Student Aid Internet Gateway

Host Communication Guide for Mainframe and Midrange Users

Version 1.2

U.S. Department of Education

282 H

Table of Contents

Section 1: Overview

Preface

Welcome to the U.S. Department of Education’s Student Financial Assistance (SFA) Student Aid Internet Gateway (SAIG) which offers Title IV-eligible post-secondary institutions, third-party servicers, state agencies, lenders and guarantors a secure, Internet-based method of exchanging Title IV data with the SFA Application Systems. The SAIG replaces what was formerly known as "TIV WAN" by moving Title IV transmissions from the General Electric value-added network to the Internet.

This guide is designed to meet the reference needs of programmers and data processing staff who transmit Title IV Data via a 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.

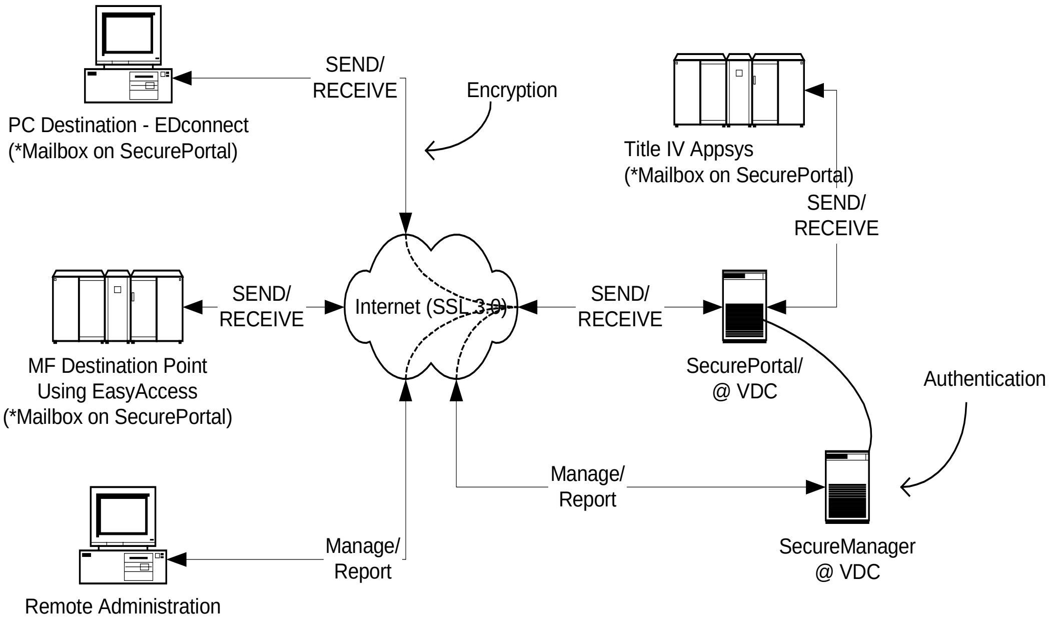
The SAIG is designed around SFA's vision and target architecture to provide an Internet solution for data transmissions. SFA to the Internet offers an integrated solution for SFA’s constituents by implementing a Commercial Off-the-Shelf (COTS) application that supports multiple hardware and operating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

Note: To all third party software providers:

1. Do not include transmission headers and trailers (O*N) on files to be transmitted via EDconnect.
2. Use the appropriate Technical Reference when creating output (low values and nulls should not be in any data transmitted over SAIG).
3. Provide a Carriage Return/Line Feed (CR/LF) in the final position of the data file to be transmitted.

The diagram in Figure 1-1 represents the flow of data between Title IV destination points and Application Systems.

Figure 1-1: System Diagram



* Title IV destination points and Title IV application systems using EasyAccess software component. EasyAccess incorporates the security features of SecureManager and provides data security functions (authentication, encryption) to SEND/RECEIVE data to SecurePortal. PC destination points continue to use EDconnect software (EDconnect will integrate EasyAccess API).

Migration Note: SFA Application Systems will continue to receive data from their current GEIS Open*Net mailbox during the migration period. They will also receive data from their SecurePortal (Internet) mailbox but will send all data from SecurePortal.

The integrated solution consists of EasyAccess, SecureManager, and SecurePortal components. The following sections provide more detail on each of these products.

EasyAccess

EasyAccess is the client software used to send and receive FTP (File Transfer Protocol) Title IV data transmissions securely over the Internet using SSL 3.0 and the Diffie-Hellman Dynamic Key Exchange algorithm. Port 26581 will need to be opened in your firewall to allow inbound and outbound TCP/IP traffic. The EasyAccess client software is available on the U.S. Department of Education's Student Financial Assistance Download (SFAdownload) Web site located at http://SFAdownload.ed.gov.

The EasyAccess client software has the compression and decompression steps built into it. This means you no longer need the separate steps in your MVS JCL or Unix scripts for sends and receives.

EasyAccess is supported under the following mainframe/mid-range operating systems:

- OS/390 MVS/ESA 2.6+ (with LE/370 1.9)
- OS/400 4.2+ (Compiler Level 3.7)
- AIX 4.2+
- Digital UNIX 4.0 (DEC UNIX)
- DEC Open VMS 7.1
- HPUX 10.01+
- SCO UNIX 4.3+
- Sun Solaris 2.6+

SecureManager

The SecureManager (current version is 3.4.4d) is the bTrade.com product that will be used to manage Title IV destination points. The product will run as a thin client and will be accessed via the Web. Users of the system will be: System administrators, customer service staff, and Title IV destination points. Destination points can use this system to manage their mailbox and view network traffic via the Web (referred to as Online SecureManger, OSM).

The OSM User's Guide, containing instructions on how to query your SAIG mailbox, is available on the SFAdownload Web site (http://SFAdownload.ed.gov).

