



User Guide

EasyAccess2000 Command Line

bTrade.com USER GUIDE

Document Approval

Release 1.0

Document Name: EasyAccess2000 (v. 1.38) Command Line User Guide

Released By: Bill Morgan

Date: 15-Feb-2001

Content Editor(s): Phil Arcuri

Technical Author: Linda Fowler

EasyAccess2000 (v. 1.38) Command Line User Guide

Release and Revision History

Document Contact:

Phil Arcuri
 bTrade.com
 Telephone: 972-580-2916

Release History

Document Number	Published Date
bTrade0054-1	15-February-2001

Revision History

Document Title: EasyAccess2000 (v. 1.38) Command Line User Guide					
Change Date	Release/Version #	Change Request#	Description	Resource	Pages / Sections Affected
12 Feb 2001	a	N/A	Initial Document	Phil ARcuri	All
12 Feb 2001	b	n/a	Attach template; initial format	Linda Fowler	All
13 Feb 2001	c	n/a	Initial format	Linda Fowler	All
13 Feb 2001	d	Bill M	Title change; s/w version number	Linda Fowler	All
15 Feb 2001	e	Taylor M	Callable Link Interface section	Linda Fowler	new section

WARNING! COPYRIGHTED MATERIAL. DO NOT COPY WITHOUT PERMISSION.

All rights reserved. This document, including any writing, drawings, notes, or verbal representation made or shown in the course of this communication are confidential and proprietary to bTrade.com. No part of the materials included in this communication should be: 1) reproduced; b) published in any form by any means, electronic or mechanical, including photocopy or information storage or retrieval system; or c) disclosed to third parties, without the express written authorization of bTrade.com.

EasyAccess2000™, SecureManager™ (formerly KeyManager2000™) and Comm-Press2000™ are trademarks of bTrade.com. All other brand and product names are trademarks or registered trademarks of their respective companies.

Contents

INTRODUCTION.....	2
Cancelling a Running Transfer (Command Line Only)	2
Keywords for Supported Command-Line	2
Keyword Usage Guideline.....	2
COMMAND-FILE Usage	3
Keywords for Specifying Tasks to be Performed	4
Keyword Usage Guideline.....	4
Keywords for Overriding Network Parameters from the Command-Line or -file...6	6
Keyword Usage Guideline.....	6
Keywords for the Execution of File Transfers from the Command-file.....9	9
Keyword Usage Guideline.....	9
Keywords for Overriding the Default Parameters Used to Send Files	11
Keyword Usage Guideline.....	11
Keywords for Overriding the Default Parameters Used to Receive Files	12
Keyword Usage Guideline.....	12
Keywords for Specifying Send/Receive Pre/Post Processing While Creating a Transfer	12
Keyword Usage Guideline.....	13
Keywords for PRE_SEND=, POST_SEND=, PRE_RECEIVE=, and POST_RECEIVE=.....	13
Keyword Usage Guideline.....	13
Keywords for Handling Dial Connections (Windows platforms only)	15
Keyword Usage Guideline.....	15
Keywords for Specifying Auto-Retry Options	16
Keyword Usage Guideline.....	16
Keywords for Adding/Modifying Entries in a Trading Partner Address Book.....16	16
Keyword Usage Guideline.....	16
CALLABLE LINK INTERFACE	20

INTRODUCTION

This document contains an example command-file for the EasyAccess 2000 Client v. 1.38.

Canceling a Running Transfer (Command Line Only)

To cancel an in-progress transfer, create the file 'cancel.fil' in the EasyAccess 'temp' subdirectory. The EasyAccess program will terminate the transfer if it can. If it is able to respond, EasyAccess will terminate with a return code of 2.

Keywords for Supported Command-Line

The following keywords may appear on the command-line (only) as the EasyAccess program is being invoked: directory.

(All other keywords may appear on the command-line or in a command-file).

Keyword Usage Guideline

INIPATH=

Override location of EasyAccess 'root' directory

This is the directory containing the easyacc.ini and exfer.ini files as well as the various sub-directories required (e.g. security, runtime, temp and so on). If the INIPATH= keyword is not specified, then the program assumes the current working directory is the EasyAccess root

CMDFILE=

Specify the location of an EasyAccess command-file to be used.

The command-file will completely control the work done during the program run. See below for a summary of supported keywords and an example. When running the command-line version of EasyAccess, 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 using the GUI or by hand via an editor and is specified in the easyacc.ini and exfer.ini files.

