUERY_FILE=DD:QUERY
//*
//QUERY DD DSN=your.dataset.name.QUERY.LIST,
//        DISP=(NEW,CATLG),UNIT=SYSDA,
//        DCB=(LRECL=700,BLKSIZE=7000,RECFM=FB),
//        SPACE=(CYL,(30,13)),RETPD=6
//SYSUT1 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=V
//WORK01 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS DD DSN=your.dataset.prefix.EASTATUS,
//        DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB,RETPD=6
//OUTMSG DD SYSOUT=
//SYSPRINT DD SYSOUT=*
//EAFTPLOG DD SYSOUT=*
//EALOG DD SYSOUT=*
//EXFERLOG DD SYSOUT=*
```

Example of UNIX/LINUX Command Line for Query List:

```
TDCLIENTc network=FTISAIGPORTAL RESET query_list
QUERY_FILE="./maint/query/logs/query_default.txt"
cd maint\query
```

Example of Windows Command Line for Query List:

```
TDCLIENTc network=FTISAIGPORTAL RESET query_list
QUERY_FILE =".\maint\query\logs\query_default.txt"
cd maint\query
```

Example of RECEIVE_AUDIT_LOGS:

```
//STEP0020 EXEC PGM=EA2KMVSC,REGION=4M,TIME=100,
//          PARM='CMDFILE=DD:CMDRECV'
//STEPLIB DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//EASYACC DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//EXFER DD DSN=your.dataset.prefix.EXFER.INI,DISP=SHR
//CMDRECV DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
RECEIVE_AUDIT_LOGS
//*
//AUDITLOG DD DSN=your.dataset.name,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=700,BLKSIZE=7000,RECFM=FB),
//          SPACE=(CYL,(30,13)),RETPD=6
//SYSUT1 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=V
//WORK01 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS DD DSN=your.dataset.prefix.EASTATUS,
//          DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          LRECL=8192,BLKSIZE=0,RECFM=VB,RETPD=6
//OUTMSG DD SYSOUT=
//SYSPRINT DD SYSOUT=*
//EAFTPLOG DD SYSOUT=*
//EALOG DD SYSOUT=*
//EXFERLOG DD SYSOUT=*
```

Example of UNIX/LINUX Command Line for RECEIVE_AUDIT_LOGS:

Note: Instead of issuing a file name for RECEIVE_AUDIT_LOGS, you will create a DD name called AUDITLOG. The TDClient client is programmed to look for a dataset name called AUDITLOG.

```
TDCLIENTc network=FTISAIGPORTAL RESET AUDIT_TYPE=SENT
RECEIVE_AUDIT_LOGS AUDIT_FILE="./maint/query/logs/AUDIT_sent_default.txt" cd maint/query
```

Example of Windows Command Line for RECEIVE_AUDIT_LOGS

```
TDCLIENTc network=FTISAIGPORTAL RESET AUDIT_TYPE=SENT RECEIVE_AUDIT_LOGS
AUDIT_FILE=".\maint\query\logs\AUDIT_sent_default.txt"
cd maint\query
```

3.4 File and Transmission Header & Trailer Record Layouts

The use of **O*N05 and O*N95 header and trailer records is required.** Transmission Header (O*N05) and Transmission Trailer (O*N95) records wrap the input data for each destination mailbox and message class. Thus, each transmission will contain a minimum of one header and one trailer record for send files.

The first record in the file is the Transmission Header ('O*N05') record. Your data follows the Transmission Header and after your data the Transmission Trailer ('O*N95') record follows. Transmission Header and Transmission Trailer records identify the input data for each destination mailbox and message class. (See the example below)

Note: All header and trailer records are required to be a minimum record length of 70 characters. These are examples only and may require customization at your site.

Example of O*N05 and O*N95 header and trailer:

```
O*N05FTxxxxx      ,CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
your input data for first batch is inserted here
your input data for first batch is inserted here
your input data for first batch is inserted here
O*N95FTxxxxx      ,CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
O*N05FTxxxxx      ,CLS=MSGCLASS,XXX,BAT=,
your input data for second batch is inserted here
your input data for second batch is inserted here
your input data for second batch is inserted here
your input data for second batch is inserted here
O*N95FTxxxxx      ,CLS=MSGCLASS,XXX,BAT=,
```

3.4.1 Transmission Header (O*N05) & Trailer Record (O*N95) Layouts (Required)

The Transmission Header record identifies the beginning of a group of input data records destined for an FTI-SAIG mailbox. The Transmission Trailer record identifies the end of this group of records. See Table 11 for the required record layout of the Transmission Headers and Trailers.

The Transmission Header and Transmission Trailer records require these substitutions:

- **Record Identifier** – Use O*N05 for Transmission Header; use O*N95 for Transmission Trailer.
- **Destination Mailbox ID** – The Mailbox ID of who is to receive the data when you are sending; or the Mailbox ID of the sender when you are receiving. See application-specific guides and references for the correct destination mailbox for each message class.
- **CLS=Message Class** – The Message Class of the data you are sending. An eight-character label assigned to a particular type of data by the application system.
- **BAT=, –** Represents the Batch ID or Document ID for the batch the TDClient user is sending. The parameter “BAT=” and the ending comma “,” are required.
 - **Example 1:** If you do not want to include the full batch ID use **BAT=,**
 - **Example 2:** If you choose to populate this field with the full batch ID insert “BAT=” followed by the batch ID and end with a comma “,”. As displayed in the examples below:
 - **BAT=#D300018620030816120145,** or
 - **BAT=2003-10-21T16:40:19.3092722120,**

Note: the batch ID should not exceed 50 characters.

Table 11: Transmission Header & Trailer Record Layouts

Column	Length	Entry
1-5	5	Record Identifier <i>(Required)</i> Use O*N05 for Transmission Header and O*N95 for Transmission Trailer.
6-12	7	Destination Mailbox ID <i>(Required)</i> This field must have the same value on both the header and trailer record.
13-19	7	SPACES <i>(Required)</i>
20	1	',' (Comma) <i>(Required)</i>
21-24	4	CLS = <i>(Required)</i>
25-32	8	Message Class <i>(Required)</i> This field must have the same value on both the header and trailer record.
33	1	',' (Comma) <i>(Required)</i>
34-37	4	XXX, <i>(Required)</i>
38-41	4	BAT= <i>(Required)</i>
42-91 Variable	1-50 Variable	Up to a fifty-character batch number. If using all 50 characters then adjust the remainder of the record layout accordingly. If no batch number then comma must follow the equal sign. Note: A comma must follow this number. (See next field).
42-92 Variable (Based on length of Batch Number.)	1	',' (Comma) <i>(Required)</i> Note: Must come directly after the Batch Number. If no batch number, then a comma must follow the BAT equal. E.g., BAT=,
93-EOR Variable, starts after comma.	Variable	The minimum record length required is 70. This can be spaces or any data applicable to your institution (such as NCT=).

The Transmission Header and Trailer records described above must be used with all data. **Except for the Record Identifier in positions 1-5, both the O*N05 and O*N95 records must match exactly from position 6 through the end of the batch number comma that starts in position 43+.**

Note: All header and trailer records are required to be a minimum record length of 70 characters.

4 Appendix A: Command Lines for Different Methods of Sending Data

1. Sending multiple batches of data in one file requires that you have multiple sets of O*N05 and O*N95 transmission headers and trailers around each batch within the file. You must use the O*N05FTxxxxx in the header and O*N95FTxxxxx in the trailer to specify the destination point for each batch of data. See Chapter 4, *Header, and Trailer Record Layouts*, for specifics.
2. Since FTPASSWD is stored in the tdclient.ini file in encrypted format, it is not necessary to hardcode your password in all of your scripts and/or JCL. See [Section 2](#) for recommendation to change your password.

Note: These are examples only and may require customization at your site.

Example 1: Sample of multiple Transfer statements to send multiple batches in one file. You must specify a corresponding input file for each Transfer statement.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=name1 SENDUSERID=FTxxxxxx SEND=FILENAM1
  OTHER_COMP_PARMS='SECFILE=secfilename')
TRANSFER=(NAME=name2 SENDUSERID=FTxxxxxx SEND=FILENAM2
  OTHER_COMP_PARMS='SECFILE=secfilename')
```