Policy based trading/security enforcement options include:

1. Secure FTP (SSL 3.0, Retry, Restart, Passthru)
2. AS1 (SMTP, S/MIME, MDNs)
3. GISB (HTTP, PGP)
4. AS2 (HTTP, S/MIME, MDNs)
5. MQ Series (SSL 3.0, X12.58)
6. Data/Message/Transaction Digital Signature, Authentication and Encryption Security (X12.58-secret key, X12.58, and EDIFACT public key cryptography)
7. MD5, SHA1, DES, TripleDES, RSA, NVB, and several other structures and algorithms

SecurePortal

SecurePortal open architecture gateway supports the design and implementation of mission critical applications and offers:

1. Interconnect services to bridge the gap between Internet and legacy VAN (Value-Added Network) services
2. Integrated security services
3. Portal Archive, Audit and Event Tracking Stamp Services
4. Virtual Directories to protect system resources (Oracle 8i)
5. In-stream audit and archiving for easy portal traffic review and recovery
6. Support for all popular trading protocols and standards for maximum flexibility
7. Event management and broadcasting for in-stream analysis and external process triggers

Section 2: Installation Guide

Configuring and Installing EasyAccess

Student Aid Internet Gateway (SAIG) destination points and application systems may use EasyAccess in any of the following environments:

- MVS OS/390
- AS/400
- HP-UNIX
- AIX
- Sun Solaris
- Open VMS

EasyAccess requires a physical connection to the Internet.

This product does not provide phone dialing or other functionality to establish the physical connection.

The EasyAccess Client software and the accompanying documentation is available on the SFAdownload site (http://SFAdownload.ed.gov).

Installation on MVS OS/390 Systems

To use EasyAccess, you must have MVS OS/390 2.6 or above, with the MVS feature of Language Environment Version 1, Release 9. Higher versions of MVS and OS/390 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.

To install EasyAccess:

Step 1) Create a unique directory on your PC or LAN drive that will serve as the destination for the downloaded files.

Step 2) Download the MVS OS/390 set of files containing the **EAMVS146.EXE** and **EASYACC.INI** files from the SFAdownload Web site (http://SFAdownload.ed.gov).

Step 3) Double-click the self-extracting **EAMVS146.EXE** file. This step will extract the following files:

a) Xmit.bin

b) Readme.txt

c) Decomp.log

Step 4) FTP the XMIT.BIN file in BINARY mode to an MVS dataset with the following attributes: RECFM=FB, LRECL=80, and BLKSIZE=3120. The xmit.bin file contains the EasyAccess 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 2-1), ftp client software, or 3270 emulator.

Installation on MVS OS/390 Systems (Continued)

Figure 2-1: Example FTP from a DOS FTP Prompt

```
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:\xmit.bin ‘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
```

Step 5) 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 2-2.

c. When prompted to “enter restore parameters”, type **DA ('your.install.dataset')**. See Figure 2-2. 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 above steps will create an Installation Library containing the files required to complete the installation of EasyAccess.

Figure 2-2: Example TSO RECEIVE

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

Installation on MVS OS/390 Systems (Continued)

Step 6) a. Edit the $INSTALL member of the **your.install.dataset** and make the changes described in Steps 1 - 6 of Figure 2-3 below.

b. Run the $INSTALL JCL.

Figure 2-3: Example MVS OS/390 $Install File after Editing

```
//jobname  JOB (acct),pgmr,MSGLEVEL=1,REGION=7M,CLASS=A,
//  MSGCLASS=X,NOTIFY=user
//*
//* MEMBER $INSTALL
//*
//* EasyAccess/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.eaload to a valid
//*         destination dataset name.
//*  4) Change all occurrences of your.user.easamp to a valid
//*         destination dataset name.
//*  5) Change all occurrences of your.user.cpdbrm 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(EALOAD),DISP=SHR
//EASAMP   DD DSN=your.install.dataset(EASAMP),DISP=SHR
//CPDBRM   DD DSN=your.install.dataset(CPDBRM),DISP=SHR
//CPSAMP   DD DSN=your.install.dataset(CPSAMP),DISP=SHR
//SYSTSIN  DD  *
RECEIVE INFILE(EALOAD)
       DATASET('your.user.eaload')
RECEIVE INFILE(EASAMP)
       DATASET('your.user.easamp')
RECEIVE INFILE(CPDBRM)
       DATASET('your.user.cpdbrm')
RECEIVE INFILE(CPSAMP)
       DATASET('your.user.cpsamp')
/*
```

Installation on MVS OS/390 Systems (Continued)

Step 7) Allocate a new file with attributes LRECL=80, RECFM=FB, BLKSIZE=23440 and name the file **`'your.dataset.prefix.EASYACC.EXFER.INI'.`** This file will remain blank until you send your first file using EasyAccess, at which time it will populate with parameters contained in the TRANSFER command line of your JCL (described in Section 4, Communication Procedures).

Step 8) Upload the **EASYACC.INI** file (from Step 1) as BINARY with the attributes RECFM=FB, LRECL=80, CRLF, and name the file **`'your.dataset.prefix.EASYACC.INI'.`** This file holds network configuration information and is described in Section 4, Communication Procedures. *Do not alter this file.*

Installation on HP, Sun & AIX UNIX Systems

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

To install EasyAccess:

1. Make a directory called **easyaccess** on the UNIX box you are using by typing **mkdir easyaccess** from the command prompt.
2. Type **CD easyaccess** from the command prompt and press **Enter.**
3. Make directories within the easyaccess directory by typing the following commands:
   a) **Mkdir outgoing** and then press **Enter.**

   b) **Mkdir temp** and then press **Enter.**

   c) **Mkdir runtime** and then press **Enter.**

   d) **Mkdir incoming** and then press **Enter.**
4. Download **ea2khp10.sfx (HP)** or **ea2ksun.sfx (Sun)** or **ea2kaix.sfx (AIX)** and the **easyacc.ini** from the U.S. Department of Education's SFA Software Download site (http://SFAdownload.ed.gov) into the **easyaccess** directory.
5. Ensure the EasyAccess install file has execution, read, and write permissions.
6. Run the EasyAccess self-extracting file from the command line to expand its components by typing **ea2khp10.sfx** or **ea2ksun.sfx** or **ea2kaix.sfx** and then press **Enter**.

   Note: You will see the file decompressing at this time.
7. After the file has decompressed, the EasyAccess installation is complete.