Note

Either the CMDFILE= or the IEBASE keywords may be used, but not both together during a given program run.

IEBASE

Executes "iebase" functionality: reads and parses an "IBM EXPEDITE-style" file, basein.msg, and creates transfers to be executed. Note that either the CMDFILE= or the IEBASE keywords may be used, but not both together during a given program run. Note this keyword requires no value.

HELP

Displays on-line usage guidelines.

Note

This keyword requires no value.

RESET

Instructs EasyAccess to ignore any previously failed Transfers which would it otherwise attempt to restart.

MODE= BATCH or GUI

Applies to GUI version of EasyAccess only. Specifies that the GUI program is to execute in command-line mode. MODE=GUI is the default, causing the GUI to execute in GUI mode.

DISABLE_DIALER (Windows only)

Instructs EasyAccess to turn off all logic pertaining to DialUp Networking. Useful if your system does not have the Windows DLLs required by EasyAccess for establishing dial-up connections.

COMMAND-FILE Usage

The following keywords may appear in a command-file specified using the CMDFILE= command-line keyword as the EasyAccess program is being invoked.

Notes

1. Text comments to the right of any '#' character is 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: TRANSFER= "My transfer" is the same as TRANSFER="My Transfer", 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.
 5. 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).
 6. 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 LOGINUSERDID=joe_blow is the same as LOGINUSERDID=Joe_Blow.
 7. Up to 25 transfers may be CREATED within the command-file. Each transfer created is added to the list of transfers to be run.
-

8. Up to 25 EXISTING transfers may be specified in the command-file. Each specified existing transfer is added to the list of transfers to be run.
 9. An unlimited number of trading partners may be added to your Trading Partner Address book using the TPBOOK= keyword.
-

Keywords for Specifying Tasks to be Performed

Keyword Usage Guideline

NETWORK=

See section below for full details on the usage of this keyword.

Specifies an existing network is to be used as the basis for the communications session. The text must match one of the networks already defined in your easyacc.ini file. This keyword tells EasyAccess to use the specified network in the easyacc.ini file as the current network.

Note

Any overrides to this network's data (via the keywords described below) will only apply for a single program run unless the SAVE keyword is also used.

TRANSFER=

See the section below for full details on the usage of this keyword.

Used to specify an existing transfer, or create a new transfer, and add it to the list of transfers to be run.

To specify an existing transfer, use the form:

TRANSFER=name

or

TRANSFER="name"

(quotes are required if 'name' contains spaces).

To create a new transfer, use the form:

TRANSFER=(keyword=value keyword=value... keyword=value)
where all the various keywords that define the transfer appear within the (required) parentheses.

VALIDATE_TRANSFERS_ONLY

Causes EasyAccess to validate the specified transfers (only), report on their validity, and exit.

TPBOOK=

See section below for full details on the usage of this keyword.

TPBOOK=(.....)

Specifies that an entry in the Trading Partner Address Book is to be added or edited.

Here is an example command-file entry to create/modify a Trading Partner:

TPBOOK=(TPNAME=MyPartner
NETWORK1=Btrade.com
MAILBOX1=MyPartnersMailbox
NETWORK2='IGN-I/E SSL'

MAILBOX2=CMAP

MyPartnersIGNAccount

RETRY=

See section below for full details on the usage of this keyword.

Y or N

Enables or disables Auto-Retry.

RECEIVE_RUNTIMES

Instructs EasyAccess to create and execute a transfer to receive your security runtimes from a location designated in the SECURITY section of your easyacc.ini file. The runtimes file is then automatically installed giving EasyAccess access to the public keys of your trading partners.

GENKEYS

Instructs EasyAccess to generate a public/private key pair, to bundle these into a certificate request, and to execute a create and execute a transfer to send the certificate request to a location designated in the SECURITY section of your easyacc.ini file.

PASSLOC=

Specifies the passphrase location for encrypting the keys. Provides user with ability to specify the location of a security token. If not specified, then a default value is used.

QUERY_LIST

Instructs EasyAccess to create and execute a transfer to receive a list of available files from the current server.

QUERY_FILE=

Specifies the qualified file name of the file to receive the server file list. If not present, the file list is written to the default file, list.fil, in the EasyAccess 'temp' directory.

RECEIVE_AUDIT_LOGS

Instructs EasyAccess to create and execute a transfer to receive an audit report from the current server of files sent and received.