Example 2: Sample JCL to compress a file prior to the TDClient step. See example 3 to send the compressed file with compression turned off in TDCClient.

```
//STEP0010 EXEC PGM=COMPRESS,REGION=4M,TIME=1440,
//          PARM='FILTER ASCII CRLF SECFILE=DD:SECFILX'
//STEPLIB DD DSN=FTISAIG.EASYACC.LOADLIB,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//COMPLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DATAIN DD
//DATAOT DSN=FTISAIG.UNCOMP.EAPS03IN,DISP=SHR
// DD DSN=FTISAIG.COMP.EAPS03IN,
// DISP=(NEW,CATLG),RETPD=30,
// SPACE=(CYL,(60,60),RLSE),UNIT=SYSDA,
//*
//SECFILX DD*
HEADERLITERAL(O*N05) HEADERSTART(1) RECEIVERSTART(6)
RECEIVERLENGTH(14);
TRAILERLITERAL(O*N95) TRAILERSTART(1);
LITERAL(O*N01) LITERALSTART(1) DROP(Y);
LITERAL(O*N99) LITERALSTART(1) DROP(Y);
```

Example 3: Sample JCL to send a compressed file with compression turned off in the TDClient step. See example 2, above, to compress a file prior to the TDClient step.

```
//STEP0020 EXEC PGM=EA2KMVSC,REGION=4M,TIME=1440,
//          PARM='CMDFILE=DD:CMDSEND'
//STEPLIB DD DSN=FTISAIG.EASYACC.LOADLIB,DISP=SHR
//*
//EASYACC DD DSN=FTISAIG.TDCLIENT.INI,DISP=SHR
//*
//EXFER DD DSN=FTISAIG.EASYACC.EXFER.INI,DISP=SHR
//*
//CMDSEND DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=comp SENDUSERID=FTxxxxxx SEND=DD:SENDFILE
COMPRESS=N)
//*
//SENDFILE DD DSN=FTISAIG.D110501.FILTER,DISP=SHR
//*
//WORK04 DD DISP=NEW,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS DD DSN=FTISAIG.EASTATUS,
//        DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(5,5)),
//        LRECL=8192,BLKSIZE=0,RECFM=VB,RETPD=60
//OUTMSG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//EAFTPLOG DD SYSOUT=*
//CPFTPLOG DD SYSOUT=*
```

Example 4: Sample JCL to pre-define files required by TDClient prior to sending.

```

//*****
//*          Run IEBGENER to create your SYSUT1 file
*
//*****
//SYSUT1 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.SYSUT1.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//*          Run IEBGENER to create your SYSUT2 file
*
//*****
//SYSUT2 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.SYSUT2.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//*          Run IEBGENER to create your WORK01 file
*
//*****
//WORK01 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.WORK01.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//*          Run IEBGENER to create your WORK02 file
*
//*****
//WORK02 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.WORK02.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//*          Run IEBGENER to create your WORK03 file
*
//*****
//WORK03 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY

```

```

//SYSUT2 DD DSN=your.dataset.WORK03.file,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
// SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//* Run IEBGENER to create your WORK04 file
*
//*****
//WORK04 EXEC PGM=IEBGENER

//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.WORK04.file, //
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
// SPACE=(TRK,(5,5))
//SYSIN DD DUMMY
//*****
//* Run IEBGENER to create your EASTATUS file
*
//*****
//EASTATUS EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=your.dataset.EASTATUS.file,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
// SPACE=(CYL,(5,5))
//SYSIN DD DUMMY
//*****
//* Run EA2KMVSC To Send
*
//*****
//STEP0020 EXEC
// PGM=EA2KMVSC,REGION=4M,PARM='CMDFILE=DD:CMDSND'
//*
//STEPLIB DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//*
//EASYACC DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//*
//EXFER DD
// DSN=your.dataset.prefix.TDCLIENT.EXFER.INI,DISP=SHR
//*
//CMDSND DD *
NETWORK=FTI-SAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=xxxxxxxxxx SEND=DD:SENDFILE
SENDUSERID=FTxxxxxx OTHER_COMP_PARMS='SECFILE=DD:SECFILE')
//*
//SENDFILE DD DSN=your.dataset.SEND.file,DISP=SHR
//SYSUT1 DD DSN=your.dataset.SYSUT1.file,
// DISP=(OLD,DELETE,DELETE) ←=== CAN KEEP
//SYSUT2 DD DSN=your.dataset.SYSUT2.file,