Installation on DEC Open VMS System

EasyAccess for VMS is provided in a self-extracting file that must be extracted from a VMS session. The file must be transferred to a VMS account using some sort of file transfer application such as FTP. Once the file is transferred, log in to the VMS system and execute the file **ea2kvms71.exe**. *(Note: Make sure this file is in the directory in which the application is to be installed.)* See Figure 2-4:

Figure 2-4

```
$ run ea2kvms71.exe
This will create several files:
$ run ea2kvms71.exe
COMM-PRESS 2000 Ver 4.4.2 *o* (010) (master, triple DES) (c) Copyright 1990-2000
DECOMPRESS STARTED - Fri Aug 31 01:49:11 2001
Decomping apha1$dka0:[sys0.syscommon.][user.bjackson.ea2k]ea2kvms71.exe;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]CMDPARSEPFX.EXE;
1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]COMPRESS.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]DECOMP.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]EA2KCMD.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]GENKEYS.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]IEBASE.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]IMPORT.EXE;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]LICENSE.TXT;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]PFXHELP.TXT;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]README.TXT;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]RELEASE.TXT;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]README.INV;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]EASYACC.CMD;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]EXFER.INI;1
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]TPADDRSS.INI;1
```

Figure 2-4 (Continued)

```
Decomped
APHA1$DKA0:[SYS0.SYSCOMMON.][USER.BJACKSON.EA2K]INSTALL.COM;1
```

Execute install.com at the command prompt:

$ @install

This will create the directory structure for EasyAccess for VMS.

(Install.com)

```
$! Simple command-procedure to create required sub-directories
$! for EasyAccess 2000 Client program for OpenVMS
$ create/dir [.incoming]
$ create/dir [.outgoing]
$ create/dir [.maint]
$ create/dir [.temp]
$ create/dir [.security]
$ create/dir [.runtime]
$ create/dir [.runtime.ign]
```

Installation on AS/400 System

To use EasyAccess, you must have OS/400 v4.2 or above. You must also have installed and configured the TCP component of AS/400 and establish a physical connection to the internet.

Installing EasyAccess AS/400 Software:

1. Download **ea400148.exe** and the **easyacc.ini** files from the U.S. Department of Education's SFADownload site **(http://SFAdownload.ed.gov).**
2. Create an empty save file on the AS/400.

   An example command to type would be - **CRTSAVF SAVEFILE.**
3. On a Windows 95/98/2000/NT system, decompress the distributed files by running the self-extracting executable file (EA400148.exe). This will generate the following files in the current directory:

   EA400.BIN (EA code for AS/400)

   README.TXT (brief notes on install)

   decomp.log (log on de-compressing the above 2 files)
4. Upload the distributed SAVE file (EA400.BIN) from the Windows PC to the new AS/400 SAVE file using a binary mode FTP transfer.

See Figure 2-5 on the next page.

Figure 2-5: Example FTP Session to Transfer EasyAccess2000 Software to AS/400

```
> ftp x.x.x.x (x.x.x.x = IP Add)           connect to AS/400 at network
address

220 User (as/400:(none)): userid           type userID
necessary

331 Enter password.                  type password needed

230 USERID logged on.

ftp> bin                                   switch to binary
transfer mode

220 Representation type is binary IMAGE.    confirmation

ftp> put EA400.BIN SAVEFILE          transfer installation save
file

200 PORT subcommand request successful.

150 Sending file to member SAVEFILE in file SAVEFILE.

250 File transfer completed successfully.

ftp> quit                                  end FTP session
```

1. Use the RSTLIB command to unload the EasyAccess library. An example command is: **RSTLIB SAVLIB(EA148LIB) DEV(*SAVF) SAVF(SAVEFILE) RSTLIB(EA2KLIB)**
2. After the restore is complete, create the EASYACC object: An example command is: **CRTPF FILE(EA2KLIB/EASYACC) RCDLEN(200) FILETYPE(*SRC)**
3. FTP the easyacc.ini over to the OS400 in ASCII mode. An example command is: **Put easyacc.ini ea2klib/easyacc rep**
4. Make EasyAccess programs available to users. An individual user can do this by running "ADDLIBLE newlib" per session or the system administrator can make this available to all user by adding the newlib to the LIBRARY LIST.

Exchanging Data using AS/400 EasyAccess

1. Use the AS/400 command line or use the EasyAccess command-line interface application with the name EA2K400C.
2. Use the PARM keyword on the CALL statement to specify this information:
   - Name of a stored transfer, User ID and password required to logon.
   - Compression and decompression program options, or Name of the command file that contains this information.

   NOTE: When using the command file option from the command-line interface, the command file must be in a physical file format and should contain the appropriate transfer, compression, and decompression parameters.

   The next table displays several command syntax examples used to implement an EasyAccess data transfers on an AS/400 computer.

AS/400 Data Transfers - Command Syntax Examples

Example	Command Examples
Command file name	**CALL EA2K400C PARM('TRANSFER=CMDFILE=LIBRARY/FILE')**
Use a stored transfer called from command-line interface.	**CALL EA2K400C PARM('TRANSFER=TRANSFER')**
Use a stored transfer with a blank in its name.	**CALL EA2K400C PARM('TRANSFER="TEST TRANSFER"')** Use double quotes to surround the stored transfer name.
Creating a stored transfer for later use.	**CALL EA2K400C PARM('TRANSFER=NAME= "NEW TRANSFER" 'PASSWD=password' 'NETWORK=network' 'ASCII' 'CRLF' 'COMPRESS')**

NOTE: The new transfer name and parameters are appended to the EXFER file and can be called later by using only the stored transfer name. Specifying the transfer and compression options later will not be necessary.

AS/400 Operating System-specific EasyAccess Considerations

To simplify your command syntax when running utility programs or EasyAccess data transfers, you may want to use the CHGCURLIB (change current library) command to make the User specific RUNTIME library the current library.

Temporary Work Files

EasyAccess creates several temporary files as part of its normal application processing. These files are written to the AS/4000 designated "current library". This is another reason for making the User specific RUNTIME library the current library prior to EasyAccess execution (with the CHGCURLIB command).