AUDIT_START_DATE=

Used in conjunction with RECEIVE_AUDIT_LOGS to specify the interval of time the audit report should cover.

AUDIT_END_DATE=

Used in conjunction with RECEIVE_AUDIT_LOGS to specify the interval of time the audit report should cover.

AUDIT_FILE=

Specifies the qualified file name of the file to receive the audit logs. If not present, the audit logs are written to the default file, audit.log, in the EasyAccess 'temp' directory.

DIAL= (Windows only)

See section below for full details on the usage of this keyword. Text name of one of the DialUp Networking entries you have previously set up on your computer.

Keywords for Overriding Network Parameters from the Command-Line or -file

Keyword Usage Guideline

NETWORK= Text

Specifies an existing network is to be used as the basis for the communications session. The text must match one of the networks already defined in your easyacc.ini file. This keyword tells EasyAccess to use the specified network in the easyacc.ini file as the current network. Note that any overrides to this network's data (via the keywords described below) will only apply for a single program run unless the SAVE keyword is also used.

IP= Hostname or IP address

Overrides the server address.

SMTP_SERVER=Same as IP. (AS1 and EMAIL)

Hostname or IP address of SMTP server.

Note

The SMTP server should be configured to handle messages with signatures and encrypted messages. Some SMTP servers, such as Microsoft's Exchange Server must be specifically configured to allow signed messages to pass thru the SMTP server without being modified by the server.

URL= Same as IP. (AS2)

Hostname or IP address of AS2 HTTP or HTTPS server.

IP2= Hostname or IP address.

Overrides the backup server address.

POP3_SERVER= Same as IP2. (AS1)

Hostname or IP address of POP3 server.

URI= Same as IP2. (AS2)

URI portion of AS2 server address

USERID= Text name

Overrides the server login userid.

POP_USERID=Same as USERID

Overrides the POP3 server login userid.

PASSWD= Text

Overrides the server login password.

Notes

1. If you use this keyword, your password is entered as clear text on the command-line or from the command-file. YOU SHOULD CHECK WITH YOUR SECURITY MANAGER BEFORE USING THIS KEYWORD.
2. If the network server supports the ability to change your password, this may be accomplished using either `PASSWD=old-password/new-password`, or `PASSWD=old-password/new-password/new-password`, depending on the syntax expected by the network server.

POP_PASSWD= Same as PASSWD

Overrides POP3 server login password.

Notes

1. If you use this keyword, your password is entered as clear text on the command-line or from the command-file. YOU SHOULD CHECK WITH YOUR SECURITY MANAGER BEFORE USING THIS KEYWORD.
2. If the network server supports the ability to change your password, this may be accomplished using either `PASSWD=old-password/new-password`, or `PASSWD=old-password/new-password/new-password`, depending on the syntax expected by the network server.

EMAIL_ADDRESS Text

Your email address. For EMAIL and AS1 styles, your email address is put in the 'From:' field in all messages created.

For AS1, your email address is the default address used when specifying where Message Disposition Notifications (MDNs) should be sent by the receiver of messages you send.

For AS2, the email address is used merely to indicate that an MDN is being requested. This is a required field for AS1 and AS2; it is stored in the [SECURITY] section of your easyacc.ini file.

NETSTYLE=

Overrides the communications style used to communicate with the server.

Supported FTP styles are:

- "GENERIC"
- "GENERIC-DOS"
- "GENERIC_SSL"
- "IGN-IE",
- "FEDEXNET"

- "WALMART"
- "EAFTP"
- "MARK_III" (GEIS)
- "EDI*Express" (GEIS)
- "EDISwitch" (GEIS)
- "CONNECTMAIL" (Sterling), "Sterling-Commerce"
- "MCI-Edi*Net"

Supported SMTP/POP3 styles are:

- "EDI-INT"
- "GISB-CLIENT"
- "GISB-SERVER"

Supported style for compression/encryption without any file transfer is:

- "LOCAL-ARCHIVE"

CASE=U or L

Overrides the case-sensitivity setting for the network. CASE=U tells EasyAccess to convert everything sent to the server to upper case. CASE=L tells EasyAccess to send data to the server unchanged.

COMMAND_OVER_DATA=Y or N

EAFTP style only.

Note

1. If 'Y', then uses "command-over-data" variant of FTP, which uses a single socket connection.
 2. If 'N', then uses conventional FTP command and data socket connections.
-

CONTROLPORT= Numeric value

Overrides the Command-channel port number for communicating with the server (FTP only).