```

```
//
//                                DISP=(OLD,DELETE,DELETE)    ←=== CAN KEEP
//WORK01 DD                      DSN=your.dataset.WORK01.file,
//                                DISP=(OLD,DELETE,DELETE)    ←=== CAN KEEP
//WORK02 DD                      DSN=your.dataset.WORK02.file,
//                                DISP=(OLD,DELETE,DELETE)    ←=== CAN KEEP
//WORK03 DD                      DSN=your.dataset.WORK03.file,
//                                DISP=(OLD,DELETE,DELETE)    ←=== CAN KEEP
//                                DISP=(OLD,DELETE,DELETE)
//
//WORK04 DD                      DSN=your.dataset.WORK04.file,
//                                DISP=(OLD,DELETE,DELETE)    ←=== CAN KEEP
//EASTATUS DD                    DSN=your.dataset.EASTATUS.fil
//                                e,                            ←=== CAN KEEP
//SECFILEX DD *                  DISP=(OLD,DELETE,DELETE)
```

SENDER(FTxxxxx);
 HEADERLITERAL(O*N05) HEADERSTART(1) RECEIVERSTART(6) RECEIVERLENGTH(14)
 CLASSSTART(25) CLASSLENGTH(8); TRAILERLITERAL(O*N95)
 TRAILERSTART(1); LITERAL(O*N01) LITERALSTART(1) DROP(Y);
 LITERAL(O*N99) LITERALSTART(1) DROP(Y);
 /**
 //COMPLOG DD SYSOUT=*

```
//OUTMSG DD SYSOUT=*
```

```
//SYSPRINT DD SYSOUT=*
```

```
//EAFTPLOG DD SYSOUT=*
```

```
//EALOG DD SYSOUT=*
```

```
//EXFERLOG DD SYSOUT=*
```

```
//CPFTPLOG DD SYSOUT=*
```

```
/**
```

5 Appendix B: Command Lines for Different Methods of Receiving Data

Since FTPPASSWORD is stored in the tdclient.ini file in encrypted format when you change your password, it is not necessary to hardcode your password in all of your scripts and/or JCL. See [Section 2](#) for recommendation to change your password.

Note: These are examples only and may require customization at your site.

Example 1: Receive all data by specific sender ID: **RECEIVEUSERID=** the sender of the data being requested.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=name RECEIVE=name RECEIVEUSERID=FTxxxx)
```

Example 2: Receive all data by specific message class:

```
RECEIVECLASS= the message class of the data being
requested.NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=name RECEIVE=name RECEIVECLASS=messclass)
```

Example 3: Receive all data in mailbox: Notice that **RECEIVEUSERID=** and **RECEIVECLASS=** have been removed.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=name RECEIVE=name)
```

Example 4: Receive data by unique file name: **RECEIVE_SERVER_FILE=** the Unique Filename on FTI-TDCM or the Available status record within your Query List.

```
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=name RECEIVE=name
RECEIVE_SERVER_FILE=xxxxxxxxxxxxxxxxxxxxxx)
```

Example 5: Receive multiple files by specific message class using multiple TRANSFER statements. Specify **RECEIVECLASS=** for the message class of the data being requested. You must specify a corresponding output file for each Transfer statement. We recommend that you specify the exact record length if receiving fixed block data.

```

NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=nam1 RECEIVE=RECVFL1 RECEIVECLASS=SARA02OP)
TRANSFER=(NAME=nam2 RECEIVE=RECVFL2 RECEIVECLASS=CORR02OP)

```

Example 7: Concatenate the O*N01 File Header and O*N99 File Trailer records into your Receive data file using IEBGENER. Sample for users whose programs require the O*N01 & O*N99 records.

```

//STEP01      EXEC PGM=IEBGENER
//SYSPRINT    DD SYSOUT=*
//SYSUT1      DD DSN=your.dataset.prefix.IEBGEN01,DISP=SHR
//SYSUT2      DD DSN=your.dataset.prefix.IEBGALL.RECV1,
//              DISP=(MOD,CATLG),UNIT=SYSDA,
//              DCB=(LRECL=nnnn,BLKSIZE=nnnnn,RECFM=FB),
//              SPACE=(CYL,(n,n))
//SYSIN       DD DUMMY
//*
//STEP02 EXEC PGM=EA2KMVSC,REGION=4M,PARM='CMDFILE=DD:CMDRECV'
//*
//STEPLIB     DD DSN=your.dataset.prefix.TDLOAD,DISP=SHR
//*
//EASYACC     DD DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
//*
//EXFER       DD DSN=your.dataset.prefix.TDCLIENT.EXFER.INI,DISP=SH
//*
//CMDRECV DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxx RESET
TRANSFER=(NAME=name RECEIVE=DD:receive RECEIVEUSERID=FTxxxxx)
//*
//receive     DD DSN=your.dataset.prefix.IEBGALL.RECV1,
//              DISP=(MOD,CATLG),UNIT=SYSDA,
//              DCB=(LRECL=nnnn,BLKSIZE=nnnnn,RECFM=FB),
//              SPACE=(CYL,(nn,nn))
//*
//SYSUT1      DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK01      DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02      DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03      DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04      DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS    DD DSN=your.dataset.prefix.EASTATUS,
//              DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(5,5)),
//              LRECL=8192,BLKSIZE=0,RECFM=VB
//*
//DCMPLOG     DD SYSOUT=*
//OUTMSG      DD SYSOUT=*
//SYSPRINT    DD SYSOUT=*
//EAFTPLOG    DD SYSOUT=*
//EALOG       DD SYSOUT=*

```

```
//EXFERLOG DD SYSOUT=*
//STEP03 EXEC PGM=IEBGENER,COND=(00,NE,STEP02)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD
DSN=your.dataset.prefix.IEBGALL.RECV1,DISP=(OLD,PASS)
// DD DSN=your.dataset.prefix.IEBGEN99,DISP=SHR
//SYSUT2 DD DSN=your.dataset.prefix.IEBGNOC.RECV6,
// DISP=(,CATLG),UNIT=SYSDA,
// DCB=(your.dataset.prefix.IEBGALL.RECV1),
// SPACE=(CYL,(nn,nn))
//SYSIN DD DUMMY
//*
//STEP04 EXEC PGM=IEFBR14,COND=(00,NE,STEP03)
//FILE1 DD DSN=your.dataset.prefix.IEBGALL.RECV1,
// DISP=(MOD,DELETE),
// SPACE=(CYL,(nn,nn))
```

Example 8: Sample JCL to receive a file with decompression turned off. See example 9 to decompress file later.