During transmission, EasyAccess creates temporary files named **SYSUT1** and **SYSUT2** in the current directory. These files hold directory listings and copies of compressed data files. System defaults are usually adequate for creating these temporary files; however, if you send or receive large files you may need to pre-create one or both of the temporary files with an adequate size to hold the data. If this is the case, then create these files as physical files with a record length of 256 bytes. You may need to experiment with the number and size of the record extents to allocate files of the desired sizes.

Naming and Allocating Work Files

Specify the file names used to send and receive data by using the LIBRARY/FILENAME(MEMBER) syntax. If the file is available via the **LIBLIST** command, then you can omit the LIBRARY portion of the command. If the first (or only) library member is needed, then you can omit the (MEMBER) portion of the command.

When receiving data, EasyAccess creates the output files if they do not exist. However, the files are created in the current library with default values for maximum record length and file size. If the defaults are not acceptable, then you should create the files with the appropriate with number and size of the record extents, prior to receiving the transmitted data.

Section 3: Security

Password Update Procedure

General Information

At initial setup, the default password to your Student Aid Internet Gateway (SAIG) mailbox User ID is **PASSWORD**. You will be required to update your password before performing any other activity. (See Figure 3-1.)

- Passwords must be 7 to 8 characters long.
- Passwords can contain alpha and numeric characters, but must begin with an alpha character.
- Passwords submitted via EasyAccess are forced to all upper case, unless you over-ride the default in your Transfer command line.
- Passwords expire every 120 days, but you may change your password as often as you want.
- You must update the password on all batch jobs used to communicate with the SAIG.
- If you have any difficulty establishing your first password, contact CPS/WAN Technical Support at 1-800-330-5947 and request to have your SAIG mailbox password reset.

Note: Network passwords can be changed by either of two methods: via batch job with the EasyAccess software, or with the On-Line SecureManager (OSM). This document addresses only the batch process (this is the preferred method). The OSM User Manual, containing instructions on the alternate method of updating your password is available on the SFAdownload site.

Batch Procedure

The example below (Figure 3-1) shows how to change your mailbox password. You must manually change all other JCLs or Scripts to reflect this new password, and you must also use this new password when signing onto OSM. A password change cannot be submitted by itself, but must be accompanied by some other network activity: sending data, receiving data, or a query list. See figure 3-1.

The only data elements required as input to the batch password update procedure are:

• Old password:	7 to 8 characters
• Backslash separator:	(Always a backslash): \
• New password:	7 to 8 characters
• Backslash separator	(Always a backslash): \
• Verify password:	7 to 8 characters

Note: No spaces between the old password, backslash, and the new password.

Figure 3-1: Example JCL Command Line for Password Change

```
//CMDSEND   DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx
FTPPASSWD=oldpass\newpass\newpass RESET
QUERY_LIST QUERY_FILE=DD:QUERY
//*
//QUERY   DD DSN=your.dataset.name.receiving.query.list
//                 DISP=(NEW,CATLG),UNIT=SYSDA,
//                 DCB=(LRECL=700,BLKSIZE=7000,RECFM=FB),
//                 SPACE=(CYL,(30,13))
//*
```

Figure 3-2: Example UNIX Script for Password Change

Unix command lines are NOT case sensitive. Backslashes are being used at the end of each line for line continuation. Double quotes or no quotes may be used in command lines that use the backslashes for line continuation. Do not use single quotes.

```
ea2khp10c "network=saigportal" ftpuserid=TGxxxxx         \
ftppasswd=OLDPASSWD/NEWPASSWD/NEWPASSWD reset query_list
```

Figure 3-3: Example AS400 Command Line for Password Change

Chgpasswd CMDFILE

```
crtpf file(ea148lib/passchg) rcdlen(80) filetype(*src)

*************** Beginning of data **********
NETWORK=SAIGPORTAL
FTPUSERID=TGxxxxx
FTPPASSWD=OLDPASSWORD/NEWPASSWORD/NEWPASSWORD RESET
QUERY_LIST QUERY_FILE=DD:QUERY
****************** End of data *************
```

Section 4: Communication Procedures

Introduction

This section describes basic procedures for sending and receiving data over the Student Aid Internet Gateway (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

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 4-1.

Figure 4-1: Example Network and Transfer Command Lines for Sending a Single File

```
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password  RESET
TRANSFER=(NAME=yourname SENDUSERID=TGxxxxx  SEND=DD:inputfilename
         OTHER_COMP_PARMS='SECFILE=DD:SECFILEX')
```

Figure 4-2: Network Command Line Keywords

Figure 4-2 lists keywords used in the NETWORK command line to sign on to SecurePortal and perform any network activity: sending, receiving, query list or password changes.

Keyword	Definition
NETWORK	This parameter will define the secure network being used for the communications session. This will always be SAIGPORTAL to match the network defined in your EASYACC.INI file.
FTPUSERID	This is your mailbox user ID (TG number). This will not change from what you currently use.
FTPPASSWD	This is the password associated with the FTPUSERID. Your initial password is set to PASSWORD (all caps). The EASYACC.INI file has the default set to CASE=U (to convert and send data to the server in upper-case format.) You can override the default in your Transfer command line by using CASE=L (to send data to the server in lower case) or CASE= blank (to send case sensitive data to the server).

Figure 4-3: Transfer Command Line Keywords for Sending Data

Keyword	Definition	
TRANSFER	This defines the transfer parameters of data being sent. A TRANSFER=NAME= can be saved to the EASYACC.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. TRANSFER contains the following keywords (parameters):	
	NAME=	This names the transfer being created. The definition will be saved in the EASYACC.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 TG ID as a place holder in this field. Using your own TG ID is highly recommended. Note: Using a TG ID other than your own will cause your job to fail if the TG ID is deleted or inactivated. The O*N05 header record in your data file is the default for the TG 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: 1) The SECFILE builds the network headers and trailers used by SecurePortal to separate files and place files in the correct mailboxes; and 2) It forces EasyAccess to use the SECFILE parameters to send and receive data properly. See Figure 4-4.
	SAVE (Optional)	A TRANSFER NAME can be saved to the EASYACC.EXFER.INI file by using this keyword at the end of your command line command. This is helpful when data of the same message class is sent or received on a routine basis. Example: *TRANSFER=(NAME=yourname RECEIVE=DD:outputfilename* *RECEIVEUSERID=TGxxxxx RECEIVECLASS=messageclass) SAVE* 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.

