CA XCOM™ Data Transport® for UNIX and Linux

User Guide

Release 11.6

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2012 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Easytrieve®
- CA Roscoe®
- CA XCOM™ Data Transport®

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Using CA XCOM Data Transport 25

CA XCOM Data Transport GUI 25
Start CA XCOM Data Transport 26
Access CA XCOM Data Transport 26
Start from the Home Page 27
About CA XCOM Gateway 29
Associate CA XCOM Data Transport with CA XCOM Gateway 29
CA XCOM Data Transport Client Authentication 30
Set Other Optional Parameters 31
CA XCOM Gateway Staging Area Access 33
Using CA XCOM Data Transport with CA XCOM Gateway 33
CA XCOM Gateway Transfers 33
Format of CA XCOM Gateway File Names 35
Set Up User Permissions 35
Insert Files into CA XCOM Gateway 36
Extract Files from CA XCOM Gateway to External Server 36
Insert Files from External Server to CA XCOM Gateway 37

Chapter 2: Operating Environment 39

CA XCOM Data Transport Globals 39
PATH and Directory Considerations 40
Transfer Protocols 40
About Parameters 40
Parameter Values 41
Parameter Format 41
Guidelines 42
The xcomd Command 43
Control the Daemon 43
Options 44
Start xcomd 44
Global Parameters 45
Parameters 45
AGE_TIME 45
ATOE_FILENAME 45
AUTH_TYPE 46
CACHE_READ_SZ 46

CACHE_WRITE_SZ 46
CAPKIHOME 47
CODETABL 47
CONVERT_CLASSES 48
CREATE_DIRECTORIES 48
DATACLAS 48
DEFAULT_CHARSET 48
DEFAULT_CONVERROR 49
DEFAULT_DELIM 49
DEFAULT_INPUTERROR 51
DSNTYPE 52
EOL_CLASSES 52
ETOA_FILENAME 53
EXPIRATION_TIME 53
FIPS_MODE 53
GATEWAY_IP 53
GATEWAY_PORT 54
GATEWAY_VERSION 54
GATEWAYCERT 54
GATEWAYCPASS 54
GATEWAYPKEY 55
INTERNAL_CONVERSION_TABLES 55
LCLNTFYL 55
LICENSE_DIR 56
LOG_CONNECT_MSG 56
MAIL_TYPE 56
MAX_QUEUE_ENTRIES 57
MAX_REMOTE_TCP 57
MAX_SESSIONS_ENTRIES 57
METACODE_CLASSES 57
MGMTCLAS 58
PAM_PATH 58
PORT 58
PROTOCOL 58
QUEUE_PATH 59
REMOVE_TRAIL_BLANKS 59
SAVE_PASSWORD_IN_CNF 59
SECURE_SOCKET 60
SESSIONS_FILE 60
SHELL_CMD 60
SMTP_SERVER 60
SOCK_DELAY 61

SOCK_RCV_BUF_SIZE 61
SOCK_SEND_BUF_SIZE 61
STAT_FREQUENCY 62
STORCLAS 62
SYSID 62
SYSNAME 62
TCP_CMD 63
TEMPDIR 63
TP_CMD 63
TRACE_PATH 63
TRNENCRL_CIPHER 64
TRNENCRR_CIPHER 66
TRNENCRR_DHBITS 68
TRUST_DATABASE_NAME 69
TRUST_DATABASE_TYPE 69
TRUST_ODBC 69
TRUST_OWNER 70
TRUST_PASS 70
TRUST_PORT 70
TRUST_SERVER 70
TRUST_TABLE_PFX 70
TRUST_USER 71
TXPI_BUF_SIZE 71
TXPI_RECEIVE_TIMEOUT 71
TXPI_SEND_CHECK_FREQ 71
TXPI_TERM_TIMEOUT 72
UMASK 72
USEROVR 72
XBUFFSIZE 73
XCOM_CONFIG_SSL 73
XCOM_ICUPATH 73
XCOM_JVM 73
XCOM_MYSQLODBC 73
XCOM_ODBC 74
XCOM_SHLIBPATH 74
XCOM_PASSWORD 74
XCOM_SHOW_CIPHER 74
XCOM_TRUSTED_OVR 75
XCOM_USERID 75
XCOMHIST 75
XCOMHIST_BACKSLASH 75
XCOMHIST_FILE 76

XCOMHIST_OWNER 76
XCOMHIST_PASSWORD 76
XCOMHIST_SPLIT_FILE 76
XCOMHIST_TBL 76
XCOMHIST_USER 77
XCOMPRE_LOCAL 77
XENDCMD 77
XLOGFILE 77
XLPCMD 78
XNOTIFYCMD 78
XPPCMD 78
XPRECMD 79
XTRACE 79
Local Session Control Parameters 80
Transfers Using SNA 80
Transfers Using TCP/IP 80
Example 81
Manage the Queue 81
The xcomqm Command 81
Syntax 82
Options 83
Purge the Log File 84
Syntax 84
Example 84
About the xcompre Pre-allocation Exit 85
How to Use xcompre 85
xcompre Options 86
About Post Processing Scripts 88
Shells and Script Changes 88
Process Summary 88
Troubleshooting 88
How to Modify Post Processing Parameters 88
About xcompp 89
How to Modify the Post Transfer Script 91
About xcomend 91
How to Modify Printing Processes 95
About xcomlp 95
How to Customize the Notification Process 97
About xcomntfy 97
Requirements for SMTP Notifies 100
Pluggable Authentication Modules (PAM) Based Authentication 103
PAM Overview 103

Enabling PAM authentication in CA XCOM 103
Configure PAM Service 104
Trusted Access Database 104
Set Up Trusted Access 105
Trusted Security 105
Releases of CA XCOM Data Transport Using Trusted Transfers 105
Reports Containing Machine Code Characters 106
Example 106
About History Records 107
Step 1. Install a Relational Database Management System 107
Step 2. Create a Database 107
Step 3. Create the XCOM History Table 108
Step 4. Modify the XCOM Default Global Table 119
Step 5. Install the Appropriate ODBC Driver 120
Step 6. Configure ODBC 121
Configuring JDBC Database Access 123
Security Considerations for History Records 124
Instructions to Upgrade Existing R11.5 History Database 125
Interdependent Transfers 131
Coding Interdependent Transfers 132
Password Protection by Encryption 133
How Transmission Password Encryption Works 134
Security of Passwords in Traces and XCOM Files 135
Transfer Control (XTC) Parameters 135
HOLD_TRANSFER 135
HOLDCOUNT 136
XTCERRDECR 136
XTCERRINCR 137
XTCERRPURGE 137
XTCERRREL 137
XTCGOODDECR 138
XTCGOODINCR 138
XTCGOODPURGE 138
XTCGOODREL 139
XTCJOB 139
XTCNET 139

Chapter 3: The Command Line Interface 141

Using Configuration Files 142
Creating Configuration Files 143
Symbolic Parameters in Configuration Files 144

&DATE(format-code) 144
&ID 145
&IPNAME 145
&LU 145
&LUSER 145
&TIME(format-code) 145
&USERID 146
How to Use Symbolic Parameters 146
Assigning Values 147
Parameters Supporting Symbolic Variables 148
Transferring Files 148
Specifying Protocols 148
Queuing Transfers 149
Multiple Transfers 149
Using Semicolons 149
Wildcard Characters 149
Syntax 150
Options 151
Transferring Files Using SNA/APPC Protocols 152
Using SNA/APPC Protocols 153
Transferring Files Using TCP/IP Protocols 154
Using IP Addresses and Names 154
Specifying the Remote System 155
TCP/IP Name Resolution 155
Using TCP/IP Protocols with CA XCOM Data Transport 156
PORT 157
PROTOCOL 157
SECURE_SOCKET 158
XCOM_CONFIG_SSL 158
Setting Data Transfer Attribute Parameters 158
List of Parameters 158
COMPRESS 159
MAXRECLEN 160
TRUNCATION 160
VERSION 160
Using the Send File Command 161
Parameters 161
CREATE_DIR 162
FILE_OPTION 163
LOCAL_FILE 164
REMOTE_FILE 165
REMOTE_SYSTEM 166

Creating a File on an IBM Mainframe 166
Parameters 166
ALLOCATION_TYPE 167
AVGREC 168
BLKSIZE 168
COMPRESS_PDS 169
CREATEDELETE 170
DATACLAS 171
DSNTYPE 172
EATTR 173
LRECL 173
MGMTCLAS 173
NUM_OF_DIR_BLOCKS 174
PRIMARY_ALLOC 174
RECORD_FORMAT 174
SECONDARY_ALLOC 175
SPACE 176
STORCLAS 176
UNIT 176
VOLUME 177
Tape Parameters for an IBM Mainframe 177
DEN 177
EXPDT 178
LABEL 178
LABELNUM 179
RETPD 179
TAPE 179
TAPEDISP 180
UNITCT 180
VOLCT 180
VOLSQ 180
Using the Retrieve File Command 181
Parameters 181
FILE_OPTION_RF 182
LOCAL_FILE_RF 183
REMOTE_FILE_RF 184
REMOTE_SYSTEM_RF 184
UNIT_RF 184
VOLUME_RF 185
Using the Send Report Command 185
Parameters 185
CARRIAGE_CONTROL_CHARACTERS 186

CLASS 186
COPIES 187
DESTINATION 187
DISPOSITION 187
FCB 188
FORM 188
HOLDFLAG 188
LOCAL_FILE_SR 189
REMOTE_SYSTEM_SR 189
REPORT_TITLE 189
SPOOL_FLAG 191
Support for IBM Machine Code Characters 191
Using the Send Job Command 192
Parameters 192
LOCAL_FILE_SJ 193
REMOTE_SYSTEM_SJ 193
How to Send a Metatransfer 193
Parameters 194
How to Inquire on the Status of Metatransfers 196
Parameters 196
Return Codes 198
How to Retrieve History Records 199
Parameters 199
History Record Filter Parameters 200
Command Examples for Metatransfers and Inquire on Status 219
How to Send a Ping Request 220
Parameters 220
Command Example of Ping Request 223
Tracing Problems 224
Return Codes 225
Parameter 225
XTRACE 226
Setting Up Log Files 226
Parameter 226
XLOGFILE 227
Setting File Type Conversion 228
Parameters 228
CARRIAGE_FLAG 229
CODE_FLAG 230
Creating Custom Character Sets for File Conversion 230
When to Use Character Conversion 231
Procedure 231

Hexadecimal Numbers in Conversion Tables 232
Customizing Default Character Sets 232
Procedure 233
Specifying a Custom Character Set 234
Unicode and Multi-Byte Character Sets Support for Data Transfer 234
Using Unicode Transfer 235
Specifying Charset 235
Handling Conversion Errors 236
Record Processing 237
Examples 237
Run Reports on History Records 238
Generate a Transfer Information Report 239
Generate a Statistical Data Transfer Report 240
Using Store and Forward 242
About Store and Forward 242
When to Use 243
Parameter 243
Setting Password and User ID Security 243
Encrypting Parameter Values in Existing Configuration Files 245
Changing an Encrypted Value 247
Queuing Locally Initiated Transfers 247
Notifying of Transfer Completion 250
Checkpointing and Restarting Transfers 255

Chapter 4: Application Programming Interface 259

XcomAPI 259
XcomQAPI 260
Utility Functions 260
msgstring 260
XsetSystemPriv 260
Using API Member Names 260
Member Names 261
allocation_type 261
avgrec 262
blksize 262
carriage_control_characters 263
carriage_flag 264
checkpoint_count 264
class 265
code_flag 265
codetabl 266

compress 267
compress_pds 268
configssl 268
convert_classes 269
copies 269
createdelete 270
dataclas 271
debug_flag 271
den 271
destination 272
disposition 272
domain 272
dsntype 273
ear_cipher 274
ear_digest 276
ear_hash 276
ear_key 277
eattr 277
eol_classes 278
expdt 278
fips_mode 278
fcb 278
file_option 279
file_type 279
form 280
gatewayguid 280
hold 280
hold_transfer 281
label 281
labelnum 281
lclntfyl 282
lear_cipher 282
lear_digest 284
lear_hash 284
lear_key 285
lgatewayguid 285
local_charset 286
local_delim 286
local_file 289
local_file_rf 290
local_file_sj 290
local_file_sr 290

local_notify....291
lrecl....291
maxreclen....291
mbcs_converror....292
mbcs_inputerror....293
metacode_classes....293
mgmtclas....293
notify_name....294
notifyl....294
notifyr....295
notify_term....295
number_of_retries....296
num_of_dir_blocks....296
password....296
primary_alloc....296
print_class....297
port....297
protocol....297
queue....298
rmtntfyl....298
record_format....298
remote_file....300
remote_file_rf....301
remote_system....301
remote_charset....302
remote_delim....303
remove_trail_blanks....305
report_title....306
restart_supported....306
retpd....306
secondary_alloc....307
secure_socket....307
spool_flag....307
start_date....307
start_time....308
stat_frequency....308
storclas....308
tape....308
tempdir....309
trnencrl_cipher....309
truncation....311
trusted....311

unit 312
unitct 312
userid 312
version 312
volct 313
volsq 313
volume 313
xcomfullssl 313
xcom_show_cipher 314
xidest 314
xlogfile 314
xluname 315
xmode 315
xnodespec 315
xtcerrdecr 316
xtcerrincr 316
xtcerrpurge 316
xtcerrrel 317
xtcgooddecr 317
xtcgoodincr 317
xtcgoodpurge 318
xtcgoodrel 318
xtchold_count 318
xtcjob 318
xtcnet 319
xtrace 319
Upgrade from Previous Releases 320
Changes to Access Permissions 320
Link Libraries 321
Starting States 321
Starting States in startst.h 321
API Control Block Structure 322
Using XcomQAPI Option Parameters 326
Option Parameters 326
funccode 327
nMax_Queue_Entries 328
queue_open_once 328
tid 328
trace_level 328
trace_filename 328
API Examples 329
C Language API File Transfer to Remote System File Example 330

Queue Request Using XcomQAPI Routine 334
XcomQAPI Return Codes 339

Chapter 5: Generating SSL Certificates 341

Using SSL Mode 341
Setting Expiration 342
Creating the CA Certificate 343
Creating the Server Certificate 344
Creating the Client Certificate 344
Configuring the CA XCOM Data Transport SSL Server 345
Configuring the CA XCOM Data Transport Client 346
Default Sample Scripts 346
Cipher Suites 347
Supported Cryptographic Protocols 348
Supported Cipher Suites 348
Cipher Suites Supported in TLSV1 when FIPS_MODE=YES 349
Cipher Suites Supported in TLSV1 when FIPS_MODE=NO 349
Cipher Suites Supported in v3 350