Note: These are examples only and may require customization at your site.

```
//STEP0020 EXEC
PGM=EA2KMVSC,REGION=4M,PARM='CMDFILE=DD:CMDRECV'
//*
//STEPLIB DD DSN=FTISAIG.EASYACC.LOADLIB,DISP=SHR
//EASYACC DD DSN=FTISAIG.D081501.APPSYS.TDCLIENT.INI,DISP=SHR
//EXFER DD DSN=FTISAIG.TDCLIENT.EXFER.INI,DISP=SHR
//*****
/* THESE COMMANDS WILL RECEIVE MULTIPLE FILES AND PUT INTO
THE
/* RECEIVE FILE. BUT, WHEN YOU DECOMPRESS THE FILE YOU MUST
/* HAVE THE EXACT DECOMPRESS LRECL OR THE O*N05 WILL NOT
BEGIN IN
/* POSITION 1. IT WILL INSTEAD BE AT THE END OF O*N95.
//*****
//CMDRECV DD *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=COPY RECEIVE=DD:RECV
OTHER_DECOMP_PARMS='COPYONLY')
/*
//RECV DD DSN=FTI-SAIG.COMP.CORR03OP,
// DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
/*
//SYSUT1 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//SYSUT2 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
// LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK01 DD DSN=FTISAIG.WORK01.D112701.ALL,
// DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(TRK,(5,5)),
```

```
//      LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK02 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//      LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK03 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//      LRECL=8192,BLKSIZE=0,RECFM=VB
//WORK04 DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//      LRECL=8192,BLKSIZE=0,RECFM=VB
//EASTATUS DD SYSOUT=*
//OUTMSG DD SYSOUT=*
//DCMPLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//EAFTPLOG DD SYSOUT=*
//
```

Example 9: Sample JCL to decompress a file that has already been received by TDClient with decompression turned off. When you decompress the file, you must have the exact decompressed record length of the file or the O*N05 record will not begin in position one. It will instead be at the end of the O*N95 record.

Note: These are examples only and may require customization at your site.

```
//STEP0010 EXEC PGM=DECOMP,REGION=4M,TIME=1440,
//      PARM='APPEND UNCOMP'
//STEPLIB DD
DSN=FTISAIG.EASYACC.LOADLIB,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DCMPLOG DD SYSOUT=*
//DATAIN DD
//DATAOT DSN=FTISAIG.COMP.CORR03OP,DISP=SHR
//      DD DSN=FTI-
//      SPACE=(CYL,(60,60)),UNIT=SYSDA,
//      LRECL=2850,BLKSIZE=0,RECFM=FB
```

Example 10: Sample JCL to pre-defined files required by TDClient prior to receiving.

```

//*****
//*      Run IEBGENER to create your RECVFLE file
//*****
//RECVFLE  EXEC  PGM=IEBGENER
//SYSUT1   DD    DUMMY
//SYSUT2   DD    DSN=your.dataset.receive.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=nnnn,BLKSIZE=nnnn,RECFM=FB),
//          SPACE=(CYL,(nn,nn))
//SYSIN     DD    DUMMY
//*****
//*      Run IEBGENER to create your SYSUT1 file
//*****
//SYSUT1   EXEC  PGM=IEBGENER
//SYSUT1   DD    DUMMY
//SYSUT2   DD    DSN=your.dataset.SYSUT1.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN     DD    DUMMY
//*****
//*      Run IEBGENER to create your SYSUT2 file
//*****
//SYSUT2   EXEC  PGM=IEBGENER
//SYSUT1   DD    DUMMY
//SYSUT2   DD    DSN=your.dataset.SYSUT2.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN     DD    DUMMY
//*****
//*      Run IEBGENER to create your WORK01 file
//*****
//WORK01   EXEC  PGM=IEBGENER
//SYSUT1   DD    DUMMY
//SYSUT2   DD    DSN=your.dataset.WORK01.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN     DD    DUMMY
//*****
//*      Run IEBGENER to create your WORK02 file
//*****
//WORK02   EXEC  PGM=IEBGENER
//SYSUT1   DD    DUMMY
//SYSUT2   DD    DSN=your.dataset.WORK02.file,
//          DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//          SPACE=(TRK,(5,5))
//SYSIN     DD    DUMMY
//*****