Note: The EASYACC.INI file has the default set to CASE=U (to send the password in upper-case format.) You can override the default in your Transfer command line by using CASE=L (to overwrite the password in lower case format) or CASE= blank (to overwrite the password in case sensitive format). This affects the password field only.

Figure 4-4: Example SECFILE

```
SENDER(TGxxxxx);
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);
```

***Note:** Pay close attention to the placement of semicolons and spaces or errors may result when you submit your JCL. The SECFILE is required when sending data.

- SENDER(TGxxxxx) parameter is required by EasyAccess but is not being read, so you may insert any literal in the parenthesis. We recommend using your TG# in this parameter.
- 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 EasyAccess 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.
  - The O*N01 File Header and the O*N99 File Trailer are the first and last records, respectively, on every data file being sent.

Figure 4-5: Example MVS JCL to SEND Data

```
//STEP0020  EXEC PGM=EA2KMVSC,REGION=4M,TIME=1440,
//          PARM='CMDFILE=DD:CMDSEND'
//STEPLIB   DD DSN=your.dataset.prefix.EALOAD,DISP=SHR
//*
//EASYACC   DD DSN=your.dataset.prefix.EASYACC.INI,DISP=SHR
//*
//EXFER     DD DSN=your.dataset.prefix.EASYACC.EXFER.INI,DISP=SHR
//*
//CMDSEND   DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=xxxxxxxx RESET
TRANSFER=(NAME=yourname SENDUSERID=TGxxxxx 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(TGxxxxx);
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 and TG numbers.

Figure 4-6: Example UNIX Script to SEND Data

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx reset \
"transfer=(name=yourname                                                \
   senduserid=TGxxxxx                                                   \
   send=/your/send/file.txt                                             \
 other_comp_parms=secfile=./path/to/your/secfile.txt)”
```

Note: Replace the ea2khp10c with the appropriate client name. Backslashes are being used at the end of each line for line continuation. Double quotes or no quotes may be used in command lines that use the backslashes for line continuation. Do not use single quotes. You can have more that one “transfer=(DATA)” line to transmit multiple files.

See Figure 4-4 for example of SECFILE.

Figure 4-7: Example AS/400 Commands to SEND Data

```
crtpf file(ea148lib/ncssend) rcdlen(80) filetype(*src)
```

```
*************** Beginning
NETWORK=SAIGPORTAL
FTPUSERID=TGxxxxx
FTPPASSWD=PASSWORD RESET
TRANSFER=(NAME=yourname
SEND=EA148LIB/FILE
SENDCLASS=CLASS
OTHER_COMP_PARMS=’SECFILE=EA148LIB/SECFILEX’)
****************** End o
```

1) Create a physical file for the command parameters and a physical file for the secfile.

```
CRTPF FILE(EA2KLIB/CMDFILE) RCDLEN(80) FILETYPE(*SRC)
CRTPF FILE(EA2KLIB/SECFILE) RCDLEN(80) FILETYPE(*SRC)
```

2) Writer the transfer parameters to the command file and the sender receiver characteristics for the secfile. See Figure 4-4 for example of SECFILE.

Figure 4-8: Example Open VMS Commands to SEND Data

```
ea2kcmd “network=SAIGPORTAL” ftpuserid=TGxxxxx ftppasswd=PASSWORD reset
“transfer=(name=’yourname’ send=/your/send/file.txt senduserid=TGxxxxx)”
```

Figure 4-9: Example of Send File with Network Headers

```
O*N01                                                73
O*N05TGxxxxx       ,CLS=messclas,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,
CPS HEADER  0450HTG51234  20000718150719            0001#C10025002000071
10024720300'01002SAM                         008002472031
CPS TRAILER 0450H       20000718150719              000100010#C100250020
O*N95TGxxxxx       ,CLS=messclas,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,
O*N99                                                73
```

See Figure 4-3 for parameters SENDUSERID and SENDCLASS in the Command Line Input section. **Note: All files must be fixed block (FB) and padded with spaces to full record length.**

Following are lists of Input, Temporary and Output logs for send and receive JCL or scripts.

Input DD or Filename	Definition
STEPLIB	Dataset name (your.dataset.prefix.EALOAD) containing the EasyAccess program libraries you installed.
EASYACC	Dataset name containing the EASYACC.INI file, which contains network access information. - *Do NOT alter this file*.
EXFER	Dataset name containing the EASYACC.EXFER.INI file, which stores your saved TRANSFER commands.
CMDSEND*	**Command and Transfer statements to Send data.**
CMDRECV**	**Command and Transfer statement to Receive data.**
SENDFILE*	Location of the input file you wish to send from your SAIG mailbox.
RECVFILE**	Location of the pre-allocated files that will receive data pulled from your SAIG mailbox.

*** Used only on Send Transmissions**
**** Used only on Receive Transmissions**

Temporary DD or Filename	Definition
SYSUT1***	
WORK01***	Receives data and decompresses into the Receive file
WORK02***	Contains list from query.
WORK03***	
WORK04***	Works in conjunction with Eastatus
EASTATUS	Contains any errors during a send or receive session. Must be an allocated file; cannot be a temp file.
DCMPLOG**	Logs decompression step for each file received. Indicates if any files failed decompression.