PASSIVE= Y or N

Overrides the "passive" mode setting for the session (FTP only). PASSIVE=Y is used in certain circumstances to allow communications through a firewall.

SSL=Y or N

Tells EasyAccess whether to use SSL 3.0 when establishing a session with the server (FTP only).

SITEDELAY= Numeric value

Specifies a value, in seconds, to wait prior to sending each command (FTP only). A value of zero is usual. Infrequently a server will require a delay to handle certain timing problems.

SAVE

Instructs EasyAccess to save the network data you specify on the command-line or in the command-file in the easyacc.ini file, causing the data to be permanently in effect until you change it.

Note

The default is NOT TO SAVE THE NETWORK DATA.

In the default case, the network data only applies for the duration of the current program run. This is not true for transfers that are created on the command-line or command-file; these are written to the exfer.ini file and are automatically available to you in subsequent runs of the program.

SAVE_ONLY

Acts like the SAVE keyword, except the program exits after saving the specified network data in the easyacc.ini file.

Keywords for the Execution of File Transfers from the Command-file

Keyword Usage Guideline

TRANSFER=

Used to specify an existing transfer, or create a new transfer, and add it to the list of transfers to be run. To specify an existing transfer, use the form:

TRANSFER=name

or

TRANSFER="name" (quotes are required if 'name' contains spaces).

To create a new transfer, use the form:

TRANSFER=(keyword=value keyword=value... keyword=value)
where all the various keywords which define the transfer appear within the (required) parentheses.

NAME= REQUIRED

Specifies the name of the transfer

LOGINUSERID= Used with LOGINPASSWD

Specifies a server login other than the default network login is to be used prior to executing the transfer.

LOGINPASSWD= Used with LOGINUSERID

Specifies a server login other than the default network login is to be used prior to executing the transfer.

Note

1. If you use this keyword, your password is entered as clear text on the command-line or from the command-file. YOU SHOULD CHECK WITH YOUR SECURITY MANAGER BEFORE USING THIS KEYWORD.
-

- 2. If the network server supports the ability to change your password, this may be accomplished using either LOGINPASSWD=old-password/new-password, or LOGINPASSWD=old-password/new-password/new-password, depending on the syntax expected by the network server.
-

SEND=

Specifies the qualified filename of a file to be sent to the server. A transfer may either specify SEND or SENDEDI, but not both.

SENDEDI=

Specifies the qualified filename of an EDI file to be sent to the server.

SENDUSERID=

Name of userid (mailbox) on the server to receive the file being sent. For IGN, if SENDEDI is specified, this keyword specifies the Alias Table to be used.

TO_EDINAME=Same as SENDUSERID

For AS1 and AS1 must be an entry in the Trading Partner address book.

SENDCLASS=

Specifies the Class or APRF to receive the file being sent. For EMAIL, AS1 and AS2, specifies the Subject line in the message being sent.

SENDAPRF=

Same as SENDCLASS.

SENDSUBJECT= (EMAIL, AS1)

Same as SENDCLASS.

TO_ARCHIVE=(DATAGUARD only)

Specifies directory and file name to receive the secured file.

RECEIVE=

Specifies the file to receive a mailbox entry being downloaded from the server.

RECEIVEEDI=

Specifies the file to receive an EDI mailbox entry being downloaded from the server.

RECEIVEUSERID=

Specifies that only files sent to your mailbox from the specified userid are to be downloaded.

RECEIVECLASS=

Specifies that only files sent to the specified Class or APRF are to be downloaded. For EMAIL and AS1, specifies that only files with the specified subject-line are to be downloaded.

RECEIVEAPRF=

Same as RECEIVECLASS.

RECEIVESUBJECT= (EMAIL, AS1)

Same as RECEIVECLASS.

FROM_ARCHIVE= (DATAGUARD only)

Specifies directory and file name of the secured file to be accessed.

Keywords for Overriding the Default Parameters Used to Send Files

Keyword Usage Guideline

COMPRESS=Y or N

Tells EasyAccess whether to compress a file before it is sent to the server.

SECURE=Y or N

Tells EasyAccess whether to use the SECURE option during compression.

FILTER=Y or N

Tells EasyAccess whether to use the FILTER option during compression.

SENDASCII=Y or N

Tells EasyAccess whether to treat the file being sent to the server as an ASCII file.

CRLF= Y or N

Tells EasyAccess whether to use the CRLF option during compression.