```

```

/**          Run IEBGENER to create your WORK03 file
/*****
//WORK03      EXEC   PGM=IEBGENER
//SYSUT1      DD     DUMMY
//SYSUT2      DD     DSN=your.dataset.WORK03.file,
//              DISP=(NEW,CATLG),UNIT=SYSDA,
//              DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB), //
SPACE=(TRK,(5,5))
//SYSIN       DD     DUMMY
/*****
/**          Run IEBGENER to create your WORK04 file
/*****
//WORK04      EXEC   PGM=IEBGENER
//SYSUT1      DD     DUMMY
//SYSUT2      DD     DSN=your.dataset.WORK04.file,
//              DISP=(NEW,CATLG),UNIT=SYSDA,
//              DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//              SPACE=(TRK,(5,5))
//SYSIN       DD     DUMMY
/*****
/**          Run IEBGENER to create your EASTATUS file
/*****
//EASTATUS    EXEC   PGM=IEBGENER
//SYSUT1      DD     DUMMY
//SYSUT2      DD     DSN=your.dataset.EASTATUS.file,
//              DISP=(NEW,CATLG),UNIT=SYSDA,
//              DCB=(LRECL=8192,BLKSIZE=0,RECFM=VB),
//              SPACE=(CYL,(5,5))
//SYSIN       DD     DUMMY
/*****
/**          Run EA2KMVSC To Receive
/*****
//STEP0020    EXEC   PGM=EA2KMVSC,REGION=4M,PARM='CMDFILE=DD:CMDRECV'
/**

//STEPLIB     DD     DSN=your.dataset.prefix.TDLOAD,DISP=SHR
/**
//EASYACC     DD     DSN=your.dataset.prefix.TDCLIENT.INI,DISP=SHR
/**
//EXFER       DD
DSN=your.dataset.prefix.TDCLIENT.EXFER.INI,DISP=SHR
/**
//CMDRECV     DD     *
NETWORK=FTISAIGPORTAL FTPUSERID=FTxxxxxx RESET
TRANSFER=(NAME=xxxxxxxxxx RECEIVE=DD:RCVFLE
RECEIVEUSERID=FTxxxxxx RECEIVECLASS=messageclass)
/**
//RCVFLE      DD     DSN=your.dataset.receive.file,DISP=SHR
/**
//SYSUT1      DD     DSN=your.dataset.SYSUT1.file,
//              DISP=(OLD,DELETE,DELETE)
//SYSUT2      DD     DSN=your.dataset.SYSUT2.file,

```

←=== CAN Keep

//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//WORK01	DD	DSN= your.dataset.WORK01.file ,	
//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//WORK02	DD	DSN= your.dataset.WORK02.file ,	
//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//WORK03	DD	DSN= your.dataset.WORK03.file ,	
//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//WORK04	DD	DSN= your.dataset.WORK04.file ,	
//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//EASTATUS	DD	DSN= your.dataset.EASTATUS.file ,	
//		DISP=(OLD,DELETE,DELETE)	←=== CAN Keep
//DCMPLOG	DD	SYSOUT=*	
//OUTMSG	DD	SYSOUT=*	
//SYSPRINT	DD	SYSOUT=*	
//EAFTPLOG	DD	SYSOUT=*	
//EALOG	DD	SYSOUT=*	
//EXFERLOG	DD	SYSOUT=*	
//*			

6 Appendix C: Command File Usage and Transfer Statements

See below for considerations while using the command-file specified using the **CMDFILE=** command-line keyword as the TDCClient program is being invoked.

1. Text comments to the right of any '#' character are ignored unless the # occurs with text offset by single or double quotes.
2. Blank lines are ignored.
3. Command-line and command-file parameters can be delimited using single or double quotes, parentheses, or square or curly braces. Sub-expressions can be delimited within main expressions using a different delimiter. For example, "TRANSFER=(name='my transfer'... OTHER_COMP_PARMS='parm1 parm2' ...)"
4. Spaces are used only to separate keyword/value pairs and are otherwise ignored unless they are within a delimited expression. For example:
5. TRANSFER= "My transfer" is the same as TRANSFER="My Transfer" (case), but TRANSFER = "My Transfer" is illegal (TRANSFER= is the keyword – no spaces allowed within the keyword itself), and TRANSFER= My Transfer is also illegal (it is saying to use transfer-name "My", not "My Transfer") because there are no quotes around the transfer-name.
6. The end-of-line has no special significance. This means that you can put all your keywords on one line or spread them out across multiple lines (see examples below).
7. The keywords are NOT case-sensitive, but the values may be, depending on the server with which you are communicating. For example, transfer= and Transfer= are the same as TRANSFER=, but the server may not agree that SENDUSERDID=FTXXXXX is the same as SENDUSERDID=ftXXXXX.
8. Up to 200 transfers may be CREATED within the command-file. Each transfer created is added to the list of transfers to be run.
9. Up to 200 EXISTING transfers may be specified in the command-file. Each specified existing transfer is added to the list of transfers to be run.

The following is an example of command for Windows:

```
TDCLIENTc.exe."network=ftisaigportal" ftpuserid=FTxxxxx  
CMDFILE=MAINT\TRANS\CMDFILENAME RESET
```

Command file with send and receive transfer statements.

```
TRANSFER=(NAME=SEND1 SENDUSERID=FT00000
SEND=OUTGOING\FT00000.TXT
OTHER_COMP_PARMS='SECFILE=MAINT\HEADER.DEF')
TRANSFER=(NAME=SEND2 SENDUSERID=FT00000
SEND=OUTGOING\FT00000.TXT
OTHER_COMP_PARMS='SECFILE=MAINT\HEADER.DEF')
TRANSFER=(NAME=SEND3 SENDUSERID=FT00000
SEND=OUTGOING\FT00000.TXT
OTHER_COMP_PARMS='SECFILE=MAINT\HEADER.DEF')
TRANSFER=(NAME=RECV1 RECEIVECLASS=GOODSEND
RECEIVE=INCOMING\GOODSEND1.0000 UNCOMP=Y APPEND=N AUTOEXT=4)
TRANSFER=(NAME=RECV2 RECEIVECLASS=GOODSEND
RECEIVE=INCOMING\GOODSEND1.0000 UNCOMP=Y APPEND=N AUTOEXT=4)
```

For send Command the following configurations are needed for the SECFILE:

```
HEADERLITERAL(O*N05) HEADERSTART(1) RECEIVERSTART(6) RECEIVERLENGTH(14)
CLASSSTART(25) CLASSLENGTH(8);
TRAILERLITERAL(O*N95) TRAILERSTART(1);
```

Figure 18: Example SECFILE

IMPORTANT: Pay close attention to the placement of semicolons and spaces, or errors. The SECFILE is required when sending data.

- HEADERSTART(1) indicates the O of the O*N05 to start in the first position.
- RECEIVERSTART(6) indicates that the receiver ID of a file being sent begins in the 6th position of the N05 header.
- RECEIVERLENGTH(14) indicates the length of the receiver ID field, including spaces.
- Using the CLASS options in the SECFILE forces TDCClient to use the message class (CLS=) contained in the O*N05 header record.
 - CLASSSTART(25) indicates that the message class begins in the 25th position of the N05 header.
 - CLASSLENGTH(8) indicates that the CLS= field is 8 positions long.
- All SECFILE parameters referring to Headers and Trailers that define the network headers and trailers are required.
 - The O*N05 Transmission Header and the O*N95 Transmission Trailer surround each set of application system headers and trailers in the data file being sent.