Appendix A: Parameters 351

Using Parameters 351
Parameter Format 352
Syntax 352
Guidelines 353
List of Parameters 353
AGE 354
AGE_TIME 354
ALLOCATION_TYPE 355
ATOE_FILENAME 355
AUTH_TYPE 355
AVGREC 356
BLKSIZE 356
CACHE_READ_SZ 357
CACHE_WRITE_SZ 357
CAPKIHOME 358
CARRIAGE_CONTROL_CHARACTERS 359
CARRIAGE_FLAG 360
CHARS 360
CHECKPOINT_COUNT 361
CKPT 361
CLASS 362

CODE 363
CODE_FLAG 364
CODETABL 365
COMPRESS 365
COMPRESS_PDS 366
CONTROL 367
CONVERT_CLASSES 367
COPIES 367
CREATE_DIRECTORIES 367
CREATEDELETE 368
DATACLAS 369
DEBUG_FLAG 369
DEFAULT_CHARSET 369
DEFAULT_CONVERROR 370
DEFAULT_DELIM 370
DEFAULT_INPUTERROR 372
DEN 373
DESTINATION 373
DESTINATION_TYPE 373
DISPOSITION 374
DOMAIN 374
DROPSESS 374
DSNTYPE 375
EAR_CIPHER 376
EAR_DIGEST 377
EAR_HASH 378
EAR_KEY 378
EATTR 379
EOL_CLASSES 379
EPRTY 379
ETOA_FILENAME 380
EXPDT 380
EXPIRATION_TIME 380
FCB 381
FILE_OPTION 382
FILE_OPTION_RF 383
FILEDATA 384
FIPS_MODE 384
FORM 385
GATEWAY_IP 385
GATEWAY_PORT 385
GATEWAY_VERSION 386

GATEWAYCERT 386
GATEWAYCPASS 386
GATEWAYGUID 387
GATEWAYPKEY 387
GATEWAYPROTOCOL 388
HIST_FILE 388
HOLD_SESS 389
HOLD_TRANSFER 389
HOLD 389
HOLDCOUNT 390
HOLDFLAG 390
IDENT 390
INQ_FILE 391
INQ_WAIT 391
INTERNAL_CONVERSION_TABLES 391
IPNAME 392
IPPORT 392
LABEL 392
LABELNUM 393
LCLNTFYL 393
LDOMAIN 393
LEAR_CIPHER 394
LEAR_DIGEST 395
LEAR_HASH 396
LEAR_KEY 396
LFILE 397
LGATEWAYGUID 397
LOCAL_CHARSET 397
LOCAL_DELIM 398
LOCAL_FILE 400
LOCAL_FILE_RF 401
LOCAL_FILE_SJ 401
LOCAL_FILE_SR 402
LOCAL_NOTIFY 402
LOG_CONNECT_MSG 403
LPASS 403
LPASSWORD 403
LRECL 404
LUSER 404
LUSERID 404
MAIL_TYPE 405
MAX_QUEUE_ENTRIES 405

MAX_REMOTE_TCP 405
MAX_SESSIONS_ENTRIES 406
MAXRECLEN 406
MBCS_CONVERROR 407
MBCS_INPUTERROR 408
METACODE_CLASSES 408
MGMTCLAS 408
NOTIFY_NAME 409
NOTIFY_TERM 409
NOTIFYL 410
NOTIFYR 411
NUM_OF_DIR_BLOCKS 411
NUMBER_OF_RETRIES 412
OEDATE 412
OETIME 413
OFILE 414
OFILECASE 415
OFILETYPE 415
OFLMAX 416
OFLMIN 417
OID 418
OINIT 418
OJOB 419
OLMSG 420
OLOCATN 421
OLU 421
OREQ 422
OSDATE 423
OSTIME 424
OSYSID 424
OSYSNAME 425
OTNAME 425
OTYPE 426
OTYPEREQ 426
OUSER 427
OVOL 427
PASSWORD 428
PORT 428
PRIMARY_ALLOC 428
PRIORITY 428
PROGLIB 429
PROTOCOL 429

QUEUE 430
QUEUE_PATH 430
RECORD_FORMAT 430
RELEASE 432
REMOTE_CHARSET 433
REMOTE_DELIM 434
REMOTE_FILE 437
REMOTE_FILE_RF 438
REMOTE_SYSTEM 438
REMOTE_SYSTEM_RF 439
REMOTE_SYSTEM_SJ 439
REMOTE_SYSTEM_SR 440
REMOVE_TRAIL_BLANKS 440
REPORT_TITLE 440
RESTART_SUPPORTED 441
RETPD 442
RETRY_TIME 442
RMTNTFYL 442
RNOTIFY 443
RNOTIFYNAME 444
SAVE_PASSWORD_IN_CNF 445
SECONDARY_ALLOC 445
SECURE_SCHEDULE 445
SECURE_SOCKET 446
SESSIONS_FILE 446
SHELL_CMD 446
SMTP_SERVER 446
SOCK_DELAY 447
SOCK_RCV_BUF_SIZE 447
SOCK_SEND_BUF_SIZE 447
SPACE 448
SPOOL_FLAG 448
SPRTY 449
START_DATE 449
START_TIME 449
STARTDATE 450
STARTTIME 451
STAT_FREQUENCY 451
STCIP 451
STCPORT 452
STCTRNENCRL_CIPHER 452
STORCLAS 454

SYSID 454
SYSNAME 455
SYSTEM_USER_DATA 455
SYSUDATA 455
TAPE 456
TCP_CMD 456
TDUDATA 456
TEMPDIR 456
TP_CMD 457
TRACE_PATH 457
TRANSFER_NAME 457
TRANSFER_TYPE 458
TRANSFER_USER_DATA 458
TRNENCRL_CIPHER 459
TRNENCRR_CIPHER 461
TRNENCRR_DHBITS 463
TRUNCATION 464
TRUSTED 464
TRUST_DATABASE_NAME 465
TRUST_DATABASE_TYPE 465
TRUST_ODBC 465
TRUST_OWNER 465
TRUST_PASS 466
TRUST_PORT 466
TRUST_SERVER 466
TRUST_TABLE_PFX 466
TRUST_USER 467
TXPI_BUF_SIZE 467
TXPI_RECEIVE_TIMEOUT 467
TXPI_SEND_CHECK_FREQ 467
TXPI_TERM_TIMEOUT 468
UMASK 468
UNIT 468
UNIT_RF 469
UNITCT 469
USE_TP_SECURITY 469
USERID 469
USEROVR 470
VERSION 470
VOLCT 470
VOLSQ 470
VOLUME 471

VOLUME_RF 471
WRITER 471
XBUFFSIZE 471
XCOM_CONFIG_SSL 472
XCOM_ICUPATH 472
XCOM_JVM 472
XCOM_MYSQLODBC 472
XCOM_ODBC 473
XCOM_SHLIBPATH 473
XCOM_PASSWORD 473
XCOM_SHOW_CIPHER 473
XCOM_TRUSTED_OVR 474
XCOM_USERID 474
XCOMPRE_LOCAL 474
XENDCMD 474
XIDEST 475
XLOGFILE 475
XLPCMD 475
XLUNAME 476
XMODE 476
XNODESPEC 476
XNOTIFYCMD 476
XPPCMD 477
XPRECMD 477
XTCERRDECR 477
XTCERRINCR 478
XTCERRPURGE 478
XTCERRREL 478
XTCGOODDECR 479
XTCGOODINCR 479
XTCGOODPURGE 479
XTCGOODREL 480
XTCJOB 480
XTCNET 480
XTRACE 481

Appendix B: Messages **483**

Error Messages 483
Message ID Content 484
List of Messages 485

Appendix C: Character Sets for Unicode Transfer 597

Unicode 597
LATIN_1 598
US_ASCII 599
Multibyte character sets (MBCS) 599
Single Byte Character Sets (SBCS) 602
Double-byte character sets (DBCS) 613
ISO_2022 613
HZ 614
Indian Script Code for Information Interchange (ISCII) 614
LMBCS_1 615
EBCDIC_STATEFUL 615

Appendix D: Problem Determination 617

Where to Find Information 617
General Procedures 618
Test Procedure 619
Problem Determination Worksheet 620
- General Information 620
- Environment Information 621
- Transfer Type 621
- Problem Description 622
- Problem History 623
- Error Messages 623
- Network Configuration Diagram 624
- Environmental Information Inventory 624
- Documentation 625

Contact Technical Support 627
Product Releases and Maintenance 627

Index 629

Chapter 1: Using CA XCOM Data Transport

This section contains the following topics:

CA XCOM Data Transport GUI (see page 25)
Start CA XCOM Data Transport (see page 26)
Access CA XCOM Data Transport (see page 26)
About CA XCOM Gateway (see page 29)
Using CA XCOM Data Transport with CA XCOM Gateway (see page 33)

CA XCOM Data Transport GUI

The CA XCOM Data Transport Graphical User Interface (GUI) provides access to all of the features of CA XCOM Data Transport. From the GUI you can do the following:

- Create and edit CA XCOM Data Transport transfer configuration files
- Initiate transfers
- Monitor the log
- Update the status of active and pending transfers
- Get the history of transfer records in the queue and/or the database
- Save the history of transfer records in formats to produce a CA Easytrieve report or to display in a spreadsheet
- Export the history filter parameter data, which can be used with batch history retrieval processing

Users in the UNIX Administrator Group can also do the following:

- Trusted transfer configuration
- Global parameter configuration

The CA XCOM Data Transport GUI is the front-end for CA XCOM Data Transport. You use this GUI to communicate with the CA XCOM Data Transport server. All requests, such as scheduling transfers and getting history records, are passed to the CA XCOM Data Transport server using the TCP/IP interface. Results obtained are returned to the GUI.

For more information about using the CA XCOM Data Transport GUI, see the *CA XCOM Data Transport Graphical User Interface Online Help*.

Start CA XCOM Data Transport

Before you can use CA XCOM Data Transport to perform transfers, the XCOMD CA XCOM Scheduler service must be running. For a procedure for starting the XCOMD CA XCOM Scheduler service, see the *CA XCOM Data Transport for UNIX and Linux Installation Guide*.

To start the CA XCOM Data Transport GUI, run the following command:

```
$XCOM_HOME/bin/StandaloneUI.sh
```

Note: This is the full path name specification for StandaloneUI.sh. Make sure that the XCOM_HOME environment variable is set to the XCOM installed location.

Note: For starting and stopping the CA XCOM Data Transport GUI, ERRORLEVEL has no meaning. When executing the command-line programs (xcomtcp.exe and xcom62.exe), the ERRORLEVEL indicates successful or failed processing. An ERRORLEVEL of ZERO indicates successful operation and termination. An ERRORLEVEL of NON_ZERO values indicates program processing was not successful, and that the program has terminated.

Access CA XCOM Data Transport

Use the CA XCOM Data Transport GUI to access all the functions of CA XCOM Data Transport.

Start from the Home Page

The GUI home page is the beginning point for all transfers.

From the home page you can access other pages to configure parameters for the four CA XCOM Data Transport transfer types:

- Send File
- Send Job
- Send Report
- Receive File

From the home page of the GUI, you can access the following pages and functions:

- Schedule Transfer
- Get History Records
- Trusted Transfer (superuser (root) only)
- Global Parameters (superuser (root) only)

The functions and options of this window can be selected from the menu bar or from the buttons on the toolbar at the top of the window.

For more information about using the CA XCOM Data Transport GUI, see the *CA XCOM Data Transport Graphical User Interface Online Help*.

Schedule Transfer Page

The Schedule Transfer page allows you to build transfer records and then process them in either of the following ways:

- By submitting the scheduled transfers
- By saving them to a configuration XML file

From this page, you can also use all of the following functions for transfers:

- Add a new transfer
- Edit a transfer
- Delete a transfer
- Copy or duplicate a transfer
- Load transfers from existing configuration files
- Clear existing loaded transfers to create new transfers
- Select a transfer, select all transfers, or clear all selected transfers
- Save a transfer or save all transfers
- Submit a transfer or submit all transfers

Get History Records Page

The Get History Records page allows you to extract and display the history record information for the CA XCOM Data Transport server. You can also limit the number of records displayed by using different search criteria.

The history record information can be extracted and displayed from the following places:

- From the CA XCOM Data Transport server, depending on the history database setup at your site
- From the queue

Trusted Transfer Page

The Trusted Transfer page allows the superuser to add and remove users, groups, or remote systems to the Trusted Transfer database. You can also update remote systems only.

Global Parameters Page

The Global Parameters page allows the superuser to modify global parameters in the xcom global (xcom.glb) file.

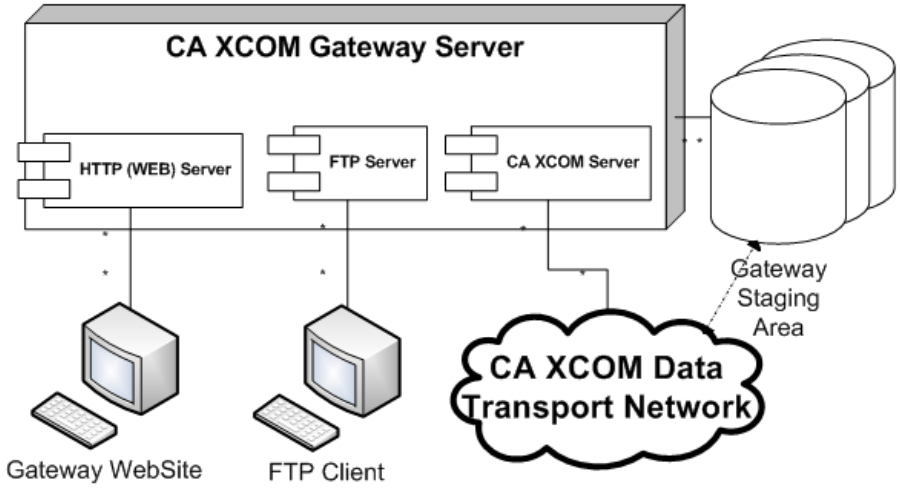
About CA XCOM Gateway

CA XCOM Gateway is a platform-independent component to enable users to upload or download files from a CA XCOM Data Transport network by using devices that are not equipped with any CA XCOM Data Transport software components.

Note: These sections apply only to those sites that have CA XCOM Gateway installed.

Associate CA XCOM Data Transport with CA XCOM Gateway

Before you can insert or extract the files from CA XCOM Gateway using CA XCOM Data Transport, your CA XCOM Data Transport system needs to be associated with the CA XCOM Gateway server.



You need to set the following global parameters to associate your CA XCOM Data Transport system with CA XCOM Gateway.