DELETE_AFTER_SEND=Y or N

Tells EasyAccess whether or not to delete the file after it has been successfully sent

PERPETUAL_SEND=Y or N

Tells EasyAccess whether or not to make the transfer repeat it's Send-cycle as specified by the RETRY=, MAX_RETRY=, and RETRY_DELAY= keywords.

OTHER_COMP_PARMS=

Specifies advanced Comm-Press compression parameters.

Please see the Comm-Press program documentation for supported keywords. The advanced parameters should be entered just as they would appear on the command-line invocation of the Comm-Press compression program.

Example Usage:

```
"TRANSFER=(NAME=mytransfer OTHER_COMP_PARMS='lrecl=72 delim=250')"
```

Note

The advanced parameters can be delimited using single or double quotes, parentheses, or square or curly braces.

Keywords for Overriding the Default Parameters Used to Receive Files

Keyword Usage Guideline

APPEND=Y or N

Tells EasyAccess to append all downloaded files into the file specified by the RECEIVE or RECEIVEEDI keyword. Either APPEND or AUTOEXT must be Y, but not both.

AUTOEXT=Y or N

Tells EasyAccess to "auto-extend" the filename specified by the RECEIVE or RECEIVEEDI keyword. As each file is received, it is given a filename with a unique numeric extension. Either APPEND or AUTOEXT must be Y, but not both.

RECEIVEASCII=Y or N

Tells EasyAccess whether to treat the file being downloaded from the server as ASCII files.

UNCOMP=Y or N

Tells EasyAccess whether to use the UNCOMP option during decompression. If a file received is not compressed, this parameter must be Y in order to correctly process the file.

PERPETUAL_RECEIVE=Y or N

Tells EasyAccess whether or not to make the transfer repeat its Receive-cycle as specified by the RETRY=, MAX_RETRY=, and RETRY_DELAY= keywords.

OTHER_DECOMP_PARMS=

Specifies additional Comm-Press decompression parameters. Please see the Comm-Press program documentation for supported keywords. The advanced parameters should be entered just as they would appear on the command-line invocation of the Comm-Press decompression program.

Example Usage:

```
"TRANSFER=(NAME=mytransfer OTHER_DECOMP_PARMS='unwrap delim=250')"
```

Note

The advanced parameters can be delimited using single or double quotes, parentheses, or square or curly braces.

Keywords for Specifying Send/Receive Pre/Post Processing While Creating a Transfer

These keywords allow you to tell EasyAccess to execute a program before or after you have it send or receive files, and, you must use these keywords inside a transfer definition.

Keyword Usage Guideline

SEND_VERIFY=Y or N

Tells EasyAccess whether it should check for the existence of the file(s) you are telling it to Send at the time the transfer is being created. The default is to check that the files exist, to catch typing errors and so on. However, if you are executing a Send Pre-processing program which will create the files to be sent, then you will want to disable this checking, since the files may not exist until the transfer is executed.

RECEIVE_VERIFY= (DATAGUARD only) Y or N

Tells EasyAccess to check for the existence of the file(s) you are telling it to Unsecure at the time the transfer is being created. The default is to check that the file(s) exist, to catch typing errors and so on. However, if you are executing an Unsecure (Receive) Pre-processing program, which will create the files to be unsecured, then you will want to disable this checking, since the files may not exist until the transfer is executed.

PRE_SEND=

Specifies a program or command-file is to be run prior to the Send cycle of a transfer. See below for the syntax and related keywords.

POST_SEND=

Specifies a program or command-file is to be run after to the Send cycle of a transfer. See below for the syntax and related keywords.

PRE_RECEIVE=

Specifies a program or command-file is to be run prior to the Receive cycle of a transfer. See below for the syntax and related keywords.

POST_RECEIVE=

Specifies a program or command-file is to be run after to the Receive cycle of a transfer. See below for the syntax and related keywords.

Keywords for PRE_SEND=, POST_SEND=, PRE_RECEIVE=, and POST_RECEIVE=

These keywords all use the following syntax to specify a program or command-line to be executed and how to tell EasyAccess how to check if the program or command-line executed successfully.

Keyword Usage Guideline

CMDLINE=

Specifies the directory and filename of the program, batch file, script file, or operating system command to be executed, along with any arguments to be passed to the program/script/command.

EACH_FILE= Y or N (Post-processing only)

If Y, then EasyAccess will execute the post-processing after each file has been received. If N, then EasyAccess will execute the post-processing after all files have been received to the program/script/command.