Output DD or Filename	Definition
SECFILEX	This name comes from the "SECFILE=DD:" in the TRANSFER statement. It stores the SECFILE command lines that tell EasyAccess how to format the headers and trailers during compression/decompression.
OUTMSG	Confirms successful: a) login, compression, and send of file, or b) decompression and receive 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 the commands and the 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 EasyAccess Client software. They may be defined as temporary files with the following parameters: LRECL= 8192, RECFM=VB, BLKSIZE=0.**

Figure 4-10: Transfer Command Line Keywords for Receiving Data

Keyword	Definition	
TRANSFER	This defines the transfer parameters of data being received. A TRANSFER=NAME= can be saved to the EASYACC.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. Within this keyword are the following keywords (parameters).	
	NAME=	This 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=	This is the 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 wish 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 _PARMS= (Optional)	These are decompression parameters used only during the decompression step for receiving data.
	SECFILE= (Optional)	This is a parameter used in OTHER_DECOMP_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: 1). The SECFILE builds the network headers and trailers used by SecurePortal to separate files and place files in the correct mailboxes; and 2). It forces EasyAccess to use the SECFILE parameters to send and receive data properly. See Figure 4-4.
	SAVE (Optional)	A TRANSFER NAME can be saved to the EASYACC.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=DD:outputfilenameRECEIVEUSERID=TGxxxxx RECEIVECLASS=messageclass) SAVE*

Note: The EASYACC.INI file has the default set to CASE=U (to send the password in upper-case format.) You can override the default in your Transfer command line by using CASE=L (to overwrite the password in lower case format) or CASE= blank (to overwrite the password in case sensitive format). This affects the password field only.

Figure 4-11: Example Network and Transfer Command Lines for Receiving a Specific Message Class from a Specific Sender (the RECEIVEUSERID):

```
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password  RESET
TRANSFER=(NAME=yourname  RECEIVE=DD:outputfilename
          RECEIVEUSERID=TGxxxxx RECEIVECLASS=messageclass)
```

You can modify the commands in many different ways, depending on what data you wish to receive. To receive all files of a given message class, specify that message class (EAPS02OP, 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. Appendix B, Example 7, gives an example of how to receive a batch and concatenate the optional O*N01 header and O*N99 trailer. Remember, when receiving files from your mailbox, files are received in the order of the query list option.

Figure 4-12: Example JCL to Receive Data

```
//STEP0020  EXEC PGM=EA2KMVSC,REGION=4M,PARM='CMDFILE=DD:CMDRECV'
//*
//STEPLIB   DD DSN=your.dataset.prefix.EALOAD,DISP=SHR
//*
//EASYACC   DD DSN=your.dataset.prefix.EASYACC.INI,DISP=SHR
//*
//EXFER     DD DSN=your.dataset.prefix.EASYACC.EXFER.INI,DISP=SHR
//*
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=xxxxxxxx RESET
TRANSFER=(NAME=xxxxxxxx RECEIVE=DD:RECVFLE
   RECEIVEUSERID=TGxxxxx RECEIVECLASS=messageclass)
//*
//RECVFLE   DD DSN=your.dataset.receive.file,
//             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=*
//*
```

***Note:** Insert your own dataset names and TG 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.

Figure 4-13: Example UNIX Script to Receive Data

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx reset \
“transfer=(name=yourname                                               \
   receive=./path/to/the/file/to/receive.txt                            \
   receiveuserid=TGxxxxx                                                \
   receiveclass=xxxxxxxx)”
```

Note: Backslashes are being used at the end of each line for line continuation. Double quotes or no quotes may 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.

Figure 4-14: Example AS/400 Commands to Receive Data

```
crtpf file(ea148lib/ncsrecv) rcdlen(80) filetype(*src)

*************** Beginning
NETWORK=NCSPORTAL
FTPUSERID=TGxxxxx
FTPPASSWD=PASSWORD RESET
TRANSFER=(NAME=yourname
RECEIVE=EA148LIB/RECEIVE
RECEIVECLASS=messageclass)
****************** End o

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

crtpf file(ea148lib/receive) rcdlen(80) filetype(*src)
```

Figure 4-15: Example Open VMS Commands to Receive Data

```
Ea2kcmd “network=SAIGPORTAL ftpuserid=TGxxxxx ftppasswd=PASSWORD reset
 “transfer=(name=’yourname’ receive=./path/to/the/file/to/receive.txt)”
```

Query List

The QUERY_LIST command is used in the Transfer statement. It creates a list of available files from the User ID (TGxxxxx) that you specify. Additionally, it gives the STATUS (Received, Accepted, Rejected) of each file. You can specify a DD file name to store the query list. (See Figure 4-13 for JCL example.)

LIST.FIL is the default file that is created for a QUERY_LIST and is stored in the Temp subdirectory. It creates two records for each file queried. You will see LIST.FIL referenced in the output logs for debugging.

Example of query list record:

```
\TG50000\TG40000\EAPS01OP\U\\\DD:EDIPDS(E0000008)\\\29501\TG50000\295011761
51633026581\20010625151607\1\1523\\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.

See Figure 4-16 on the next page for specific field names and descriptions.

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

Figure 4-16: Description of Fields for a Query List Record

#	Field Name	Description
0	VERSION	EA version; e.g. 2K.01.39 *(Send only)*
1	SENDER	TG number of the Sender (active user)
2	RECVR	TG 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 *(Send Only).*
6		Not used
7	ORIGFILENAME	Temporary Send file name from EDIPDS file.
8		Not used
9		Not used
10	SESSIONID	EAFTP generated session ID
11	USERID	Userid 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		Not used
17	CHARFORMAT	A = ASCII, I = BINARY
18	STATUS	Intransit, available [received, accepted, rejected]
19	GETDATETIME	EAFTP generated (ctime)
20	GETDURATION	EAFTP generated
21		Not used
22		Not used
23		Not used
24		Not used
25		Not used
26		Not used
27		Not used

Figure 4-17: Example Query List JCL

```
//STEP0020  EXEC PGM=EA2KMVSC,REGION=4M,TIME=1440,
//          PARM='CMDFILE=DD:CMDSEND'
//STEPLIB   DD DSN=your.dataset.prefix.EALOAD,DISP=SHR
//*
//EASYACC   DD DSN=your.dataset.prefix.EASYACC.INI,DISP=SHR
//*
//EXFER     DD DSN=your.dataset.prefix.EXFER.INI,DISP=SHR
//*
//CMDSEND   DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
QUERY_LIST 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=60
//*
//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=60
//*
//COMPLOG   DD SYSOUT=*
//*
//DCMPLOG   DD SYSOUT=*
//*
//OUTMSG    DD SYSOUT=*
//*
//SYSPRINT  DD SYSOUT=*
//*
//EAFTPLOG  DD SYSOUT=*
//*
//EALOG     DD SYSOUT=*
//*
//EXFERLOG  DD SYSOUT=*
//*
```

Figure 4-18: Example Query List AS400

```
Network=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=PASSWORD RESET
QUERY_LIST QUERY_FILE=DD:QUERY
```

Header and Trailer Record Layouts

***SFA to the Internet Migration Note:* File and Transmission Header and Trailer records will be handled exactly the same under SFA to the Internet as they currently are under TIV WAN over the GEIS network.**

The use of O*N05 and O*N95 header and trailer records is required. O*N01 and O*N99 header and trailer records are now optional. The File Header record (O*N01) identifies the beginning of the transmission. The File Trailer record (O*N99) identifies the end of the transmission. 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 two header and two trailer records for send files. Please refer to Appendix B, Example 7, to add the optional O*N01 and O*N99 to the receive files.