GATEWAY_IP

Indicates the CA XCOM Gateway domain (or IP address) to which this instance of the CA XCOM Data Transport server is associated.

Range: 1 to 64 characters

Default: None

GATEWAY_PORT

Indicates the CA XCOM Gateway port to which this instance of the CA XCOM Data Transport server is associated.

Range: 1 to 65535

Default: None

GATEWAY_VERSION

Indicates the CA XCOM Gateway version to which this instance of the CA XCOM Data Transport server is allied.

For CA XCOM Gateway transfers only.

R115

Indicates that the CA XCOM Data Transport server is allied to the R11.5 Gateway

R116

Indicates that the CA XCOM Data Transport server is allied to the R11.6 Gateway

Default: R115

CA XCOM Data Transport Client Authentication

The CA XCOM Data Transport server can extract files from and insert files into CA XCOM Gateway. To ensure data security, it is important to prevent software, other than your own CA XCOM Data Transport servers, from connecting to CA XCOM Gateway.

For more information, see the chapter “Set Up CA XCOM Data Transport Client Authentication” in the *CA XCOM Data Transport Gateway Product Guide*.

You need to set the following parameters to allow CA XCOM Gateway to authenticate this CA XCOM Data Transport server.

GATEWAYCERT

Specifies the file name for the PEM encoded certificate that will be passed to CA XCOM Gateway to confirm that the request is from an authorized CA XCOM Data Transport server.

Range: 0 to 256 characters

Default: None

GATEWAYPKEY

Specifies the file name for the PEM encoded certificate that includes the private key corresponding to the public key in GATEWAYCERT.

Range: 0 to 256 characters

Default: None

GATEWAYCPASS

Specifies the passphrase if the GATEWAYPKEY private key has been encoded.

Range: 0 to 32 characters

Default: None

Set Other Optional Parameters

CA XCOM Gateway policies can be configured to specify encryption or hashing of stored files. To perform these cryptographic functions, you need to set the following parameters.

CAPKIHOME

Specifies the CAPKI/ETPKI library path. Set the CAPKI/ETPKI library path before using any of the following items:

- Password encryption transfers with cipher negotiation
- Encryption at rest transfers
- SSL transfers
- Transfers using CA XCOM Gateway

The CA XCOM Data Transport installer installs the CAPKI/ETPKI library. Typically it is installed to the CA common components directory. A default path is populated in XCOM.glb during installation. If CAPKI is installed at a different location, update CAPKIHOME with the correct path where libcapki.so resides.

Range: 0 to 256 characters

Default: The default depends on the platform, as follows:

- For AIX—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/AIX/powerpc/32/lib
- For HP-UX PA RISC—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/HP-UX/pa-risc/11.11_32_pa2.0/lib
- For HP-UX IA64— CAPKIHOME= /opt/CA/SharedComponents/CAPKI/HP-UX/Itanium/64/lib
- For Linux—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/Linux/x86/32/lib
- For Solaris—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/SunOS/sparc/32/lib

FIPS_MODE

Specify the mode to enable or disable FIPS-compliant cipher and digest algorithms to be used for SSL transfers or for encryption at rest.

YES

Indicates that encryption/decryption will be running in FIPS mode.

NO

Indicates that encryption/decryption will not be running in FIPS mode.

Default: NO

CA XCOM Gateway Staging Area Access

The CA XCOM Gateway staging area is a physical storage area on disk that is used to hold files that have been imported into CA XCOM Gateway. Both the CA XCOM Gateway server and its companion CA XCOM Data Transport server require access to the staging area. So it must either be located on the same server or be accessible from that server as a network drive.

Using CA XCOM Data Transport with CA XCOM Gateway

CA XCOM Gateway is a platform-independent component to enable users to upload or download files from a CA XCOM Data Transport network by using devices that are not equipped with any CA XCOM Data Transport software components.

CA XCOM Gateway Transfers

When CA XCOM Data Transport needs to transfer files into or out of CA XCOM Gateway, the local CA XCOM Data Transport server connects to the CA XCOM Gateway server and uses web service requests to request access to CA XCOM Gateway's file storage area and to access existing CA XCOM Gateway files or import new ones.

The CA XCOM Data Transport server identifies a particular transfer as a CA XCOM Gateway transfer as follows:

- GATEWAYGUID/LGATEWAYGUID is non-null.
- This instance of CA XCOM Data Transport server is configured as a CA XCOM Gateway allied server.
- Format of the CA XCOM Gateway file name.

GATEWAYGUID

Identifies the remote file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. This is a unique value that identifies each CA XCOM Gateway file.

Note: When the CA XCOM Gateway GUID is not known, the keyword ANY can be used to identify the remote file as a CA XCOM Gateway file.

Range: 0 to 36 characters

Default: None

GATEWAYPROTOCOL

Identifies the protocol that the Gateway uses to access the external server

This parameter must be specified as 'FTP' or 'SFTP' or 'FTPS' for Transfers between associated Gateway and External Server. In all other cases it will be XCOM by default.

FTP

For transfers between Gateway and registered FTP Server

SFTP

For transfers between Gateway and the registered SFTP Server

FTPS

For transfers between Gateway and the registered FTPS Server

XCOM

Default used for Transfers between Gateway and the XCOM server

Default: XCOM

Note:

- This parameter will be used only when requesting for Extract file from Gateway Environment to External Server or Insert file from External Server to Gateway Environment.
- This can be used only if the associated Gateway version is R11.6
- This parameter is not valid for SEND JOB and SEND REPORT actions.

LGATEWAYGUID

Identifies the local file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. The CA XCOM Gateway GUID is a unique value that identifies each CA XCOM Gateway file. The keyword ANY can be used to identify the local file as a CA XCOM Gateway file when the CA XCOM Gateway GUID is not known.

Range: 0 to 36 characters

Default: None

Format of CA XCOM Gateway File Names

You need to provide the CA XCOM Gateway file name in a specific format. The file name should contain one and only one level of path and no leading drive specification, as follows:

```
[/\]policyname[/\]filename
```

Examples:

```
\payroll\employeelist.doc

/payroll/employeelist.doc

/payroll\employeelist.doc

\payroll/employeelist.doc
```

Note: In these examples, payroll is a user-defined CA XCOM Gateway policy. For more information about CA XCOM Gateway policies, see the *CA XCOM Data Transport Gateway Product Guide*.

If CA XCOM Data Transport identifies the transfer as a non-Gateway transfer, it then continues with the conventional file transfer unrelated to Gateway.

Set Up User Permissions

Users need to have access to the CA XCOM Gateway policy to which the file will be uploaded. Specify the appropriate user permissions for the policy.

For more information about permissions and policies, see the *CA XCOM Data Transport Gateway Product Guide*.

The rest of the transfer details can be provided as usual. CA XCOM Data Transport, associated with CA XCOM Gateway, makes web services calls to get the details of policy and the actual path of the staging area on which the file will be stored.

The transmission of data occurs the same as with non-Gateway transfers. This allows the use of standard CA XCOM Data Transport features such as SSL, compression, and checkpoint restarts.

Insert Files into CA XCOM Gateway

A CA XCOM Data Transport system associated with CA XCOM Gateway can insert files into the CA XCOM Gateway staging area.

Note: An incoming (received) file may be either a locally initiated RECEIVE FILE or a remotely initiated SEND FILE.

Set Up a CA XCOM Gateway File Name

For a remotely initiated SEND FILE transfer to the CA XCOM Data Transport system associated with CA XCOM Gateway, set the CA XCOM Gateway file name to REMOTE_FILE, and set the GATEWAYGUID to any non-null value.

For a locally initiated RECEIVE FILE transfer on the CA XCOM Data Transport system associated with CA XCOM Gateway, set the CA XCOM Gateway file name to LOCAL_FILE_RF, and set the LGATEWAYGUID to any non-null value.

For an incoming file, CA XCOM Gateway allocates a new CA XCOM Gateway GUID, so the specified value by (L)GATEWAYGUID is disregarded. Typically it is set to ANY.

Extract Files from CA XCOM Gateway to External Server

A CA XCOM Data Transport system associated with CA XCOM Gateway can request to extract files from the CA XCOM Gateway staging area to External Server registered with Gateway.

Note:

- This operation can be performed only if the associated Gateway version is R11.6.
- XCOM will only make a web service call to request extraction to external server and actual transfer will be carried between Gateway and External Server. So, these transfers are not managed by XCOM and the features like Check point restart or Suspend/Resume or Encryption at Rest will not be available for these transfers.

Set Up a Configuration file

This operation will always be locally initiated SEND FILE. Local File will be specified in the below format.

```
[/\]policyname[/\]filename
```

Whenever CA XCOM Gateway receives a file, it generates a new GUID for each CA XCOM Gateway file. When extracting a CA XCOM Gateway file, we need to provide the GUID of the file in LGATEWAYGUID.

Set the CA XCOM Gateway file name to LOCAL_FILE, and set the LGATEWAYGUID to the CA XCOM Gateway GUID value. If you want to extract the latest version of the file, then set the GUID to ANY.

External Server name (name used to register server in Gateway) to which the file will be exported should be provided in REMOTE_SYSTEM.

If the External Server credentials are registered with Gateway and the you want to use the same credentials to access the External Server, then USERID can be set to Symbolic Variable '&GXSUSER'. Any password supplied in PASSWORD will be ignored here.

If the External Server credentials are not registered with Gateway, then USERID and PASSWORD parameters can be used to provide the External Server credentials.

Insert Files from External Server to CA XCOM Gateway

A CA XCOM Data Transport system associated with CA XCOM Gateway can request to insert files from External Server registered with Gateway to the CA XCOM Gateway staging area.

Note:

- This operation can be performed only if the associated Gateway version is R11.6.
- XCOM will only make a web service call to request insertion of file from external server and actual transfer will be carried between Gateway and External Server. So, these transfers are not managed by XCOM and the features like Check point restart or Suspend/Resume or Encryption at Rest will not be available for these transfers.

Set Up a Configuration file

This operation will always be locally initiated RECEIVE FILE. Local File will be specified in the below format.

```
[/\]policyname[/\]filename
```

Set the CA XCOM Gateway file name to LOCAL_FILE_RF, and set the LGATEWAYGUID to any non-null value

For an incoming file, CA XCOM Gateway allocates a new CA XCOM Gateway GUID, so the specified value by (L)GATEWAYGUID is disregarded. Typically it is set to ANY.

External Server name (name used to register server in Gateway) from which the file will be imported should be provided in REMOTE_SYSTEM_RF

If the External Server credentials are registered with Gateway and the you want to use the same credentials to access the External Server, then USERID can be set to Symbolic Variable '&GXSUSER'. Any password supplied in PASSWORD will be ignored here.

If the External Server credentials are not registered with Gateway, then USERID and PASSWORD parameters can be used to provide the External Server credentials.

Chapter 2: Operating Environment

The components used for scheduling transfers, handling files, and managing CA XCOM Data Transport resources are the xcomd command and the daemon process, the global parameters, the session control parameters, the queue, and the use of post processing scripts.

A substantial portion of a user's ability to control CA XCOM Data Transport comes from the use of files to set variables that govern the system's behavior. These files are xcom.glb and xcom.cnf.

This section contains the following topics:

CA XCOM Data Transport Globals (see page 39)
About Parameters (see page 40)
The xcomd Command (see page 43)
Global Parameters (see page 45)
Parameters (see page 45)
Local Session Control Parameters (see page 80)
Manage the Queue (see page 81)
The xcomqm Command (see page 81)
Purge the Log File (see page 84)
About the xcompre Pre-allocation Exit (see page 85)
About Post Processing Scripts (see page 88)
How to Modify Post Processing Parameters (see page 88)
How to Modify the Post Transfer Script (see page 91)
How to Modify Printing Processes (see page 95)
How to Customize the Notification Process (see page 97)
Pluggable Authentication Modules (PAM) Based Authentication (see page 103)
Trusted Access Database (see page 104)
Reports Containing Machine Code Characters (see page 106)
About History Records (see page 107)
Configuring JDBC Database Access (see page 123)
Security Considerations for History Records (see page 124)
Instructions to Upgrade Existing R11.5 History Database (see page 125)
Interdependent Transfers (see page 131)
Coding Interdependent Transfers (see page 132)
Password Protection by Encryption (see page 133)
Transfer Control (XTC) Parameters (see page 135)

CA XCOM Data Transport Globals

You should be aware of the following information before you begin using CA XCOM Data Transport.

PATH and Directory Considerations

- Be sure that you have properly set up your directories and path variables before executing a file transfer request.
- The CA XCOM Data Transport daemon, xcomd, is installed in the $XCOM_HOME/sbin directory.
- Be sure that $XCOM_HOME/bin is in your PATH variable and is set so that your shell can find xcom62 or xcomtcp, or both.
- The $XCOM_HOME directory is the default repository that CA XCOM Data Transport uses for logging and for configuration information.
- For further information about PATH and directory defaults, see the appendix Parameters.

Transfer Protocols

CA XCOM Data Transport supports transfer protocols, such as Ethernet, SDLC, Token Ring, and TCP/IP.

SNA/APPC Protocols

For SNA/APPC protocols, support is based on the capabilities provided by the particular APPC vendor.

For SNA/APPC, CA XCOM Data Transport requires an active LU 6.2 session to transfer files. When an LU 6.2 session is established, local users can begin using CA XCOM Data Transport to initiate transfers.

TCP/IP Protocols

The TCP/IP component of CA XCOM Data Transport supports TCP/IP protocols for performing transfers between CA XCOM Data Transport r11.x platforms using TCP/IP.

About Parameters

Use parameter values to control the variables that govern the behavior of CA XCOM Data Transport. For a full list of CA XCOM Data Transport for UNIX and Linux parameters, see the appendix "Parameters."

Parameter Values

Parameter values can be defined to CA XCOM Data Transport as indicated in the table below. The values take precedence depending on where they are specified.

The order in which parameter values take precedence is as follows:

Order of Precedence	Parameter Value Specified in	Explanation
1	Command Line	If the user specifies a value for a parameter on the command line, this overrides every other specification in the files or the program.
2	xcom.cnf or *filename.cnf* or *filename.xml*	If the value is specified in the xcom.cnf file, or in a user-customized configuration file, *filename.cnf* or *filename.xml*, it overrides the value in the xcom.glb file for locally initiated transfers.
3	xcom.glb	If the value is specified in the xcom.glb file, it overrides the value in the program. The values specified in xcom.glb are generally used by the system administrator to start CA XCOM Data Transport and for remotely initiated transfers.
4	Program Defaults	If a value is not specified anywhere, the program has its own defaults.