7 Appendix D: UNIX/LINUX Scripts for Different Methods of Receiving Data

These following commands are examples only and may require customization at your site.

Example 1: Receive data by a specific sender ID, RECEIVEUSERID= the sender of the data being requested. Depending on options in the TDCClient.INI file it will either all be concatenated into one file (if APPEND=Y and AUTOEXT=N) or each item will go into separate files named receive.txt.xxxx, where xxxx can be any number from 0001 to 9999 (if APPEND=N and AUTOEXT=4.)

```
tdclienttc network=ftisaigportal ftpuserid=FTxxxxxx reset\  
"transfer=(name=xxxxxx  
receive=./path/to/the/file/to/receive.txt  
\receiveuserid=FTxxxxxx)"
```

Example 2: Receive all data by a specific message class, RECEIVECLASS= the message class of the data being requested. Depending on options in the TDCClient.INI file it will either all be concatenated into one file (if APPEND=Y and AUTOEXT=N) or each item will go into separate files named receive.txt.xxxx, where xxxx can be any number from 0001 to 9999 (if APPEND=N and AUTOEXT=4.)

```
tdclienttc network= ftisaigportal ftpuserid=FTxxxxxx reset\  
"transfer=(name=xxxxxxxx  
receive=./path/to/the/file/to/receive.txt  
\receiveclass=xxxxxxxxxx)"
```

Example 3: Receive data by unique file name: RECEIVE_SERVER_FILE= the Unique Filename on TDCM or the Available status record within your Query List.

```
tdclienttc network= ftisaigportal ftpuserid=FTxxxxxx reset\  
"transfer=(name=xxxxxx  
receive=./path/to/the/file/to/receive.txt  
\receive_server_file= xxxxxxxxxxxxxxxxxxxxx)"
```

Example 4: Receive all data in mailbox: Notice that RECEIVEUSERID= and RECEIVECLASS= have been removed.

```
tdclienttc network= ftisaigportal ftpuserid=FTxxxxxx reset\  
"transfer=(name=xxxxxxxx\  
receive=./path/to/the/file/to/receive.txt)"
```

8 Appendix E: Windows Command Lines for Different Methods of Receiving Data

These following commands are examples only and may require customization at your site.

Example 1: Receive data by a specific sender ID, RECEIVEUSERID= the sender of the data being requested. Depending on options in the TDClient.INI file it will either all be concatenated into one file (if APPEND=Y and AUTOEXT=N) or each item will go into separate files named receive.txt.xxxx, where xxxx can be any number from 0002 to 9999 (if APPEND=N and AUTOEXT=4..)

```
tdclntc network=ftisaigportal ftpuserid=FTxxxxxx reset  
"transfer=(name=xxxxxx receive= receive=.\incoming\receive.txt  
APPEND=N AUTOEXT=  
4)"
```

Example 2: Receive all data by a specific message class, RECEIVECLASS= the message class of the data being requested. Depending on options in the TDClient.INI file it will either all be concatenated into one file (if APPEND=Y and AUTOEXT=N) or each item will go into separate files named receive.txt.xxxx, where xxxx can be any number from 0001 to 9999 (if APPEND=N and AUTOEXT=4..)

```
tdclntc network=ftisaigportal ftpuserid=FTxxxxxx reset\  
"transfer=(name=xxxxxxxx  
receive=.\path\to\the\file\to\receive.txt  
receiveclass=xxxxxxxx)"
```

Example 3: Receive data by unique file name: RECEIVE_SERVER_FILE= the Unique Filename on FTI-TDCM or the Available status record within your Query List.

```
tdclntc network=ftisaigportal ftpuserid=FTxxxxxx reset  
"transfer=(name=xxxxxx  
receive=.\path\to\the\file\to\receive.txt  
\receive server file= xxxxxxxxxxxxxxxxxxxxxxxxx)"
```

Example 4: Receive all data in mailbox: Notice that RECEIVEUSERID= and RECEIVECLASS= have been removed.

```
tdclntc network=ftisaigportal ftpuserid=FTxxxxxx reset  
"transfer=(name=xxxxxx  
receive=.\path\to\the\file\to\receive.txt)"
```

9 Appendix F: Troubleshooting

Listed below are common errors received when sending and receiving data. We are providing you with common resolutions to these errors. Return codes are written to the TDCClient and temp directories in TDCClient. Users can view the SYSOUT file for details of return codes and ftp errors received. Midrange users can view the Temp directory for files with return codes.

Table 12: Errors Received When Sending and Receiving Data

Error	Resolution
RC=03	Can indicate an invalid command line argument. This error is often received because of missing parameters or incorrect syntax in the command line. A missing "SENDUSERID" can generate this error. For instance: NETWORK=FTI-SAIGPORTAL FTPUSERID=FT00000 FTPPASSWD=PASSWORD RESET TRANSFER=(NAME=anything SEND=sendfilename SENDUSERID=FT00000 OTHER COMP PARMS='SECFILE=secfilename')
RC=05	Can indicate a missing network header (O*N05) or trailer (O*N95) in the data file; or the SECFILE may have a syntax error. May indicate an invalid password.
RC=32 Error writing output file WORK01 WARNING: Compression of file failed Error: Compress of file failed Compress failed. Error Code: CMP0033	Resolution is to increase space for WORK01 or receive data by tape.
RC=103	Decompression error; check the SECFILE for syntax errors.
421 Peer Closed Connection	If you receive the error "421 peer closed connection" when receiving files then your internet connection has dropped. Connect again and you should receive as normal. To prevent manually restarting your job in the future, make sure your TDCCLIENT.INI file has the following parameters specified: AUTO_RETRY=Y MAX_RETRY=5 RETRY_DELAY=30

425	Make sure these parameters are in the Tdclient.ini file, and are set to "Y": DATA_OVER_COMMAND=Y
332 or 531 Change password required	Your FTI-SAIG password has expired. You must change your mailbox password every 90 days.
Autoextent function not working properly	When setting options in the tdclient.ini file to make separate files for every file received(AUTOEXT=Y APPEND=N), the files are not created correctly. The 05 header is placed in one file, and the data and 95 trailer is getting placed in a different file. Solution is to set AUTOEXT=N APPEND=Y to associated file and receive multiple files. Designate file name to append all files plus N05.
EDC81281 Connection Refused	In MVS 2.8 LE 1.9+, an LE(Language Environment) error. IT staff need to look at the LE setup.
<u>UNIX/LINUX only</u> ld.so.1: ea2ksunc: fatal: libcpsql.so: open failed: No such file or directory Chngpasswd: 18540 Killed	The solution is to copy the shared objects to the shared library. Then, issue a chmod on the files from the lib directory (chmod -R 755 libcpsql.so). Note: the lib directory can be located from the root (/lib).libcpsql.so, if this object is not located in the shared library then please contact customer service.