The first record in the transmission is a File Header (‘O*N01’) record. Only one File Header record is sent with each connection. The next record following the File Header 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. The File Trailer (‘O*N99’) record is entered last in the data stream. Figure 4-14 shows two batches of data being sent to TGxxxxx for a message class of MSGCLASS.

Figure 4-19: Example Data File with Headers and Trailer

```
O*N01                                                  73
O*N05TGxxxxx       ,CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,NCT=000
your input data for DEST1 is inserted here
O*N95TGxxxxx        CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,NCT=000
O*N05TGxxxxx       ,CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,NCT=000
your input data for DEST2 is inserted here
O*N95TGxxxxx       ,CLS=MSGCLASS,XXX,BAT=xxxxxxxxxxxxxxxxxxxxxxxxxx,NCT=000
O*N99                                                  73
```

The File Header and File Trailer records require the following data:

- ***Record Identifier:*** Use O*N01 for File Header; use O*N99 for File Trailer.
- ***File Type:*** Specifies the code for the data type; 73 should always be used for compressed data.

Figure 4-20: File Header and Trailer Record Layouts

Column	Length	Entry
1-5	5	*Record Identifier* Use O*N01 for File Header and O*N99 for File Trailer.
6-53	48	Blank
54-55	2	*File Type* (Always 73)
56-EOR	Variable	EOR=The end of record based on the message class being transmitted. Blank space is appended to make the header and trailer the same length as the data records. Must be a minimum record length of 70 characters.

Note: All header and trailer records are required to be a minimum record length of 70 characters. All records must be FB (fixed block) and padded with spaces to match the record length of your data records. Please do not send null values.

Transmission Header and Trailer Record Layouts

The Transmission Header record identifies the beginning of a group of input data records destined for a SAIG mailbox. The Transmission Trailer record identifies the end of this group of records. See Figure 4-16 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.
- ***Message Class:*** An eight-character label assigned to a particular type of data.

Figure 4-21: Transmission Header and Trailer Record Layouts

Column	Length	Entry
1-5	5	*Record Identifier* Use O*N05 for Transmission Header and O*N95 for Transmission Trailer.
6-12	7	*Destination Mailbox ID* This field must have the same value on both the header and trailer record.
13-19	7	SPACES
20	1	‘,’ (Comma)
21-24	4	CLS =
25-32	8	Message Class This field must have the same value on both the header and trailer record.
33-37	5	,XXX, (Always)
38-41	4	BAT=
42-67 Variable	1-26 Variable	Up to a twenty-six character batch number. If not used, leave one space. If using all 26 characters then adjust the remainder of the record layout accordingly. Note: A comma must follow this number (see next field).
43-68 Variable (Based on length of Batch Number.)	1	‘,’ (Comma) Note: Must come directly after Batch Number.
69-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.

Note: All header and trailer records are required to be a minimum record length of 70 characters. All records must be FB (fixed block) and padded with spaces to match the record length of your data records. Do not send null values.

Appendix A: Command Lines/JCL for Different Methods of Sending Data

1) Sending a single file to multiple destination points does not require a JCL change. You need to have multiple O*N05 and O*N95 records around each set of data and use the O*N05TGxxxxx in the header and O*N95TGxxxxx in the trailer to specify the destination point for each set of data.

2) Sending multiple files to multiple destination points within a single transmission requires a change to your JCL. You must specify multiple TRANSFER commands and a corresponding input file for each Transfer.

Example JCL:

```
//CMDSEND   DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name1 SENDUSERID=TGxxxxx
  SEND=DD:FILENAM1
  OTHER_COMP_PARMS='SECFILE=DD:SECFILEX')
TRANSFER=(NAME=name2 SENDUSERID=TGxxxxx
  SEND=DD:FILENAM2
  OTHER_COMP_PARMS='SECFILE=DD:SECFILEX')
//*
//FILENAM1  DD DSN=YOUR.DATASET.NAME,DISP=SHR
//*
//FILENAM2  DD DSN=YOUR.OTHER.DATASET.NAME,DISP=SHR
//*
```

Appendix B: Command Lines/JCL for Different Methods of Receiving Data

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

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password
RESET
TRANSFER=(NAME=name RECEIVE=DD:name
     RECEIVEUSERID=TGxxxx)
```

Example 2: Receive all data by specific message class: RECEIVECLASS= the message class of the data being requested.

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name RECEIVE=DD:name
     RECEIVECLASS=messageclass)
```

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

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name RECEIVE=DD:name)
```

Example 4: Receive data by unique file name: RECEIVE_SERVER_FILE= the Unique Filename on OSM (Online SecureManager) or the Available status record within your Query List.

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name RECEIVE=DD:name
     RECEIVE_SERVER_FILE=xxxxxxxxxxxxxxxxxxxx)
```

Example 5: Delete data by unique file name: DELETE_SERVER_FILE= the Unique Filename on OSM (Online SecureManager) or the Available status record within your Query List.

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name RECEIVE=DD:name
     DELETE_SERVER_FILE=xxxxxxxxxxxxxxxxxxxx)
```

Appendix B: Command Lines/JCL for Different Methods of Receiving Data (Continued)

Example 6: Receive multiple files by specific message class: Use multiple TRANSFER statements with RECEIVECLASS= the message class of the data being requested.