Parameter Format

The CA XCOM Data Transport parameters consist of assignment statements. The format for assignment statements is as follows

- PARAMETER_NAME (always all uppercase, with underscore character (_) when indicated)
- An equal sign (=)
- A character string terminated by a newline

Syntax

The syntax for assignment statements is as follows:

```
PARAMETER_NAME=value
```

Example

In the following example, the parameter EXPIRATION_TIME is set to a value of 6000 seconds.

```
EXPIRATION_TIME=6000
```

This controls the maximum time in seconds that a transaction is held in the transfer queue.

Guidelines

The following guidelines apply when using CA XCOM Data Transport parameters:

- When you type trailing spaces and tabs from the command line as part of a parameter value, they are stripped by the command line processor (the shell).
- When you type trailing spaces and tabs into a file or script with an editor, such as vi, they are treated as part of a parameter value. These trailing spaces and tabs should be removed. If they are not removed they can cause confusion and unpredictable or undesirable results.
- Empty lines and lines beginning with a pound sign (#) are discarded.
- When the parameter value includes a special character, it is interpreted as a command by the command processor (the shell) than as a literal value. Use the appropriate escape sequence to specify the literal meaning.

Examples

In the following examples, the *spacespacespace* represents trailing spaces.

If you type the following at the command line, the trailing spaces are ignored:

```
xcom62 -c1 LOCAL_FILE=xyzspacespacespace
```

If you type the following into any configuration file using an editor, the *spacespacespace* would be treated as part of the name of the LOCAL_FILE:

```
LOCAL_FILE=xyzspacespacespace
```

If you type the following at the command line, the “!” character will be interpreted as history expansion command with the bash shell.

TRNENCRL_CIPHER=ALL:!AES

The xcomd Command

Use xcomd from the command line to start and stop the CA XCOM Data Transport daemon process, to kill the daemon process, to set trace levels, and to report the release level of xcomd.

The xcomd command controls the daemon. The daemon itself runs as a background process to control file transfers and manage CA XCOM Data Transport resources. The daemon does the following:

- Schedules and synchronizes transfer requests.
- Controls shared memory for transfers.
- Establishes the default parameter values by reading the parameter file, xcom.glb and then running in the background.
- Controls the automatic restart of locally initiated transfers.
- Writes queue information out to disk periodically.
- Deletes aged entries from the queue.
- Notifies a local user by executing the xcomntfy script when LOCAL_NOTIFY is required.
- Communicates with active or pending transfers to terminate a transfer.

Control the Daemon

When CA XCOM Data Transport is installed, the superuser (root) has permission to control the daemon. Without authorized permission, other users cannot control the daemon. Users are normally given permission to perform transfers and check the status of the queue.

The daemon must be running before you can do any transfers. For performing transfers using SNA protocols, CA XCOM Data Transport also requires an active LU 6.2 session to transfer files. Once an LU 6.2 session is established, local users can begin using CA XCOM Data Transport to initiate transfers. For performing transfers using TCP/IP, your system must be configured for using CA XCOM Data Transport with TCP/IP.

Syntax

The syntax for using the xcomd command is as follows:

`xcomd` *`option`*

Options

The following table explains the options for xcomd:

-c[y]

Kill the daemon process and free shared memory. This option will stop the daemon while transfers are running.

Note: Use this option as a last resort because it brings the scheduler down immediately, whether or not there are running or scheduled transfers. If you use the **-c** option without the **y**, the system prompts you to confirm this action by typing **y** for yes.

Important! This option can cause corruption of the CA XCOM queue. It can shut down CA XCOM Data Transport while it is updating the queue. To prevent problems, delete the queue by deleting all files in $XCOM_HOME/Q after issuing this command.

-d*tracelevel*

Set the trace level of the daemon process to *tracelevel*. Higher numbers (up to 10) give more trace information. The trace goes to stderr. For more information, see the topic, Tracing Problems, in the chapter "The Command Line Interface."

-r

Report the release level of xcomd and exit.

-s

Stop the CA XCOM Data Transport daemon by sending a request to the scheduler.

Note: This option does not stop the daemon immediately if there is an active transfer going on.

Start xcomd

To start xcomd, enter the following command at the system prompt:

```
$XCOM_HOME/sbin/xcomd
```

Note: Ensure the XCOM_HOME environment variable is set to the CA XCOM Data Transport install directory that is selected during installation.

The options for xcomd are described in the previous Options topic.

Global Parameters

The xcom.glb file lets you set defaults independently of the execution environment for transfers. This file is created as part of the installation process. Use vi or another editor to modify the parameters.

The parameters in xcom.glb are a superset of the parameters in xcom.cnf. Any parameters set in xcom.cnf can be set in xcom.glb. However, the parameters that are specific to xcom.glb cannot be set in xcom.cnf. If there are any changes made to the xcom.glb file, the CA XCOM Data Transport daemon must be stopped and restarted.

Note: The xcom.glb parameters apply to incoming and outgoing transfers unless they are overridden. The xcom.cnf parameters override those in xcom.glb for outgoing transfers. The parameters in xcom.glb are used for remotely initiated transfers.

For a full list of CA XCOM Data Transport parameters, see the appendix "Parameters." For an explanation of parameters and defaults, see Using CA XCOM Data Transport Parameters in this chapter. This section contains information about using the global parameters contained in the xcom.glb file.

Parameters

The parameters for xcom.glb are shown below.

AGE_TIME

The number of seconds before waiting queue entries are removed from the queue. If the value is 0, the waiting queue entries never age and are never removed from the queue.

Range: 0 to 86313600 (999 days)

Default: 432000 (5 days)

ATOE_FILENAME

The name of the file containing the ASCII-to-EBCDIC character conversion table.

This file is a custom file that is used only for specifying custom translation tables from ASCII to EBCDIC, if needed.

Range: 0 to 256 characters

AUTH_TYPE

Specifies the type of authentication to be used for transfers.

PAM

Enables Pluggable Authentication Modules authentication.

SYSTEM

Enables traditional UNIX authentication.

Default: SYSTEM

CACHE_READ_SZ

This parameter specifies the size (in KB) of the cache for reading files.

***nnnn* (KB)**

The size of the cache for reading files.

Range: 0 – 9999

Default: 0

Note: In the XCOM.GLB file, a value of 0 to 1023 can be entered for this parameter without causing an error, but it is interpreted differently, as follows:

- A value of 0 is interpreted to use the old method of reading files.
- A value of 1 to 1023 is automatically converted to the default value of 1024.

CACHE_WRITE_SZ

Specifies the size (in KB) of the cache for writing files.

***nnnn* (KB)**

The size of the cache for writing files.

Range: 0 – 9999

Default: 0

Note: In the XCOM.GLB file, a value of 0 to 1023 can be entered for this parameter without causing an error, but it is interpreted differently, as follows:

- A value of 0 is interpreted to use the old method of reading files.
- A value of 1 to 1023 is automatically converted to the default value of 1024.

CAPKIHOME

Specifies the CAPKI/ETPKI library path. Set the CAPKI/ETPKI library path before using any of the following items:

- Password encryption transfers with cipher negotiation
- Encryption at rest transfers
- SSL transfers
- Transfers using CA XCOM Gateway

The CA XCOM Data Transport installer installs the CAPKI/ETPKI library. Typically it is installed to the CA common components directory. A default path is populated in XCOM.glb during installation. If CAPKI is installed at a different location, update CAPKIHOME with the correct path where libcapki.so resides.

Range: 0 to 256 characters

Default: The default depends on the platform, as follows:

- For AIX—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/AIX/powerpc/32/lib
- For HP-UX PA RISC—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/HP-UX/pa-risc/11.11_32_pa2.0/lib
- For HP-UX IA64— CAPKIHOME=/opt/CA/SharedComponents/CAPKI/HP-UX/Itanium/64/lib
- For Linux—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/Linux/x86/32/lib
- For Solaris—CAPKIHOME=/opt/CA/SharedComponents/CAPKI/SunOS/sparc/32/lib

CODETABL

This parameter specifies the prefix to the file names, atoe.tab and etoa.tab, that contain the external ASCII-to-EBCDIC and EBCDIC-to-ASCII custom character conversion tables. These custom character conversion tables determine which external translation tables will be used by the transfer.

This parameter is valid only if INTERNAL_CONVERSION_TABLES=NO.

Range: 0 to 3 alphanumeric characters

Default: None

CONVERT_CLASSES

A character string containing print classes for EBCDIC-to-ASCII conversions to be performed. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

CREATE_DIRECTORIES

This parameter indicates whether a new directory should be created to accommodate incoming file transfers. CREATE_DIRECTORIES=YES causes a new directory to be created.

Range: YES or NO

Default: YES

DATACLAS

This parameter specifies the name of the data class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

DEFAULT_CHARSET

DEFAULT_CHARSET specifies the default character set CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, the value can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

Default: ISO-8859-1

DEFAULT_CONVERROR

This parameter identifies the appropriate action when the input file contains characters that cannot be converted. They are not included within the output character sets character repertoire.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

REPLACE

Replace each unconvertible character with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each unconvertible character with the Unicode character that the decimal value nnnnnnn identifies. (If the specified replacement character cannot be represented in the output character set, then the transfer is failed).

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of skipped characters.

FAIL

The transfer terminates with an error condition.

Default: FAIL

DEFAULT_DELIM

DEFAULT_DELIM specifies an optional encoding for which the specified DEFAULT_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list. If LOCAL_CHARSET/REMOTE_CHARSET/DEFAULT_CHARSET is UTF based charset, UTF rules are applied disregarding encoding specified in DEFAULT_DELIM.

Additionally, specifies a colon separated list of record delimiters that are used to mark and detect the end of a record can be specified.

This parameter is used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

EBCDIC Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR

NL	
CRLF	CR
LFCR	LF
CRNL	CR
FF	

ASCII Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: ASCII:LF:CRLF

The tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which the *XXXX*_CHARSET parameter value is based, it should not be used (*XXXX*_CHARSET refers to the parameter that is in effect at the time of the transfer, i.e. LOCAL_CHARSET, REMOTE_CHARSET, or DEFAULT CHARSET).

Example: If *XXXX*_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if *XXXX*_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in DEFAULT_DELIM, delimiters will be derived as explained below.

If DEFAULT_DELIM is not specified or doesn't contain at least one valid delimiter, platform default values are used as shown below,

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

DEFAULT_INPUTERROR

This parameter identifies the appropriate action when the input file contains data that is not consistent with the specified input character set.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

REPLACE

Replace each piece of erroneous data with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each piece of erroneous data with the Unicode character that the decimal value nnnnnnn identifies.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of ignored bytes.

FAIL

The transfer terminates with an error condition.

Default: FAIL

DSNTYPE

This parameter specifies the data set definition.

Note: This parameter applies only to mainframe SMS data sets.

LIBRARY

Defines a PDSE.

PDS

Defines a partitioned data set.

BASIC

Defines a legacy sequential dataset.

LARGE

Defines a large format sequential dataset.

EXTREQ

Defines an extended format dataset.

EXTPREF

Specifies an extended format is preferred. If the extended format is not possible, a basic format is used.

<blank>

Defines a partitioned or sequential data set based on the data set characteristics entered.

Note: These values are IBM standards for SMS processing.

Range: One to eight characters

Default: None

EOL_CLASSES

A character string containing print classes for an ASCII newline that is appended to each record. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

ETOA_FILENAME

The name of the file containing the EBCDIC-to-ASCII character conversion table.

This file is a custom file that is used only for creating custom translation tables from EBCDIC to ASCII, if needed.

Range: 1 to 256 characters

EXPIRATION_TIME

The maximum time, in seconds, that a transaction is held in the transfer queue after completion. When the maximum time is reached, all references to the transaction are removed from the queue, including trace files and temporary files.

Note: If EXPIRATION_TIME is set to no value in xcom.glb, the program default of 6000 is used.

Range: 0 to 32767

Default: 6000

FIPS_MODE

Specify the mode to use for encryption/decryption.

Yes

Encryption/decryption will be running in FIPS mode.

No

Encryption/decryption will not be running in FIPS mode.

Default: No

GATEWAY_IP

Indicates the Gateway domain (or IP address) to which this instance of the CA XCOM Data Transport server is allied.

For Gateway transfers only.

Default: None

GATEWAY_PORT

Indicates the Gateway port to which this instance of the CA XCOM Data Transport server is allied.

For Gateway transfers only.

Default: None

GATEWAY_VERSION

Indicates the CA XCOM Gateway version to which this instance of the CA XCOM Data Transport server is allied.

For CA XCOM Gateway transfers only.

R115

Indicates that CA XCOM Data Transport server is allied to the R11.5 Gateway.

R116

Indicates that CA XCOM Data Transport server is allied to the R11.6 Gateway.

GATEWAYCERT

Specifies the file name for the certificate that will be passed to CA XCOM Gateway to confirm that the request is from an authorized CA XCOM Data Transport server.

For CA XCOM Gateway transfers only.

Range: 1 to 256 characters

Default: None

GATEWAYCPASS

Specifies the password for the CA XCOM Gateway certificates when they are created with a password. If the CA XCOM Gateway certificates were not created with a password, then this parameter is ignored.

For CA XCOM Gateway transfers only.

Range: 1 to 32 characters

Default: None

GATEWAYPKEY

Specifies the path and file name for the CA XCOM Gateway certificate's private key.

For CA XCOM Gateway transfers only.

Range: 1 to 256 characters

Default: None

INTERNAL_CONVERSION_TABLES

Indicates whether internal or external conversion tables should be used for ASCII-to-EBCDIC conversion and EBCDIC-to-ASCII conversion. The external conversion files may be customized to meet your needs. For the appropriate directory and file name, see ATOE_FILENAME and ETOA_FILENAME.

YES

Use internal conversion tables.

NO

Use the external tables contained in ATOE_FILENAME and ETOA_FILENAME.

Default: YES

LCLNTFYL

Specifies the local user notification level.

A or ALL

Notify on transfer completion.

W or WARN

Notify only if the transfer received a warning or error.

E or ERROR

Notify only if the transfer received an error.

Default: ALL

LICENSE_DIR

Specifies the location of the license file.

Range: 0 to 256 characters

Default: $CALIB

LOG_CONNECT_MSG

Specifies whether the CA XCOM Data Transport log file should include the following connection messages:

- XCOMU0785I
- XCOMU0786I
- XCOMU0793I
- XCOMU0805I
- XCOMU0811I
- XCOMU0812I
- XCOMU0813I
- XCOMU0814I
- XCOMU0818I

Range: YES or NO

Default: YES

MAIL_TYPE

Specifies the type of MAIL server used for sending mail notifications.