RETCODE= Numeric value

Specifies a return code value to be used in determining if the program/script/command executed successfully or not.

SUCCEEDS_IF_GT

Tells EasyAccess that the program/script/command succeeded if the return code is greater than the value specified by the RETCODE= keyword.

SUCCEEDS_IF_LT

Tells EasyAccess that the program/script/command succeeded if the return code is less than the value specified by the RETCODE= keyword.

SUCCEEDS_IF_EQ

Tells EasyAccess that the program/script/command succeeded if the return code is equal to the value specified by the RETCODE= keyword.

SUCCEEDS_ALWAYS

Tells EasyAccess that the program/script/command always succeeds, regardless of its return code.

FAILS_IF_GT

Tells EasyAccess that the program/script/command failed if the return code is greater than the value specified by the RETCODE= keyword.

FAILS_IF_LT

Tells EasyAccess that the program/script/command failed if the return code is less than the value specified by the RETCODE= keyword.

FAILS_IF_EQ

Tells EasyAccess that the program/script/command failed if the return code is equal to the value specified by the RETCODE= keyword.

FAILS_ALWAYS

Tells EasyAccess that the program/script/command always fails, regardless of its return code.

Note

This is only useful in testing, since the pre- or post-processing will fail, causing EasyAccess to consider the transfer to have failed.

The syntax used to specify pre- and/or post-processing is shown in the following example:

```
TRANSFER=(name=...sendclass=.....
```

UNIX example:

```
PRE_SEND=[ CMDLINE='sh -x myScript.ksh 2>err.out' RETCODE=127  
SUCCEEDS_IF_LT]
```

Windows example:

```
POST_SEND=[ CMDLINE='C:\MyPrograms\cleanup.exe /log' RETCODE=0  
SUCCEEDS_IF_EQ ]
```

This transfer will execute a UNIX shell script (myScript.ksh) before the Send step and will execute a Windows program (cleanup.exe) after the Send step. (Obviously, pick the example for YOUR platform; the program won't run a Windows program if you are running it on a UNIX platform).

Keywords for Handling Dial Connections (Windows platforms only)

Keyword Usage Guideline

DIAL=

Text name of one of the DialUp Networking entries you have previously set up on your computer.

DIAL_PROGRAM=

Text filespec (fully qualified filename) of a Dialer program that you want launched (instead of DialUp Networking).

For example, the AT&T Global Network Services Dialer program IDIALER.EXE (plus directory path) may be specified. If both DIAL= and DIAL_PROGRAM= are specified, then the Dialer program will be used.

BACKUP_DIAL=

Text name of a backup DialUp Networking entry you have previously set up on your computer. The backup Dial entry is used if the primary DialUp Networking entry or Dialer program fails to connect.

BACKUP_DIAL_PROGRAM=

Text filespec (fully qualified filename) of a Dialer program that you want launched (instead of DialUp Networking) as your backup. For example, the AT&T Global Network Services Dialer program IDIALER.EXE (plus directory path) may be specified. The backup Dialer program is used if the primary DialUp Networking entry or Dialer program fails to connect. If both BACKUP_DIAL= and BACKUP_DIAL_PROGRAM= are specified, then the Dialer program will be used as the backup.

AUTODIAL=Y or N

Tells EasyAccess whether it should "auto dial" the DialUp Networking entry if no current connection is in progress.

AUTODISCONNECT=Y or N

Tells EasyAccess whether it should "auto-disconnect" the current connection when it exits.

TIMEOUT=Numeric entry

Tells EasyAccess how long, in seconds, it should wait for a dial attempt to connect before it decides the attempt has failed.

Keywords for Specifying Auto-Retry Options

Keyword Usage Guideline

RETRY=Y or N

Enables or disables Auto-Retry.

MAX_RETRY= Numeric entry

Tells EasyAccess how many times to retry a filed transfer.

RETRY_DELAY=Numeric entry

Tells EasyAccess how long, in seconds, it should wait between retry attempts.

Keywords for Adding/Modifying Entries in a Trading Partner Address Book

Keyword Usage Guideline

TPBOOK= TPBOOK=(.....)

Specifies an entry in the Trading Partner Address Book is to be added or edited. Here is an example command-file entry to create/modify a Trading Partner:

```
TPBOOK=( TPNAME=MyPartner NETWORK1=Btrade.com  
MAILBOX1=MyPartnersMailbox NETWORK2='IGN-I/E SSL'  
MAILBOX2=CMAP.MyPartnersIGNAccount
```

TPNAME=

Specifies the Trading Partner name to be added/modified

NETWORK1=

Specifies the Primary Network to be used when sending to or receiving from this Trading Partner.