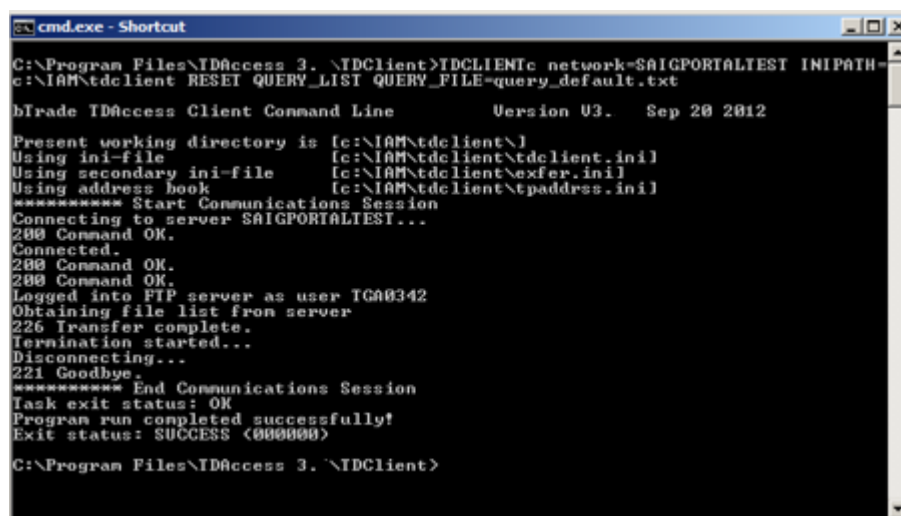
<p>B37 U4083; SOC4; CEE3250C The system or user abend SB37 R=00000004 was issued. From entry point DCCloseOutputFile at compile unit offset.</p>	<p>We have experienced multiple reasons for this error. Solutions may include writing to Tape, removing the RLSE command (with VAN reference), or pre-defining datasets prior to EA step.</p> <p>You can try one of the following solutions:</p> <p>If using the RLSE parameter in your in DD definition, i.e., SPACE=(CYL,(nn,nn),RLSE), you may receive SB37 errors when receiving multiple batches. TDClient opens the Receive file for the first batch to be received and then closes the file releasing unused space. When the next batch is received EA opens the Receive file again with a disposition of MOD, which appends the new batch to the data already in the file. Since the remaining space allocated was released after the first batch was received this may cause a space problem. A similar situation may occur with the additional files required by EA for a Send or a Receive, SYSUT1, SYSUT2, WORK01, WORK02, WORK03, WORK04 and EASTATUS. We recommend that you pre-define these datasets prior to the TDClient step. You can then use a MOD disposition to append multiple batches to the file. See Appendix B, Example 10 for a sample using IEBGENER to pre-define datasets.</p> <p>There is a known issue with EA resulting in an abend code U4083 or SOC4. This is caused by a storage overlay. Click Commerce has repaired the problem in version 1.5 of EA. A temporary solution is to pre- define the files used by TDClient. See solution number 1 above.</p>
--	--

<u>MVS v2.6 Only</u> Error Code: DCM0061 WARNING: Decompression of file failed Error: Decompression of file failed Decomp failed. Error Code: DCM0061 WARNING: Decompression of file failed Made backup copy of compressed file which failed to decompress. Backup FileName: WORK01.001	Add DSN=&&WORK01(etc) to the WORK01 DD, WORKxx, and SYSUTx DDs in the RECV job.
<u>UNIX/LINUX only</u> Decomp failed,Error Code: DCM0073	Your directory does not exist. Ensure that the AUTOEXT parameter in the TDCLIENT.INI file is set to N.(AUTOEXT=N) and the APPEND is set to Y (APPEND=Y). In your Transfer statement, make sure the RECEIVE= parm has a directory name, not a filename specified (receive=./incoming/sarfiles/).
WARNING: Action failed Unable to RESET TDClient Restart file Failed to open EA Restart file [./eastatus.txt] in write mode permission denied\	Solution is to create the eastatus.txt in the TDCLIENT.INI directory, with read & write permissions.
<u>Windows only</u> Executes prior transmission when sending or receiving a new file.	Slashes (/) used in the command line are backwards and should be facing the opposite direction (\). The system will not execute the line and submits the last transmission that was stored in the TDClient.ini file. Check the Response.log file, located in the directory where the TDClient has been installed, for errors.

10 Appendix G: Configuration Changes Needed for Connecting to SAIG System

There are two methods to connect to the existing SAIG system using the upgraded TDClient version. Users may use an alternate tdclient.ini configuration file located in a different directory or update the FTI-SAIG tdclient.ini to add the SAIG connection information.