MAIL

UNIX MAIL server

SMTP

SMTP server

Default: None

MAX_QUEUE_ENTRIES

The maximum number of entries allowed in the transfer queue. Once the maximum number of queue entries is reached, subsequent transfer attempts are rejected by CA XCOM Data Transport.

Note: This value depends on the memory available when CA XCOM Data Transport is started.

Range: 0 to 32767

The value cannot be set larger than the maximum allowed shared memory segment divided by 512.

Default: 50

MAX_REMOTE_TCP

The maximum number of simultaneous remote TCP/IP transfers accepted by CA XCOM Data Transport. Transfers received after this limit has been reached are rejected.

Note: Setting this parameter to 0 prohibits any remote transfers from taking place.

Range: 0 to 999

Default: 32

MAX_SESSIONS_ENTRIES

The maximum number of partners that can be described in the XCOM.SES file.

Range: 1 to 999

Default: 50

METACODE_CLASSES

Classes of print jobs saved in metacode format, a variable length record format. For incoming report transfers only.

Range: 0 to 64 characters

Default: None

MGMTCLAS

Specifies the name of the management class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

PAM_PATH

Specifies the path to your PAM library for your current UNIX or Linux platform.

Note: This parameter is valid only if AUTH_TYPE=PAM.

PORT

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 1 to 65535

Default: 8044

PROTOCOL

The type of communication protocols to use.

SNA

For transfers using SNA/APPC communication protocols

TCPIP

For transfers using TCP/IP communication protocols

Default: SNA

Note: If only the TCP/IP component is installed with the base components of CA XCOM Data Transport, the default value is TCPIP.

QUEUE_PATH

Directory containing the transfer queue data files.

Range: 0 to 256 characters

Default: $XCOM_HOME/Q

Note: $XCOM_HOME is an environment variable.

REMOVE_TRAIL_BLANKS

Indicates whether or not to remove the blanks at the end of each record when receiving a text file from a remote partner.

Range: Yes or No

Default: No

SAVE_PASSWORD_IN_CNF

Determines if the xcomtool GUI saves the password in the .CNF file.

Range: The range of values is listed in the following table:

YES

The xcomtool GUI saves the password in the .CNF file in encrypted format.

NO

The xcomtool GUI does not save the password in the .CNF file and blanks the password field out after each operation.

Default: NO

SECURE_SOCKET

Specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and must connect to a SSL listener on the remote partner.

NO

Performs a non-secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

SESSIONS_FILE

The path name of the xcom.ses file. This file tells CA XCOM Data Transport the number of parallel sessions for locally initiated transfers for each remote system.

Range: 0 to 256 characters

Default: $XCOM_HOME/config/xcom.ses

Note: $XCOM_HOME is an environment variable

SHELL_CMD

Name of the command that runs jobs, reports, notification scripts, and post processing scripts on the local system.

Range: 1 to 256 characters

Default: /bin/sh

SMTP_SERVER

Specifies the name of the SMTP server.

Note: This is required only when MAIL_TYPE=SMTP.

Up to 64 bytes

The name of the SMTP server

Default: None

SOCK_DELAY

TCP/IP socket option TCP_NODELAY. Refers to the Nagle algorithm for send coalescing. By default, small sends may be delayed. Should have no impact for normal CA XCOM Data Transport record sizes. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: The range of values is listed in the following table:

YES

Small sends may be delayed. (Does not disable the Nagle algorithm.)

NO

All sends are immediate. (Disables the Nagle algorithm.)

Default: YES

SOCK_RCV_BUF_SIZE

TCP/IP Socket option SO_RCVBUF. The buffer size for receives. Use zero for the default size provided by the socket implementation. The value for SOCK_RCV_BUF_SIZE can be smaller than the value for TXPI_BUF_SIZE. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: 0 to 65536

Default: 0

SOCK_SEND_BUF_SIZE

TCP/IP Socket option SO_SNDBUF. The buffer size for sends. Use zero for the default size provided by the socket implementation. The value for SOCK_SEND_BUF_SIZE can be smaller than the value for TXPI_BUF_SIZE. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: 0 to 65536

Default: 0

STAT_FREQUENCY

Indicates the frequency with which transfer statistics are made available to xcomqm. Intended for tuning high-speed links. Longer values help performance, but byte/record counts in xcomqm -D may be slightly behind the actual counts.

Range: 1 to 9999 records

Default: 10

STORCLAS

Specifies the name of the storage class for a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

SYSID

This parameter specifies the system ID (one to four characters). SYSID and SYSNAME together provide a unique system identifier.

This value is set during the installation process and is used for Trusted Transfers and for getting history records.

Range: 1 to 4 characters

Default: None

SYSNAME

Specifies the system name (one to eight characters). SYSID and SYSNAME together provide a unique system identifier.

This value is set during the installation process and is used for Trusted Transfers and for getting history records.

Range: 1 to 8 characters

Default: None

TCP_CMD

This parameter specifies the path and name of the CA XCOM Data Transport program started by the XCOMD service that is used for queued locally initiated transfers, and for all remotely initiated transfers for TCP/IP protocols.

Range: 0 to 256

Default: $XCOM_HOME/bin/xcomtcp

Note: $XCOM_HOME is an environment variable.

TEMPDIR

Indicates the directory in which temporary files for jobs and reports can be created.

Range: 1 to 256 characters

Default: /tmp

TP_CMD

The command to start a transaction program using SNA protocols.

Range: 0 to 256 characters

Default: $XCOM_HOME/bin/xcomtp

Notes:

- $XCOM_HOME is an environment variable
- If only the TCP/IP component is installed with the base components of CA XCOM Data Transport, there will be no default value.

TRACE_PATH

The directory containing the trace data files.

Range: 0 to 256 characters

Default: $XCOM_HOME/trace

Note: $XCOM_HOME is an environment variable.

TRNENCRL_CIPHER

Specify the requested list of ciphers which are used to encrypt the password fields for locally initiated transfers.

The cipher list consists of one or more ciphers from the table that are separated by colons. Each cipher in the cipher list can be optionally preceded by the characters (!) or (-).

If (!) is used then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-) is used then the ciphers are deleted from the list, but some or all of the ciphers can be added again using later options.

Default: COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining
DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB

AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining
RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.
COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To request all ciphers except for any of the DES ciphers, use the following command:

```
TRNENCRL_CIPHER=ALL:'!'DES
```

To request only a 3DES or AES cipher, use the following command:

```
TRNENCRL_CIPHER=3DES:AES
```

To disable the cipher negotiation or remain backward compatible with the earlier releases of CA XCOM Data Transport, use the following command:

```
TRNENCRL_CIPHER=ALL:COMPAT
```

TRNENCRR_CIPHER

This parameter specifies the permitted list of ciphers that are used to encrypt the password fields for remotely initiated transfers. The permitted list of ciphers is matched against the requested list of ciphers provided by the local system by the TRNENCRL_CIPHER parameter. The common cipher with the highest ranking is selected to encrypt the password fields.

The cipher list consists of one or more ciphers that colons separate, with the highest ranked cipher listed first and the lowest ranked cipher listed last. The characters (!), (-) or (+) precedes each cipher in the cipher list.

If (!) is used, then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-) is used, then the ciphers are deleted from the list. Some or all of the ciphers are added again using later options.

If (+) is used, then the ciphers are moved to the end of the list. This option does not add any new ciphers. The option just looks for existing ones to move to the end of the list.

Default: XCOM:ALL:COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining

DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining

RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.
COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation. This is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To permit all ciphers except for any of the DES ciphers and remain compatible with CA XCOM Data Transport versions before r11.6, use the following example:

```
TRNENCRR_CIPHER=ALL:!DES:COMPAT
```

To permit only a 3DES or AES cipher, use the following example:

```
TRNENCRR_CIPHER=3DES:AES
```

Note: To permit all ciphers for transfers from r11.6 but remain backward compatible with earlier versions of CA XCOM Data Transport, use the following example:

```
TRNENCRR_CIPHER=ALL:!XCOM:COMPAT
```

TRNENCRR_DHBITS

Specify the size (in bits) of the prime number that is used during DH (Diffie-Hellman) exchange for remotely initiated transfers or meta-transfers. The exchanged value generates the key that is used to encrypt the password fields transmitted in the CA XCOM Data Transport header.

Default: 1024

Value	Comments
256	DH exchange using a 256-bit prime number
512	DH exchange using a 512-bit prime number
1024	DH exchange using a 1024-bit prime number
2048	DH exchange using a 2048-bit prime number

4096	DH exchange using a 4096-bit prime number

Notes:

- The more bits used for the prime number the more secure the key exchange.
- The more bits used for the prime number the more CPU overhead is required to negotiate a secret value that is based on the prime number.
- Each CA XCOM Data Transport connection generates a unique secret value.
- No DH exchange is performed when the cipher the TRNENCRL_CIPHER/TRNENCRR_CIPHER negotiates is XCOM.

TRUST_DATABASE_NAME

Specifies the name of the database where the trusted tables were created.

Range: 1 to 256 characters

Default: None

TRUST_DATABASE_TYPE

Specifies the database type that resides on the database server.

Range: DB2 or MySQL

Default: DB2

TRUST_ODBC

Specifies the ODBC Data Source Name for the Trusted Tables.

Range: 1 to 16 characters

Default: xcomtrust

TRUST_OWNER

Identifies the owner of the database tables. May be omitted if using MySQL or if it is the same as TRUST_USER when using DB2.

Range: 1 to 32 characters

Default: xcomtrust

TRUST_PASS

Specifies the password of the user (TRUSTED_USER) for the Trusted Tables.

Note: When any changes are made via the Global Parameters tab, the updates encrypt this password and it is saved as TRUST_PASS.ENCRYPTED.

Range: 1 to 128 characters

Default: pass

TRUST_PORT

Specifies the database server port where the trusted database resides.

Range: 1 to 65535

Default: 50000

TRUST_SERVER

Specifies the IP address or name of the server where the trusted database resides.

Range: 1 to 256 characters

Default: None

TRUST_TABLE_PFX

Specifies the prefix to use for the names of the Trusted Tables.

Range: 1 to 16 characters

Default: XCOM_TRUSTED

TRUST_USER

This is a generic user ID that has been defined to the RDMS for the Trusted Tables.

Range: 1 to 64 characters

Default: root

TXPI_BUF_SIZE

For TCP/IP transfers, the internal buffer size for sends and receives. The default size allows multiple CA XCOM Data Transport records to be received in a single socket call. With this default, if your CA XCOM Data Transport record size is less than 32K, CA XCOM Data Transport attempts to receive multiple records in a single socket call. Used for TCP/IP transfers only.

Range: 0 to 65536

Default: 32768

TXPI_RECEIVE_TIMEOUT

The maximum wait time, in seconds, after a TCP/IP failure is detected before a socket error is generated. This defaults to zero and should remain at zero unless it becomes necessary for a broken connection to generate a retryable error. Changing this parameter from zero invokes the TCP/IP function "select", that adds a measure of connection detection, but sacrifices some performance. This parameter is for TCP/IP only.

Range: 0 to 1200

Default: 0

TXPI_SEND_CHECK_FREQ

Indicates the frequency with which CA XCOM Data Transport checks to see if incoming error information is available when sending data. For example, if the value is 5, a check is made every fifth time that data is sent, to determine if data is available for receiving. Larger values give better performance. Smaller values minimize the sending of data after the partner reports an error. Used for TCP/IP transfers only.

Range: 0 to 9999

Default: 10

TXPI_TERM_TIMEOUT

Maximum wait time, in seconds, for the partner to terminate TCP/IP communications. If a transfer terminates normally, both sides of the conversation coordinate the termination, and there should be no need to wait. This timeout occurs only during an error in the termination of the connection. Used for TCP/IP transfers only.

Range: 0 to 999 seconds

Default: 20 seconds

UMASK

Used to set the permissions assigned to a file when the file is being created and received on the system for the first time. The value is expressed as an octal number (base 8). The octal number has the same meaning as in the standard umask command.

Range: 000 to 777

Default: 022

Note:

- For directories—CA XCOM Data Transport sets permissions for a created directory to 7xx, no matter what owner UMASK value was specified. Group and other permissions, represented by xx, represent the permissions with the specified UMASK removed.
- For files – While the file is being transferred, CA XCOM Data Transport sets permissions for a created file to 6xx, where xx represents the permissions with the specified UMASK removed. After the transfer has been completed, CA XCOM Data Transport sets the owner permission with the specified UMASK removed.

USEROVR

This parameter specifies whether the local user ID parameter LUSERID can be used when scheduling a transfer to a remote machine, using the xcomtcp -c5 option.

YES

Specifies that LUSERID can be used.

NO

Indicates that the local user ID (LUSERID) cannot be changed.

Default: YES

XBUFFSIZE

The size of CA XCOM Data Transport's internal buffer.

Important! Do not change this parameter unless you are instructed to do so by Technical Support.

Range: 1 to 32767

Default: 32767

XCOM_CONFIG_SSL

This parameter specifies the configssl.cnf file path and file name.

Range: 1 to 256 characters

Default: $XCOM_HOME/config/configssl.cnf

Note: %XCOM_HOME% is an environment variable.

XCOM_ICUPATH

This parameter specifies the path to ICU shared libraries icudata and icuuc.

Range: 0 to 256 characters.

Default: $XCOM_HOME/bin

XCOM_JVM

This parameter specifies the full path and file name for the Java JVM shared library, libjvm.so.

Range: 1 to 256 characters

Default: /opt/CA/SharedComponents/JRE/1.6.0_32_ALL/lib/i386/client/libjvm.so

XCOM_MYSQLODBC

This parameter specifies the MySQL ODBC driver. It is used for HP only.

Default: /usr/lib/libmyodbc5.sl

XCOM_ODBC

This parameter specifies the full path to the ODBC library (libodbc.so) on Unix.

Range: 0 to 256 characters

Default: None

XCOM_SHLIBPATH

This parameter specifies the path to ODBC driver to be added to SHLIB_PATH. It is used for HP only.

Default: None

XCOM_PASSWORD

This parameter specifies the default password for remotely initiated transfers.

Range: 0 to 31 characters

Default: None

XCOM_SHOW_CIPHER

Specifies whether to display encryption algorithms in the CA XCOM Data Transport queue detailed information, which is used for transfers.

NO

Do not display encryption algorithms in the queue detail information.

YES

Display encryption algorithms in the queue detail information.

Default: NO

XCOM_TRUSTED_OVR

Specifies if the user is permitted to override the user ID by using the USERID parameter for locally initiated trusted transfers. If XCOM_TRUSTED_OVR is set to YES, the user is permitted to override the user ID by using the USERID parameter. If it is set to NO, the user ID of the process that initiated the transfer is used.

Range: YES, NO, Y, N

Default: NO

XCOM_USERID

This is the default user ID for remotely initiated transfers.