MAILBOX1=

Specifies the Primary mailbox, or user ID, or login name for the Trading Partner on the Primary Network.

NETWORK2= (Optional)

Specifies the Backup Network to be used when sending to or receiving from this Trading Partner. The Backup Network is used only if a transfer fails using the Primary Network and Mailbox (if Auto-Retry is enabled). The Backup Network will be used for the last half of the specified retries. That is, if Auto-Retry is enabled and MAX_RETRY is 2, then when a transfer fails, the first retry will use the Primary Network and the second retry will use the Backup Network.

MAILBOX2= (Optional)

Specifies the Backup Mailbox, or user ID, or login name for the Trading Partner on the Backup Network.

Example of an EasyAccess command-file:

(Remove leading '#' character to activate any given line)

```
NETWORK=Btrade.com # Select the network to use (do only once)
USERID=myUserId # Override the server login userid
PASSWD=myPassWord # Override the server password
Create some transfers, invoke other transfers already Stored
```

TRANSFER="MY RECEIVE TEST"

Execute existing 'MY RECEIVE TEST' transfer. Note use of quotes for transfer name.

Note

You can use single or double quotes, square brackets, curly braces, or parentheses for transfer creation, and for pre/post processing specification, and for TPBOOK= usage too...transfer=(# Create a new transfer and then execute it name="MY SEND TEST" Name is REQUIRED! send=c:\autoexec.bat senduserid=CPINC03. This server is case sensitive: userid in caps! sendclass=DOMINV. Receive all files currently in DOMINV class COMPRESS=Y SENDASCII=Y CRLF=Y FILTER=Y SECURE=N pre_send= [cmdline='dir *.*' retcode=0 succeeds_always] post_send= [cmdline='sh -x /home/user/cleanup.ksh -h -l=60' each_file=y retcode=0 succeeds_if_eq])

End transfer creation

Transfer= (Name=SendInvoice send=c:\inv\invoices.txt SendUserId=CPINC03 SendAPRF=INV)

Transfer= (name=ReceiveINV receive=c:\inv\new_inv.txt receiveuserid=CPINC03 receiveclass=INV autoext=y ascii=y append=n)

Tell EasyAccess to get a list of available files from the server. The file list is written to file myaudit.log. If the auditFile keyword is not specified, then the file list is written to the default file, list.fil in the EasyAccess 'temp' directory.

Note

No other transfers will be executed during the program run if the "queryList" keyword is specified -- it supersedes the execution of all send/receive transfers.

queryList queryFile=myFiles.lst

Tell EasyAccess to get an audit report from the server showing all files sent and received from/to the current login during the period from 11/1/1999 thru 12/1/1999. The audit report is written to the file audit.log in the EasyAccess "temp" directory.

Note

No other transfers will be executed during the program run if the "receiveAuditLogs" keyword is specified -- it supersedes the execution of all send/receive transfers.

receiveAuditLogs

auditStartDate=19991101

auditEndDate=19991201

auditFile=myaudit.log

Tell EasyAccess to create a public/private key pair, generate a certificate request, and send the request to the configured Certificate Authority for approval (as specified in the SECURITY section of the easyacc.ini file). In

addition, specify a pass-phrase location to store the key-encrypting key, used to provide security-token capabilities.

Note

No other transfers will be executed during the program run if the "genkeys" keyword is specified -- it supersedes the execution of all send/receive transfers.

genkeys passloc=[a:/mytoken.txt]

Tell EasyAccess to create and execute a transfer to receive the users runtime files (previously generated by the configured Certificate Authority as specified in the SECURITY section of the easyacc.ini file). Once the runtime files are received, they are automatically installed.

Note

No other transfers will be executed during the program run if the "receive_runtimes" keyword is specified -- it supersedes the execution of all send/receive transfers.

receive_runtimes

Create some Trading Partner Address Book entries, and use one in a transfer.