```
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=xxxxxxxx RESET
TRANSFER=(NAME=name1 RECEIVE=DD:RECVFL1
     RECEIVECLASS=SARA02OP)
TRANSFER=(NAME=name2 RECEIVE=DD:RECVFL2
     RECEIVECLASS=CORR02OP)
//*
//RECVFL1   DD DSN=your.dataset.SARA02OP.RECV,
//             DISP=(MOD,CATLG),UNIT=SYSDA,
//             DCB=(LRECL=2850,BLKSIZE=25650,RECFM=FB),
//             SPACE=(CYL,(nn,nn))
//RECVFL2   DD DSN=your.dataset.CORR02OP.RECV,
//             DISP=(MOD,CATLG),UNIT=SYSDA,
//             DCB=(LRECL=450,BLKSIZE=4500,RECFM=FB),
//             SPACE=(CYL,(nn,nn))
```

Appendix B: Command Lines/JCL for Different Methods of Receiving Data (Continued)

Example 7: Concatenate the O*N01 File Header and O*N99 File Trailer records into your Receive data file using IEBGENER.

```
//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.EALOAD,DISP=SHR
//*
//EASYACC   DD DSN=your.dataset.prefix.EASYACC.INI,DISP=SHR
//*
//EXFER     DD DSN=your.dataset.prefix.EASYACC.EXFER.INI,DISP=SHR
//*
//CMDRECV DD *
NETWORK=SAIGPORTAL FTPUSERID=TGxxxxx FTPPASSWD=password RESET
TRANSFER=(NAME=name RECEIVE=DD:receivefile
   RECEIVEUSERID=TGxxxxx)
//*
//receivefile  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
//*
```

Appendix B: Command Lines/JCL for Different Methods of Receiving Data (Continued)

```
//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)
```

Appendix C: UNIX Scripts for Different Methods of Receiving Data

Example 1: Receive data by a specific sender ID: RECEIVEUSERID= the sender of the data being requested. Depending on options in the easyacc.ini file it will either all be concatenated into one file, or each item will go into its separate file named receive.txt.xxx where xxx can be any number from 001 to 999.

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx reset \
“transfer=(name=xxxxxxxx                                                \
   receive=./path/to/the/file/to/receive.txt                             \
   receiveuserid=TGxxxxx)”
```

Example 2: Receive all data by a specific message class: RECEIVECLASS= the message class of the data being requested. Depending on options in the easyacc.ini file it will either all be concatenated into one file, or each item will go into its separate file named receive.txt.xxx where xxx can be any number from 001 to 999.

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx 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 OSM (Online SecureManager) or the Available status record within your Query List.

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx reset \
“transfer=(name=xxxxxxxx                                                \
   receive=./path/to/the/file/to/receive.txt                             \
   receive_server_file= xxxxxxxxxxxxxxxxxxxx)”
```

Appendix C: UNIX Scripts for Different Methods of Receiving Data (Continued)

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

```
ea2khp10c network=saigportal ftpuserid=TGxxxxx ftppasswd=xxxxxxxx reset \
“transfer=(name=xxxxxxxx                                               \
   receive=./path/to/the/file/to/receive.txt)”
```

Appendix D: Trouble Shooting

Errors Received When Sending and Receiving Data

Listed below are common errors received when sending and receiving data. We are providing you with common resolutions to these errors.

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=SAIGPORTAL FTPUSERID=TG50000 FTPPASSWD=PASSWORD RESET TRANSFER=(NAME=anything SEND=DD:SENDFILE **SENDUSERID=TG50000** OTHER_COMP_PARMS='SECFILE=DD:SECFILEX')
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.
RC=32 **Error writing output file DD: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	421 peer closed the connection when receiving files. A small glitch in the internet connection appears to be responsible. Connect again and you should receive as normal.

Errors Received When Sending and Receiving Data (Continued)

Error	Resolution
Autoextent function not working properly	When setting options in the Easyacc.ini file to make separate files for every file received(AUTOEXT=Y APPEND=N), the files aren't being created correctly. The 05 header is being 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.
B37 **U4083; SOC4; CEE3250C** **The system or user abend SB37 R=00000004 was issued. From entry point DCCloseOutputFile at compile unit offset.**	Solutions include going to tape, removing the release command(with VAN reference), or making the following JCL changes: preallocate a dataset, add a step to receive the information from SAIG, add another step to copy the data into a GDG, and then a final step to delete the preallocated data set. See the appendix for JCL example.
CEE3512S-message	Caused by not having valid USS home directories setup for the user IDs. According to IBM, the system was searching the USS files when trying to load COMMPRSS, but was failing when searching the user's home directory. In later releases of OS/390 the search continues to MVS locations such as steplib, Ipa, and linklist, but 2.6 causes an abend.
EDC81281 Connection Refused	In MVS 2.8 LE 1.9+, an LE(Language Environment) error. IT staff need to look at the LE setup.

Errors Received When Sending and Receiving Data (Continued)

Error	Resolution
MVS v2.6 Only **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: DD:WORK01.001**	Add DSN=&&WORK01(etc) to the WORK01 DD, WORKxx, and SYSUTx DDs in the RECV job.
RC=U4038	This is usually an OMVS security error. Resubmitting the job is successful. Btrade recommends following the installation and customization instructions in the OS/390 VnRn.n UNIX System Services Planning manual, which can be found in the IBM documentation CDs(usually disk 1) in the OS/390 VnRn.n UNIX System Services Base Element Bookshelf. Be sure to use the manual that corresponds to your operating system release. If your site is non-UNIX, and you don’t wish to set up OMVS for your users, B-trade recommends setting up a default OMVS segment for all users, so batch programs that make use of OMVS Services and Functions can be run.

Errors Received When Sending and Receiving Data (Continued)

Error	Resolution
<u>**UNIX only**</u> **Id.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.
WARNING: Action failed **Unable to RESET EasyAccess Restart file** **Failed to open EA Restart file [./eastatus.txt] in write mode** **permission denied\**	Solution is to create the eastatus.txt in the EASYACC.INI directory, with read & write permissions.

Index