Range: 0 to 12 characters

Default: xcom

XCOMHIST

The ODBC Data Source Name that is used to connect to the History Database.

Range: 1 to 32 characters

Default: xcomhist

XCOMHIST_BACKSLASH

Treat a backslash in a file name as a single backslash. This is dependent on the target ODBC system.

Y

Treat a backslash in a file name as a single backslash (\).

N

Treat a backslash in a file name as a double backslash (\\).

Examples

If the ODBC is z/OS or DB2, a single \ is needed in order to display the data correctly.

If the ODBC is PC based, like Mysql, for example, then \\ is treated as a single \.

XCOMHIST_FILE

If the database machine is not connected and available, the name of the flat file to contain insert records. Records are also written into this file as a result of an SQL failure.

Default: $XCOM_HOME/config/xcomhist.inserts

Note: $XCOM_HOME is an environment variable.

XCOMHIST_OWNER

Specifies the ID of the creator of the History Table. May be omitted if it is the same as XCOMHIST_USER.

XCOMHIST_PASSWORD

The password of XCOMHIST_USER.

Range: 1 to 32 characters

Default: xcomhist

XCOMHIST_SPLIT_FILE

If specified as Y, then if an insert fails, the query will be written out as 72-byte records for z/OS SPUFI compatibility.

Y

Split the file into 72-byte records.

N

Do not split the file.

Default: N

XCOMHIST_TBL

The name of the table created for the CA XCOM Data Transport history records.

Range: 1 to 18 characters

Default: xcom_history_tbl

XCOMHIST_USER

Specifies the user ID that has complete access to the XCOMHIST database.

Range: 1 to 32 characters

Default: root

XCOMPRE_LOCAL

Indicates whether the xcompre script file for a locally initiated transfer should be run.

Range: YES, NO

Default: NO

XENDCMD

The name of the post processing command file optionally invoked by the CA XCOM Data Transport transfer program after any type of transfer is finished, whether successful or not. It is invoked after partner communications have ended. A sample command file is provided in $XCOM_HOME/cmd/xcomend.

Range: 0 to 256 characters

Default: None

Note: $XCOM_HOME is an environment variable.

XLOGFILE

The name of the file where CA XCOM Data Transport logs activity. If you do not specify this parameter, the systemwide log file $XCOM_HOME/xcom.log is used. If you specify this parameter with a different file name, the logging information is only sent to the specified file.

Range: 0 to 256 characters

Default: $XCOM_HOME/xcom.log

Note:

- $XCOM_HOME is an environment variable.
- If QUEUE=YES, specify the full path name.

XLPCMD

The name of the post-processing command file used to send print jobs to the spooler. For incoming reports only.

Range: 0 to 256 characters

Default: $XCOM_HOME/cmd/xcomlp

Note: $XCOM_HOME is an environment variable.

XNOTIFYCMD

The name of the command file that CA XCOM Data Transport will use to notify users on the local system of the completion of a transfer. This is normally a shell script that composes a message and invokes a mail or write notification as appropriate.

Range: 0 to 256 characters

Default: $XCOM_HOME/cmd/xcomntfy

Note: $XCOM_HOME is an environment variable.

XPPCMD

The name of the command file used for user-defined post processing for file transfers only. Only used when the local system is receiving the file. A sample command file is provided in $XCOM_HOME/cmd/xcompp.

Range: 0 to 64 characters

Default: None

Note: $XCOM_HOME is an environment variable.

XPRECMD

The name of the command file used for user-defined pre-allocation processing for locally and remotely initiated transfers. Specify this parameter in xcom.glb to invoke the pre-allocation exit xcompre. A sample command file is provided in $XCOM_HOME/cmd/xcompre.

Range: 0 to 64 characters

Default: None

Note: $XCOM_HOME is an environment variable.

XTRACE

Indicates the level of desired execution tracing.

0

No tracing.

1 to 8

Levels of tracing from 1 (lowest) to 8 (highest).

9

Show the contents of the data buffers.

10

Includes levels 1 to 9 with additional detailed technical information.

Default: 0

Note: Turning on the trace can seriously degrade performance.

Local Session Control Parameters

XCOM.SES is a file used by the XCOMD CA XCOM Scheduler service to control the maximum number of sessions available for locally initiated queued transfers for each remote system. When CA XCOM Data Transport is installed, the XCOM.SES file is placed in the directory $XCOM_HOME/config by default where $XCOM_HOME is an environment variable.

You can set a maximum number of partners that can be described in this file by specifying a number in the MAX_SESSIONS_ENTRIES global parameter. For example, if you specify MAX_SESSIONS_ENTRIES=50, you can have up to 50 lines in this file, each one specifying the name of the remote system.

You limit the number of sessions available for CA XCOM Data Transport transfers by setting a parameter in the xcom.ses file for each connection to the remote CA XCOM Data Transport partner. If a session is not available for immediate use, the transfer is queued for subsequent execution.

To specify a remote system, as indicated in the example in this section, replace *connection_profile* with the destination address or name specified by the user in the remote system parameter (REMOTE_SYSTEM, REMOTE_SYSTEM_RF, REMOTE_SYSTEM_SJ, or REMOTE_SYSTEM_SR).

Note: CA XCOM Data Transport reserves memory for information about the number of sessions for each destination based upon the *connection_profile* entries in this file.

Transfers Using SNA

For transfers using SNA, the parameter name *connection_profile* represents the remote system name or partner LU name to which the session limit is applied.

Transfers Using TCP/IP

For transfers using TCP/IP, specify the format of the remote system that is consistent with your site's use of the remote system parameters. If the IP address is used, specify connection_profile in the form of *123.123.78.91*. If host names or domain names are used in the remote system parameter, specify *connection_profile* in the appropriate form of the name.

You may specify a *connection_profile* for each form of address or name that is used.

Note: If the XCOM.SES file does not have an entry for the host name or domain name, the value of the *connection_profile* defaults to one.

Example

In the following example, a sample xcom.ses file of session control parameters for CA XCOM Data Transport partners is shown. It indicates the following:

- That a maximum of eight sessions can be used at any one time for locally initiated transfers to a remote system identified as XCOMQA3
- That a maximum of four sessions can be used at any one time for locally initiated transfers to a remote system identified as XCOMQA4.
- That a maximum of four simultaneous transfers can go to 123.123.78.91.

```
# connection_profile=number_of_sessions_allowed
#
XCOMQA3=8
XCOMQA4=4
123.123.78.91=4
```

Manage the Queue

Use xcomqm at the command line to control the activity of the CA XCOM Data Transport transfer queue, including stopping transfers and displaying the status of a transfer.

The XcomQAPI allows your site to write programs to access the queue. The input and options for the XcomQAPI are the same as those described in the following topics. For more information about the XcomQAPI, see the chapter "The Application Programming Interface."

The xcomqm Command

The xcomqm command lets you maintain CA XCOM Data Transport queues by using the command line interface. This command lets you delete entries from the queue, suspend transfers, resume suspended transfers, display the list of entries in the queue and display detailed information about the queue entries.

Syntax

The syntax for xcomqm is as follows:

```
xcomqm [option] [option] . . .
```

You can use more than one option on a command line. You can also use the same option more than once. Each option must be separated by a space. These options can be listed in any order, but note that when the shell encounters options such as the **-r** option, it performs the command and exits xcomqm without reading the rest of the command line.

Options

The following list explains the options for xcomqm:

-r

Display the release level of xcomqm and exit.

-A*entryname*

Release a held transfer.

-C*entryname*

Get a trace of a transfer.

-D*entryname*

Display details about a queue entry.

-H*entryname*

Hold a scheduled transfer.

-L[a]

List your queued transfers. Use -La for all queued transfers.

-R*entryname*

Remove a queue entry.

-Rf*entryname*

Force the entry to be removed from the queue. Use this command when you want to remove a transfer that is still active.

-R*

Remove all completed or scheduled entries from the queue.

If you use this option on an active transfer, it will interrupt the transfer.

-T*entryname*

Terminate an active transfer.

-T[f]*entryname*

Force termination of the transfer. Use this command when you want to remove a transfer that is still active.

Note: Please be aware that when using SNA/APPC protocols, the TP will ABEND and it may be necessary to restart the underlying SNA software. When using TCP/IP, xcomtcp will ABEND.

-S*entryname*

Suspend a transfer.

-E*entryname*

Resume a suspended transfer.

-d

Display debugging information.

no option

Display the help menu.

-z

Display logging text for transfer.

Note: *entryname* represents the Transaction ID, which is the six-digit name of the transfer entry in the queue.

Purge the Log File

CLEANLOG purges entries from the CA XCOM Data Transport transfer and error log. Run this program on a regular basis to ensure that the log file does not grow too large. Back up the XCOM.LOG before running this program as all items older than the specified age are deleted.

Note: CA XCOM Data Transport should be idle when you run this program.

Syntax

The syntax for CLEANLOG is as follows:

`CLEANLOG` *`number_of_days_old`* `[`*`log_file_name`*`]`

number_of_days_old

Entries in the log longer than the number of days specified are purged.

log_file_name

The name of the log file to clean (optional). The default is the XCOM.LOG file in the current directory. If not specified, the log file name defaults to $XCOM_HOME/xcom.log where $XCOM_HOME is an environment variable.

Example

```
CLEANLOG 5
```

This command removes entries in XCOM.LOG that are older than five days.

About the xcompre Pre-allocation Exit

Use the xcompre exit to validate all locally and remotely initiated transfers before they begin, and to customize parameters for that transfer as indicated below. This exit is invoked by specifying the XPRECMD parameter in xcom.glb. If XPRECMD is set when xcomd is invoked, the pre-allocation exit will be active for remotely initiated transfers. This gives the system administrator additional capabilities to control access, such as allowing only a particular user to perform a transfer, or only allowing transfers to or from a particular directory.

Customization is useful in certain applications where the remote user does not specify the true file name, and the local system uses a database to map the specified name to the true local name. For example, certain e-mail application systems work in this way. By way of additional explanation, for example, an incoming transfer specifies a certain file name, but the administrator of the local system that is receiving the file wants to place the data in a different file, depending on the user ID of the sender. The script retrieves parameter values and sends the relevant information to standard output, using the -g option, and if desired, changes the values using the -p option.

How to Use xcompre

An xcompre shell script is provided with CA XCOM Data Transport as a sample script, and is available online. It is an option to run an exit on the local system before the transfer starts. It can be tailored to meet individual user requirements. The xcompre exit collects and passes transfer ID information (tid) to xcomqm, along with the parameter values specified by the available options, -g and -p.

Note: The -g and -p options, for use with xcomqm, as shown in the sample script are intended only for use with this exit.

The pre-allocation exit is invoked after CA XCOM Data Transport receives the CA XCOM Data Transport header from the remote partner, but before the transfer commences.

For information about shells and script changes, see Shells and Script Changes in this chapter.

For remotely initiated transfers, the pre-allocation exit is invoked after CA XCOM Data Transport receives the CA XCOM Data Transport header from the remote partner, but before the transfer commences. For locally initiated transfers, the pre-allocation exit is invoked based on the value specified by the global parameter XCOM_LOCAL. If XCOMPRE_LOCAL is set, the pre-allocation exit is invoked before sending the CA XCOM Data Transport header to a remote partner.

xcompre Options

The available options for xcompre and their associated parameters are listed below:

-g

Passes transfer ID to xcomqm, and retrieves the values for the following parameters from the local database and passes them to standard output:

- FILE_OPTION
- FILE_TYPE
- LOCAL_FILE
- REMOTE_FILE
- REMOTE_LU
- TRANSFER_TYPE
- USERID

-p

Changes the values for the following parameters based upon the specified transfer ID (tid) and applies them to the transfer:

- FILE_OPTION
- LOCAL_FILE
- REMOTE_FILE

Sample Script

The following is the sample xcompre script:

```
#!/bin/sh
# @(#)xcompre.sh

#       This procedure is invoked by the CA XCOM Data Transport
#       transaction program for all locally and remotely initiated
#       transfers before they begin.

tid=$1

echo $1 > /tmp/$tid.tidlog

# required for HPUX-10
if [ `uname` = 'HP-UX' ]
then PATH=/opt/xcom/bin:$PATH
fi

xcomqm -g$tid |grep USERID > /tmp/$tid.userlog

'egrep' 'payroll|finance' /tmp/$tid.userlog

if [ $? -ne 0 ]
    then
    exit 100
else
    xcomqm -p$tid LOCAL_FILE=/u/jc/test/000040.local  \
    REMOTE_FILE=abcdefg
    FILE_OPTION=CREATE
    2>&1 > /tmp/$tid.errlog
    exit 0
fi
```

Example

In the above sample script, the only file transfers permitted are ones that have *payroll* or *finance* in the USERID. If the USERID does not meet one of these criteria, the transfer is rejected. If the transfer is permitted, then the -p option is used to specify the values to use for LOCAL_FILE, REMOTE_FILE and FILE_OPTION.

About Post Processing Scripts

Use these scripts to handle processing after you have received a file. They are used for controlling print spooling, for managing notification facilities, and for additional processing of incoming files after a successful file transfer is received.

The post processing shell scripts include xcompp, xcomend, xcomlp, and xcomntfy. These are provided with CA XCOM Data Transport as sample scripts and can be tailored to meet individual user requirements, if necessary.

Shells and Script Changes

CA XCOM Data Transport uses the shell specified by the SHELL_CMD parameter for processing scripts. If you customize a script using a shell other than the CA XCOM Data Transport default, you should change the default. However, all other CA XCOM Data Transport scripts will then use that default which may affect the behavior of these scripts.

Process Summary

To modify a post processing script

1. Invoke an editor, such as vi, and open the script you want to modify.
2. Go to the bottom of the script.
3. Add the desired commands to the end of the script.

   Note: If the script ends with an exit statement, enter your commands above exit.
4. Save the file and exit the editor.

Troubleshooting

For script problems, uncomment the debugging statements included in the sample script and rerun the script. The debugging section is clearly marked. Debugging information is written to the file specified in the first statement.

How to Modify Post Processing Parameters

Use the xcompp script after a successful file transfer is received. The, CA XCOM Data Transport parameters that appear in the script can also be used for any modifications that are needed.

About xcompp

The xcompp script is a shell script that contains parameters for post processing of a file. After a successful file transfer is received, xcompp is always turned on (available) and will get invoked automatically. It is only invoked for incoming files and does not work for jobs or reports.

You can reference the parameters in xcompp by putting commands at the end of the script. xcompp does not take any visible actions if no modifications are made to the script.