Note

The network2 and mailbox2 entries are optional...

```
TPBOOK=( name=myPartner network1=Btrade.com
mailbox1=MyPartnersMailbox )
```

Note

You can use square brackets, or parentheses for transfer creation, and for pre/post processing specification, and for TPBOOK= usage too...

```
TPBOOK=[
name=myOtherPartnernetwork1=Btrade.commailbox1=MyOtherPartnersMail
boxnetwork2="IGN-I/E SSL"mailbox2=CMAP.MyOtherPartnersIgnMailbox]
```

This transfer will try to send to myOtherPartner on the Btrade.com network and if this fails, will switch to the IGN-I/E SSL network (since auto-retry is enabled below...)

Transfer= (Name=ShowOff send=c:\inv\invoices.txt SendUserId=myOtherPartner SendAPRF=ShowOff)

Show use of dial-up networking (Windows only)

DIAL="Dial Fedex"

Name of existing DialUp Networking entry. If the "Dial Fedex" DialUp Networking entry fails to connect, then launch the following Dialer program to try to connect:

```
BACKUP_DIAL_PROGRAM='C:\Program
Files\AT_T_GlobalDialer\IDIALER.EXE'
AUTODIAL=Y # Do the dial before
trying to connect!
AUTODISCONNECT=Y # Hang up when program
is done
TIMEOUT=300 # If no connection in 3
minutes, then failed
```

Show use of auto-retry

```
RETRY=Y # Enable auto-retry
MAX_RETRY=2 # Retry twice after
initial failure
RETRY_DELAY=10 # Delay 10 seconds
```

CALLABLE LINK INTERFACE

The collection of files in this section includes the ea2kw95s.dll, ea2kwnts.dll for Windows NT Server version, which provides a callable 'C' API for EasyAccess 2000 client functionality. The API consists of a single entry point:

```
int eaCmdMain(char* programName, ...);
```

which takes a variable length argument list. It takes as arguments all of the keywords supported by the EasyAccess 2000 client command-line program.

The first argument, programName, can be any null-terminated character array. It takes the place of the normal 'argv[0]' provided to any 'C' program when it is executed, which is the name of the program being executed.

All the subsequent arguments are expected to be taken from the EasyAccess 2000 command-line keyword set, using the same syntax as if the command-line program was being invoked.

See the command-line documentation example file, easyacc.cmd, for descriptions and examples of all the supported keywords.

See also the short example program, eatest.c, for a working example of how to use the ea2kw95s DLL. This program invokes the eaCmdMain entry point several times to perform several tasks. You may need to modify this example program to reference one of the Network instances in your easyacc.ini file.

If you are using a different type of network style than that demonstrated, you may have to tailor the examples to the capabilities of the server to which you are communicating, since different network styles (styles of server) support different features. For example, not all network-styles support the RECEIVE_AUDIT_LOGS keyword.

The eaCMdMain function returns 0 if it successfully executes the specified tasks; otherwise it returns a nonzero error code.

The ea2kw95s DLL expects to be run in a directory containing the files used by the regular EasyAccess 2000 client programs, including:

easyacc.ini	Network definitions, current tasks, program settings
exfer.ini	Stored transfers
tpaddrss.ini	Trading Partner Address book
commpr32.dll	Compression
ea2kw95d.dll	Communications objects
eaftpw95.dll	FTP transport (with optional SSL)
eapop3.dll	POP3 transport
easmtpl.dll	SMTP transport
rimport.dll	Imports SecureManager-generated 'runtimes'

eahttp.dll	HTTP/HTTPS transport (EasyAccess version 2.0 and above)
easmime.dll	S/MIME messaging (EasyAccess version 2.0 and above)
securesocket.dll	Socket API (EasyAccess version 2.0 and above)

In addition, the following sub-directories are expected:

temp	Directory for temp files and several optional log files
runtime	Directory for certificates, private keys, security tables
security	Directory for certificates, private keys, security tables
maint	Directory for receiving maintenance releases
ingoing	(Optional) directory for incoming files
outgoing	(Optional) directory for outgoing files

Note

The EasyAccess 2000 client programs and the ea2kw95s DLL 'own' the easyacc.ini and other files in the directory tree when they are executing. This means that you cannot execute multiple copies out of the same directory at the same time. The DLLs are all thread-safe, so you can run concurrent copies; however, you must provide separate directory trees for each instance. In each directory tree you must provide the three .ini files and the four required sub-directories. You may keep a single copy of all the DLLs and programs in a separate directory and execute using the 'INIPATH=' keyword when invoking the programs or ea2kw95s DLL to point to the directory you wish to use for any given run.

Note

Callable link interface will be modified for strategic clients based on design specifications jointly developed (such as an AQ wrapper, MQ wrapper, Vitria, etc).