1. The command to use this alternate configuration file is as follows and displayed in Figure 44:
 - a. `tdclientc network=SAIGPORTALTEST INIPATH=[Alternate Directory] RESET QUERY_LIST QUERY_FILE=query_default.txt`
 - b. In order to use the alternate TDCLIENT.INI configuration file, users need to copy the following files and directories to the alternate directory:
 - i. Files: `tdclient.ini`, `exfer.ini` and `tpaddrss.ini`
 - ii. Directories: `incoming`, `runtime`, `maint`, `outgoing`, `error`, `security`, and `temp`



```
cmd.exe - Shortcut
C:\Program Files\TDAccess 3. \TDClient>TDCLIENTc network=SAIGPORTALTEST INIPATH=
c:\IAM\tdclient RESET QUERY_LIST QUERY_FILE=query_default.txt

bTrade TDAccess Client Command Line          Version V3.   Sep 20 2012

Present working directory is [c:\IAM\tdclient\]
Using ini-file          [c:\IAM\tdclient\tdclient.ini]
Using secondary ini-file [c:\IAM\tdclient\exfer.ini]
Using address book      [c:\IAM\tdclient\tpaddrss.ini]
***** Start Communications Session
Connecting to server SAIGPORTALTEST...
280 Command OK.
Connected.
280 Command OK.
280 Command OK.
Logged into FTP server as user TGA0342
Obtaining file list from server
226 Transfer complete.
Termination started...
Disconnecting...
221 Goodbye.
***** End Communications Session
Task exit status: OK
Program run completed successfully!
Exit status: SUCCESS (000000)

C:\Program Files\TDAccess 3. \TDClient>
```

Figure 19: TDClient alternate configuration

```

C:\>tree "TDAccess_3_3_1"
Folder PATH listing for volume Windows8_OS
Volume serial number is 5245-29E4
C:\TDAccess_3_3_1
├── TDClient
│   ├── Error
│   ├── Incoming
│   ├── Maint
│   │   └── trans
│   ├── mdn
│   ├── Outgoing
│   ├── Runtime
│   │   └── IGN
│   ├── Security
│   └── Temp
└── TDCServer

```

Figure 20: TDClient alternate configuration displaying directory

2. Update the FTI provided tdclient.ini file with the SAIG tdclient.ini file information outlined below.

The [NETWORKS] section will need the two networks from the SAIG tdclient.ini added.

```

[NETWORKS]
1=FTISAIGPORTAL
2=SAIGPORTAL
3=SAIGPORTALTEST

```

Copy the following complete sections into the FTI tdclient.ini under the [FTISAIGPORTAL-DEFAULT_RECEIVEPARMS] section.

```

[SAIGPORTAL]
HOSTIPNAME=SAIGMAILBOX.ED.GOV
...
SMTP_PASSWORD=

[SAIGPORTAL-DEFAULT_SENDPARMS]
COMPRESS=Y
...
CRLF=Y

[SAIGPORTAL-DEFAULT_RECEIVEPARMS]
APPEND=Y
...
ASCII=Y

```

```
[SAIGPORTALTEST]
HOSTIPNAME=SAIGMAILBOXTEST.ED.GOV
...
SMTP_PASSWORD=

[SAIGPORTALTEST-DEFAULT_SENDPARMS]
COMPRESS=Y
...
CRLF=Y

[SAIGPORTALTEST-DEFAULT_RECEIVEPARMS]
APPEND=Y
...
ASCII=Y
```

11 Appendix H: tdclient.ini File Defaults

```
[EAPATH]
BASEPATH=./FT00000\
```

```
[SECURITY]
PARTICIPANT=A.T. STILL UNIV. OF XX FT00000
EDINAME=FT00000
TELEPHONE=000-000-000-EXT
COMMONNAME=A.T. STILL UNIV. OF XX FT00000
COUNTRY=US
ORGANIZATION=SAIG
TITLE=
ORGUNIT=Name      Lastname
ADDRESS1=000 Street Name
ADDRESS2=
LOCALITY=City
STATE=ST
POSTALCODE=00000-
EMAILADDRESS=FirstnameLastname@XX.EDU
RTMGENERATE=N
AUTOUPDATERUNTIME=N
MODULUS=2048
APPROVALCODE=D3CDB1BAD9C4158B8D1871C1FD6B696BB878F304F2563A07
BC6BE8C508A5E95
EXPDATE=18771BADCBFFC177A699260DC297F74C5979DA7235E783F8FDAC4545
738802DB
VALID=FULL VERSION
```

NETWORK=
CERTDEST=
RTMCLASS=

[NETWORKS]
1=FILESYSTEM
2=FTISAIGPORTAL

[IDENTIFY]
NETWORK=FTISAIGPORTAL
MULTITHREADED=N
DISABLE_DIALER=Y
MULTIFILE=N
AUTO_RETRY=N
MAX_RETRY=3
RETRY_DELAY=15
AUDIT_START_DATE=
AUDIT_END_DATE=
HELPPFILENAME=
STARTTIME=
STARTDATE=

[EDIINT_SETUP]
CERT_IMPORT_DIR=<home>/tdserver/certs
AS1_MDN_DISP_TO=
AS2_MDN_DISP_TO=
MDN_DIR=<home>/tdserver/mdns
DIVERT_DUP_DATA=N
SECURITY_LEVEL=1

[FIREWALL_SETUP]
FIREWALL_TYPE=0
FIREWALL_HOSTIPNAME=
FIREWALL_USERID=
FIREWALL_PASSWD=

FIREWALL_PORT=0

[FILESYSTEM]

HOSTIPNAME=/prgadmin/
HOSTIPNAME2=
NETWORKSTYLE=FILESYSTEM
DIAL_ENTRY=
CASE=L
PASSIVE=N
SSL=N
AUTO_DIAL=N
AUTO_DISCONNECT=N
SECURITYMENU=N
SUNIQUE=0
CONTROL_PORT=
MAX_AUTO_DIAL_DELAY=180

[FTISAIGPORTAL]

HOSTIPNAME=ftisaigmailbox.ed.gov
HOSTIPNAME2=
NETWORKSTYLE=EAFTP
DIAL_ENTRY=
CASE=L
PASSIVE=N
SSL=Y
AUTO_DIAL=N
AUTO_DISCONNECT=N
SECURITYMENU=N
SUNIQUE=0
CONTROL_PORT=26586
MAX_AUTO_DIAL_DELAY=180
USERID=FT00000
PASSWORD=DFEF041B49DBA893972A7FA35C4FA0349C2F84DA01C8A9C61C08B
C869515E06D

[FTISAIGPORTAL-DEFAULT_SENDPARMS]

FILTER=Y
CRLF=Y

ASCII=Y
COMPRESS=Y

[FTISAIGPORTAL-DEFAULT_RECEIVEPARMS]
AUTOEXT=N
ASCII=Y
APPEND=Y
UNCOMP=Y