Sample Script

The following is the sample xcompp script:

```
#!/bin/sh
# @(#)xcompp.sh     1.1     3/27/92 20:19:33

#   This procedure is invoked by the CA XCOM Data Transport transaction
#   program after the transfer is finished
#   before the conversation is terminated.
#   Not every argument will be populated;
#   the values contained in the arguments depend on those
#   provided by the remote system.

###### START OF DEBUG SECTION ######
######
###### UNCOMMENT THE LINES BELOW TO DEBUG THIS SCRIPT ######
#exec >> /tmp/xcompp.out
#exec 2>&1
#set -vx
#PS4='[$0: $LINENO]+ '
#export PS4
###### END OF DEBUG SECTION ######

compression=$1          # Compression flag
shift
notify_flag=$1          # Notify flag
shift
notify_name=$1          # Notify Name
shift
remoteuser=$1           # Notify User
shift
filetype=$1             # File type (Whether it's file, job or report)
shift
fileaction=$1           # File action (Create, replace or append)
shift
datasettype=$1          # Dataset type
shift
carriage_flag=$1        # Carriage return flag
shift
code_flag=$1            # Code flag
shift
recfm=$1                # Record format
shift
lrecl=$1                # Logical record length
shift
remote_reqno=$1         # Remote request Number
shift
local_reqno=$1          # Local Request number
shift
```

```
group=$1                # Group name
shift
sysdata=$1              # System Dependent User Data
shift
xferdata=$1             # Transfer Dependent User Data
shift
ident=$1                # Ident
shift
truncation_flag=$1      # Truncation flag
shift
tmp_file=$1             # Local temporary file name
shift
file=$1                 # Local file name
shift
remote_file=$1          # Remote file name

#    POSTPROCESS HERE !
exit 0
```

Example

The example below shows how you change the permissions on a received file by adding the following to the end of the xcompp script:

```
chmod 777 $tmp_file
```

This will change the permissions on tmp_file to *read*, *write*, and *execute*.

Note: You must type a $ sign before the variable to reference the variable in the xcompp script.

How to Modify the Post Transfer Script

Use the xcomend script for all types of transfers, after a transfer is finished and partner communications have ended, whether the transfer completed successfully or not.

About xcomend

The xcomend script is a script that is invoked by a CA XCOM Data Transport transfer program. This script can be used as provided, or it can be used by system administrators to modify printing commands and parameters to provide additional information about all types of transfers (remote and local; send and receive; successful or failed). Different arguments are passed depending on whether the transfer is a file, job, or report. This script can be used to define the context of the transfer, such as indicating that a remotely initiated send transfer failed with a particular message.

Sample Script

The following is the sample xcomend script:

```
#!/bin/sh
# @(#)xcomend.sh     1.0     8/1/96 12:23:04

#   This procedure is invoked by the CA XCOM Data Transport transfer
#   program after the transfer is finished (whether successful or not).
#   Not every argument will be populated;
#   the values contained in the arguments depend on those provided by the
#   remote system.

###### START OF DEBUG SECTION ######
######
###### UNCOMMENT THE LINES BELOW TO DEBUG THIS SCRIPT ######
#exec > /tmp/xcomend.$1.out
#exec 2>&1
#set -vx
#PS4='[$0: $LINENO]+ '
#export PS4
###### END OF DEBUG SECTION ######

#
#  The first parameters are supplied for any type of transfer.

local_reqno=$1    # local request number/tid (000000 if unassigned)
shift
initiator=$1      # LOCAL or REMOTE
shift
```

```
transfer_type=$1  # FILE or JOB or REPORT
shift
direction=$1      # SEND or RECEIVE
shift
restarting=$1     # RESTARTING or FIRST_TRY
shift
start_time=$1     # Start time of transfer
shift
end_time=$1       # End time of transfer
shift
remote_system=$1  # Remote system name
shift
status=$1         # Status of transfer
shift
error=$1          # XCOM file transfer numeric error code
shift
msg=$1            # error code translated to message text
shift
status_msg=$1     # status message from partner when error on remote sys
shift

remoteuser=$1     # Remote user
shift

remote_reqno=$1   # Remote request Number
shift
group=$1          # Group name
shift
sysdata=$1        # System Dependent User Data
shift
xferdata=$1       # Transfer Dependent User Data
shift
transfer_name=$1  # Transfer Name
shift

tmp_file=$1       # Local temporary file name
shift
file=$1           # Local file name
shift
remote_file=$1    # Remote file name
shift

carriage_flag=$1  # Carriage return flag
shift
code_flag=$1      # Code flag
shift

compression=$1    # Compression flag
shift
```

```
file_recs=$1      # No. of records read/written
shift
file_bytes=$1     # No. of bytes   read/written
shift
blocks=$1         # No. of blocks  sent/received
shift
bytes=$1          # No. of bytes   sent/received
shift

#
#  The next parameters are supplied for file transfers only

if [ $transfer_type = "FILE" ]
then

fileaction=$1         # File action (C, R, or A for create, replace, or append)
shift
datasettype=$1        # Dataset type
shift
recfm=$1              # Record format
shift
lrecl=$1              # Logical record length
shift
truncation_flag=$1    # Truncation flag
shift

#
#  The next group of parameters are supplied for report transfers only

elif [ $transfer_type = "REPORT" ]

then

jobname=$1            # Job name field from JES
shift
jobnumber=$1          # Job number field from JES
shift
class=$1              # print class
shift
copies=$1             # Number of copies to print
shift
form=$1               # Type of form to print this job on.
shift
recfm=$1              # Record format of incoming print job.
shift
lrecl=$1              # Logical record length of incoming report.
shift
blksize=$1            # Block size of incoming report.
```

```
shift
ucs_name=$1          # Name of UCS to be used for this print job.
shift
fcb=$1               # Name of FCB (form control block) for this report.
shift
room_number=$1       # Room number field from JES.
shift
programmer_name=$1   # Programmer name field from JES.
shift
tso_notify=$1        # TSO notify field from JES.
shift
destination=$1       # Destination printer specification.
shift
carriagecontrol=$1   # Type of carriage control characters being used.
shift

#
# There are no extra parameters for remote jobs

fi

#
# PUT YOUR OWN XCOM ENDING CODE HERE !

exit 0
```

How to Modify Printing Processes

Use this script after a print job is received from a remote system. Use the script as provided or modify it, if necessary. The sample script assumes the default print queue is used for printing CA XCOM Data Transport print jobs.

About xcomlp

The xcomlp script is a shell script that is invoked by the CA XCOM Data Transport transaction program that is processing a report transfer.

The contents of the parameters passed to xcomlp depend on what is supplied on the initiating side of the transfer. The only field that can be absolutely relied on is the file name. The sample procedure also assumes that the COPIES= field is valid, in addition to assigning each incoming argument to an appropriately named variable. For information about the processing of machine code characters, see Reports Containing Machine Code Characters later in this chapter

Sample Script

The following is the sample xcomlp script:

```
#!/bin/sh
# @(#)xcomlp.sh  1.1     3/27/92 20:19:33

#   This procedure is invoked by the CA XCOM Data Transport transaction
#   program after an incoming print request is received. Not every argument will
#   be populated; the values contained in the arguments depend on those provided
#   by the remote system. The most general case is that of a host partner that
#   uses the user exit EXIT02  to extract and transmit the JES parameters
#   for the print job.

#   In all cases, the file and copies parameters will be valid.

###### START OF DEBUG SECTION ######
######
###### UNCOMMENT THE LINES BELOW TO DEBUG THIS SCRIPT ######
#exec > /tmp/xcomlp.out
#exec 2>&1
#set -vx
#PS4='[$0: $LINENO]+ '
#export PS4
###### END OF DEBUG SECTION ######

rluname=$1      # Remote lu name
shift
jobname=$1      # Job name field from JES
shift
jobnumber=$1    # Job number field from JES
shift
class=$1        # print class
shift
copies=$1       # Number of copies to print
shift
form=$1         # Type of form to print this job on.
shift
recfm=$1        # Record format of incoming print job.
shift
lrecl=$1        # Logical record length of incoming report.
shift
blksize=$1      # Block size of incoming report.
shift
ucs_name=$1     # Name of UCS to be used for this print job.
shift
```

```
fcb=$1          # Name of FCB (form control block) for this report.
shift
room_number=$           # Room number field from JES.
shift
programmer_name=$1      # Programmer name field from JES.
shift
tso_notify=$1           # TSO notify field from JES.
shift
file=$1   # Name of temporary file into which XCOM 6.2 has placed report
shift
destination=$1          # Destination printer specification.
shift
nodespec=$1             # connection profile name
shift
carriagecontrol=$1      # Type of carriage control characters being used.

lp  ${copies:+-n$copies} < $file
```

How to Customize the Notification Process

The xcomntfy script is used, when notification is requested, to notify users that a transfer has completed successfully. System administrators can modify this script to customize notification procedures, if necessary. For any other post-processing modifications, we recommend that you use xcompp.

About xcomntfy

XCOMNTFY uses one of the following programs to read the queue entry to determine the user id and password to send an email:

- xcommail to use the MAPI (Messaging Application Programming Interface)
- xcomsmtp to use the SMTP interface

Notification is done using the notify script, which is defined to CA XCOM Data Transport for UNIX and Linux with the parameter, XNOTIFYCMD$XCOM_HOME/cmd/xcomntfy, in the xcom.glb global file. $XCOM_HOME is an environment variable.

You need to set up the notify script to use xcommail or xcomsmtp. A sample notify script is distributed with CA XCOM Data Transport.

Sample Script

The supplied script uses mailx(1) to notify the named user logins.

The following is the xcomntfy script:

```
#!/bin/sh
# @(#)xcomntfy.sh    1.2    3/28/92 08:42:41
#   This procedure (xcomntfy) is called by CA XCOM Data Transport
#   transaction program when the NOTIFY parameter is set to YES.
#
#   The arguments passed to the xcomntfy procedure are:
#
#   -q q_entryname
#
#   -h how
#
#   The following $how will be treated as 'M' (mail to user) by default:
#       T - TSO;
#       R - ROSCOE;
#       C - CICS;
#       L - another LU;
#       M - mail to user.
#   The following $how will be treated as 'W' (write to user) by default:
#       W - write to user.
#   The following $how will be treated as 'A' (write to all users) by
#   default:
#       A - all;
#
#   -u login
#
#   Note: 1.Other information about the transfer can be extracted
#       from the output of xcomqm -D$q_entryname command
#       for postprocessing
#         2.If specified in $how method fails, the mail message will be
#       sent to $LOGNAME or root.

#uncomment following to diagnose problems with script
# set -xv

if [ `uname` = 'SunOS' ]
then MAIL=Mail
else
     MAIL=mailx
fi

# required for HPUX-10
```

```
if [ `uname` = 'HP-UX' ]
then PATH=/opt/xcom/bin:$PATH
fi

while test $# -gt 0
do
    case "$1" in
        -h)
            how="$2"
            shift;;
        -q)
            q_entryname="$2"
            shift;;
        -u)
            login="$2"
            shift;;
    esac
    shift
done
case "$how" in
    [TRCLMV]* )
        xcomqm -D$q_entryname|
        $MAIL -s "xfer $q_entryname" ${login:-${LOGNAME:-root}}
>/dev/null 2>&1
        ;;
    [W]* )
        xcomqm -D$q_entryname|
        if [ "$login" != "" ]
        then
        (write $login || $MAIL -s "xfer $q_entryname"
${login:-${LOGNAME:-root}})  >/dev/null 2>&1
        fi
        ;;
    [A]* )
        xcomqm -D$q_entryname|
        (wall || $MAIL -s "xfer $q_entryname" ${login:-${LOGNAME:-root}})
>/dev/null 2>&1
        ;;
    * )
        ;;
esac

# put user defined notification
# script here

exit 0
```

Requirements for SMTP Notifies

The following sections describe the requirements for addresses and related parameters for SMTP notifies.

From Addresses

This section describes the requirements for **from** addresses and related parameters for SMTP notifies (local and remote).

Local Notifies

For an SMTP email *local* notify, CA XCOM Data Transport checks the following parameters to use as the **from** address:

- XCOM_USERID (in the global file)

If neither of these parameters has been specified, then the results are as follows:

- No notify happens.
- CA XCOM Data Transport displays an error message.

Note: The following parameters must also be set:

- LCLNTFYL={ALL|WARN|ERROR}
- LOCAL_NOTIFY=(local user)
- MAIL_TYPE=SMTP (global file parameter)
- NOTIFYL=MAIL
- SMTP_SERVER={SMTP server address|name} (global file parameter)
- XNOTIFYCMD=$XCOM_HOME/cmd/xcomntfy (global file parameter)

Remote Notifies

For an SMTP email *remote* notify, CA XCOM Data Transport checks the following parameters to use as the **from** address:

- First, the USERID of the remote user (in the configuration file)
- Second, XCOM_USERID (in the global file)

If none of these parameters has been specified, then the results are as follows:

- No notify happens.
- CA XCOM Data Transport displays an error message.

Note: The following parameters must also be set:

- MAIL_TYPE=SMTP (global file parameter)
- NOTIFY_NAME=(remote user)
- NOTIFYR=MAIL
- RMTNTFYL={ALL|WARN|ERROR}
- SMTP_SERVER={SMTP server address|name} (global file parameter)
- XNOTIFYCMD=$XCOM_HOME/cmd/xcomntfy (global file parameter)

To Addresses

This section describes the requirements for **to** addresses and related parameters for SMTP notifies (local and remote).

Local Notifies

For an SMTP email *local* notify, CA XCOM Data Transport uses the following parameter as the **to** address:

- LOCAL_NOTIFY (in the configuration file)

If this parameter has not been specified, then the results are as follows:

- No notify happens.
- CA XCOM Data Transport displays an error message.

Note: The following parameters must also be set:

- LCLNTFYL={ALL/WARN/ERROR}
- MAIL_TYPE=SMTP (global file parameter)
- NOTIFYL=MAIL
- SMTP_SERVER={SMTP server address|name} (global file parameter)
- XNOTIFYCMD=$XCOM_HOME/cmd/xcomntfy (global file parameter)

Remote Notifies

For an SMTP email *remote* notify, CA XCOM Data Transport uses the following parameter as the **to** address:

- NOTIFY_NAME (in the configuration file)

If this parameter has not been specified, then the results are as follows:

- No notify happens.
- CA XCOM Data Transport displays an error message.

Note: The following parameters must also be set:

- MAIL_TYPE=SMTP (global file parameter)
- NOTIFYR=MAIL
- RMTNTFYL={ALL/WARN/ERROR}
- SMTP_SERVER={SMTP server address|name} (global file parameter)
- XNOTIFYCMD=$XCOM_HOME/cmd/xcomntfy (global file parameter)

Pluggable Authentication Modules (PAM) Based Authentication

Starting with the r11.5 maintenance level 11112 and higher releases of CA XCOM, administrators have an option to select PAM-based authentication instead of the native mechanism available for the UNIX server.

PAM Overview

The Pluggable Authentication Modules (PAM) are an industry-standard framework providing authentication, account management, session management, and password services. PAM allows programs that rely on authentication to be written independently of the underlying authentication scheme. PAM uses the local password file to authenticate the user accessing the host, you can now take advantage of other authentication mechanisms such as LDAP.

Enabling PAM authentication in CA XCOM

To enable PAM authentication, modify the following global parameters.

1. Set authentication type to PAM.

   AUTH_TYPE=PAM

2. Specify the path where the PAM shared library is present.

   PAM_PATH=/usr/lib

Configure PAM Service

PAM can be configured through the use of either the /etc/pam.conf file or the /etc/pam.d directory. CA XCOM uses the service name of xcomauth.

- The /etc/pam.conf file

  Update the pam.conf file with the service name xcomauth and the other authentication settings detailed in the requirements.

  As an example:

  xcomauth auth XXXX XXXX

  xcomauth account XXXX XXXX

- The /etc/pam.d directory

  The directory is an alternate configuration mechanism. In this scheme, each policy is contained in a separate file bearing the name of the service it applies to. Create an xcomauth config file with the authentication settings detailed in the requirements.

Note: If you are using LINUX-PAM, then the pam_userpass module must precede the actual authentication module. This module is distributed with XCOM installation and is installed at $XCOM_HOME/redistrib/pam_userpass directory. For more details on PAM configuration, refer to the PAM documentation.

Trusted Access Database

Starting with r11.5, the CA XCOM Data Transport Trusted Access facility has been changed so that it works the same way across platforms. Two relational database tables are used to store trusted user authentication information:

XCOM_TRUSTED_SYS table

The XCOM_TRUSTED_SYS table contains the name of the remote SYSTEM_NAME from which the trusted transfer is initiated.

- Each row in this table can contain a NOTE as well as an optional GROUP name. There can be multiple rows for each SYSTEM; that is, each system can have multiple groups defined.
- However, one user cannot belong to more than one group on a SYSTEM_NAME.

XCOM_TRUSTED_USERS table

The XCOM_TRUSTED_USERS table contains the remote SYSTEM_NAME and the USER_NAME.

- If the user is considered as a member of a group, then the GROUP_NAME must also be specified.

Set Up Trusted Access

The Trusted Access feature allows a transfer to be sent to a remote partner without actually specifying the user ID and password in the transfer. Trusted Access transfers can be sent to and from UNIX as well as a number of other partners.

Trusted Security

The Trusted Access feature allows a transfer from a remote system to be sent without requiring the user ID and password to be specified in a CA XCOM Data Transport transfer. In addition, a group user ID can be set up so that only one user ID can be used for multiple users sending into this system.

Trusted Access has a GUI interface, making it easy to build a list of trusted partners, which resides in the following database tables:

- XCOM_TRUSTED_SYS
- XCOM_TRUSTED_USERS

For more information, see the section Setting Up Trusted Access in the chapter "Operating Environment."

Releases of CA XCOM Data Transport Using Trusted Transfers

The following prior versions and releases of CA XCOM Data Transport can send and receive trusted transfers to and from each other in addition to any CA XCOM Data Transport r11.6 platforms:

- Unicenter CA XCOM Data Transport for Windows NT Server/Professional r11
- Any Unicenter CA XCOM Data Transport for UNIX and Linux r11 platform
- CA XCOM Data Transport for Windows Server/Professional r11.5
- Any CA XCOM Data Transport for UNIX and Linux r11.5 platform
- CA XCOM Data Transport for z/OS r11.5

Reports Containing Machine Code Characters

CA XCOM Data Transport processes machine code characters in reports sent from the mainframe under either of these conditions:

- CODE=MACHINE is specified for FILETYPE=REPORT.
- The report is sent from a machine code type data set, such as FBM.

For each line that begins with a supported printer channel command code, CA XCOM Data Transport inserts the appropriate number of form feed and line feed characters.

CA XCOM Data Transport supports the following machine code characters:

- SKIP ONE LINE
- SKIP TWO LINES
- SKIP THREE LINES
- SKIP TO CHANNEL1-12
- WRITE NO SPACE
- WRITE ONE SPACE
- WRITE TWO SPACES
- WRITE THREE SPACES
- WRITE & SKIP TO CHANEL1-12

All channel commands (that is, Skip to Channel, Write and Skip to Channel) are translated to a form feed (0C) followed by *n*-1 line feeds (0A). The Skip to Channel or Write and Skip to Channel number is represented by *n*.

Example

Skip to Channel 1 is translated to a form feed (0C) followed by 0 line feeds. Skip to Channel 2 is translated to a form feed (0C) followed by 1 line feed (0A), and so on.

About History Records

You must choose a relational database to house the CA XCOM Data Transport history records.

CA XCOM Data Transport supports the following databases except where noted:

- DB2
- MySQL

The following procedures use MySQL as the database engine. MySQL is an Oracle product that can be downloaded and installed on a wide variety of platforms.

NOTE: DB2 is not supported on the HP-UX PA RISC operating system.

Step 1. Install a Relational Database Management System

Install the relational database management system (RDMS) that you will use to manage your history records.

To install an RDMS

Follow the procedures in the RDMS documentation (MySQL for this example).

Step 2. Create a Database

To create a database

Enter the following:

CREATE DATABASE XCOMHIST;

Step 3. Create the XCOM History Table

You need to create a history table to store your history records.

To create the XCOM History Table

1. Set up the history table using the following code (for MySQL):

```
use database XCOMHIST;
CREATE TABLE XCOM_HISTORY_TBL (
StartDate               char(7),
StartTime               char(6),
reqno                   char(7),
sysname                 char(8),
sysid                   char(4),
GregStartDate           char(8),
GregEndDate             char(8),
EndDate                 char(7),
EndTime                 char(6),
SchedStartDate          char(7),
SchedStartTime          char(6),
luname                  char(18),
rluname                 varchar(256),
mode                    char(9),
file                    varchar(257),
lfile                   varchar(257),
transfer_id             char(11),
msg_flag                char(5),
msg1                    varchar(121),
msg2                    varchar(121),
msg3                    varchar(121),
msg4                    varchar(121),
msg5                    varchar(121),
error_message           varchar(257),
user_name               varchar(32),
cpuflag                 char(1),
```

```
initiated_by            char(1),
remote_cpuflag          char(1),
protocol                char(6),
idest                   varchar(256),
remote_system_relay     varchar(256),
nodespec                varchar(65),
cid                     varchar(18),
rpgm                    varchar(65),
tpn                     varchar(65),
notifyl                 char(1),
notify_term             varchar(65),
localnotify             varchar(65),
remote_file_relay       varchar(257),
compression             char(1),
notifyr                 char(1),
who                     char(13),
remoteuser              char(13),
domain                  char(16),
trusted                 char(1),
trusted_ovr             char(1),
filetype                char(1),
fileaction              char(1),
datasettype             char(1),
carriageflag            char(1),
codeflag                char(1),
default_uid             varchar(32),
releasex                char(4),
keyx                    char(5),
varpart                 varchar(257),
auto_restart            char(1),
pack_text_records       char(1),
truncation              char(1),
den                     char(1),
```

```
expdt                   char(6),
retpd                   char(5),
expdate_flag            char(1),
label                   char(4),
tape                    char(1),
unitct                  char(3),
volct                   char(4),
volsq                   char(4),
storcls                 char(9),
datacls                 char(9),
mgtclas                 char(9),
dsntype                 char(9),
vlr_flag                char(1),
umask_flag              char(3),
labelnum                char(5),
tapedisp                char(1),
codetabl                char(4),
seclabel                char(9),
lclntfyl                char(1),
rmtntfyl                char(1),
xcomshowcipher          char(1),
configssl               varchar(257),
xcomfullssl             char(4),
encryptionalgorithm     varchar(257),
restart_flag            char(1),
compression_flag        char(1),
restart_negotiation_flag char(1),
alpha                   char(10),
beta                    char(10),
remote_reqno            char(7),
remote_system           varchar(64),
cpcnt                   char(5),
alcrypt                 char(1),
```

```
groupname               char(16),
sysdata                 char(11),
xferdata                char(11),
allocunit               char(1),
prialloc                char(6),
secalloc                char(6),
diralloc                char(6),
flen                    char(11),
ident                   char(11),
write_eof               char(1),
tape_flag               char(1),
label_type              char(4),
label_number            char(5),
acf_uid                 varchar(25),
local_fname             varchar(257),
msg_count               int    default 0,
recfm                   char(4),
lrecl                   char(6),
blksize                 char(6),
local_recfm              char(4),
local_lrecl              char(6),
local_blksize            char(6),
volume                  char(11),
unit                    char(11),
uic                     char(10),
whenx                   char(15),
print_class             char(1),
destination             varchar(22),
form                    char(11),
fcb                     char(5),
copies                  char(4),
holdflag                char(1),
reporttitle             varchar(22),
```

```
carriagecontrol         char(1),
spoolflag               char(1),
disposition             char(1),
eol_classes             varchar(128),
convert_classes         varchar(128),
metacode_classes        varchar(128),
metaflag                char(1),
jobname                 char(9),
jobnumber               char(9),
ucs_name                char(5),
programmer_name         varchar(21),
account_number          char(5),
room_number             char(5),
tso_notify              char(9),
shell_cmd               varchar(256),
server_name             varchar(64),
e2a_table               varchar(256),
a2e_table               varchar(256),
sdstype                 varchar(21),
sdsnotify               varchar(21),
sdsrc                   varchar(21),
sdsfeedback             char(5),
sdsmessage              varchar(81),
sdsid                   varchar(21),
login_username          char(16),
access_username         varchar(21),
mail_username           varchar(256),
relay_username          varchar(65),
login_connection        int    default 0,
access_connection       int    default 0,
create_directories      char(1),
num_of_new_dir          int    default 0,
attached_to_server      char(1),
```

```
connection_logged_in    char(1),
drive_mapped            char(1),
errmsgfile              varchar(257),
transfer_mode           char(1),
xcomlib_file            varchar(257),
xcomlib_add             char(1),
xcomlib_record_no       char(1),
xcomlib_record_recv     char(1),
xcomlib_count           int    default 0,
xcomlib_checkpoint_count int   default 0,
encryptionkeybits       int    default 0,
logged_in               char(1),
sslx                    char(1),
starting_state          int    default 0,
conversation_type       int    default 0,
header_length           int    default 0,
ovrlength               int    default 0,
recsize                 int    default 0,
version                 int    default 0,
silent                  int    default 0,
trace                   int    default 0,
buffsize                int    default 0,
maxreclen               int    default 0,
checkpoint_count        int    default 0,
num_of_retires          int    default 0,
retry_time              int    default 0,
state                   int    default 0,
next_state              int    default 0,
record                  int    default 0,
beta_block              char(10),
alpha_block             char(10),
local_user              char(12),
local_password          char(32),
```

```
local_domain            char(16),
bytes                   bigint    default 0,
record_count            bigint    default 0,
byte_count              bigint    default 0,
block_count             bigint    default 0,
bytes_trans             bigint    default 0,
status                  char(1),
port                    char(5),
exec_priority           smallint    default 0,
sched_priority          smallint    default 0,
pso_flag                char(1),
local_member_name       char(8),
katakana_control        char(1),
katakana_in_data        char(1),
katakana_in_hdr         char(1),
shift_in_chars          char(1),
dbcs_type               char(1),
charset                 char(4),
burst                   char(50),
user_banner             char(1),
destination_flag        char(1),
writer_program          char(8),
local_dataset_rrds      char(128),
secure_socket           char(1),
filedata                char(1),
vtam_sends              int    default 0,
vtam_recvs              int    default 0,
file_output_count       int    default 0,
file_input_count        int    default 0,
micr                    char(8),
the_release             char(1),
dlink_flag              char(1),
endevor_flag            char(1),
```

```
cics_notify_userid      char(12),
notify_flag             char(1),
pgmlib_flag             char(1),
remote_member_name      char(8),
gdg_flag                char(1),
remote_gdg_flag         char(1),
remote_fileaction       char(1),
remote_sysid            char(4),
remote_sysname          char(8),
local_cpuflag           char(1),
remove_trailing_blanks  char(1),
restart_supported       char(1),
local_gateway_guid       char(36),
remote_gateway_guid      char(36),
file_error_count         smallint    default 0,
session_error_count      smallint    default 0,
vtam_error_count         smallint    default 0,
total_error_count        smallint    default 0,
local_volume             char(11),
local_unit               char(11),
invoking_jobname         char(8),
create_delete_flag       char(1),
pds_compress_flag        char(1),
last_field               char(1),
pds_member_number       smallint  default 0,
cpu_time                bigint    default 0,
tcb_time                bigint    default 0,
srb_time                bigint    default 0,
ziip_time_ziip          bigint    default 0,
ziip_time_cpu           bigint    default 0,
ziip_eligible_time      bigint    default 0,
record_count2           bigint    default 0,
vtam_send_count         bigint    default 0,
```

```
vtam_receive_count      bigint    default 0,
file_output_count2      bigint    default 0,
file_input_count2       bigint    default 0,
local_ciphers           char(40)
negotiated_cipher       char(1)
primary_allocation2          char(8),
secondary_allocation2        char(8),
directory_allocation2        char(8),
allocation_record_units      char(1),
extended_attributes          char(1),
transfer_protocol            char(1),
local_charset                char(60),
remote_charset               char(60),
local_ccsid                  int           default 0,
remote_ccsid                 int           default 0,
source_ccsid                 int           default 0,
target_ccsid                 int           default 0,
local_conversion_technique   char(8),
remote_conversion_technique  char(8),
mbcs_input_error_action      char(1),
mbcs_conversion_error_action char(1),
local_delim_encoding         char(1),
remote_delim_encoding        char(1),
mbcs_input_repl_char         char(8),
mbcs_conversion_repl_char    char(8),
local_delim_options          char(16),
remote_delim_options         char(16),
mbcs_input_repl_count        int           default 0,
mbcs_conversion_repl_count   int           default 0,
transmission_format          char(1)
                        );
```

The history table is set up.

2. Create a unique index on the table to ensure that no duplicate record gets inserted:

```
create unique index XCOM_HISTORY_IDX on XCOM_HISTORY_TBL (

StartDate, StartTime, reqno, micr, sysname, sysid);

create unique index XCOM_MICR_IX on XCOM_HISTORY_TBL (

micr, reqno, sysname, sysid)
```

The history table now has a unique index.

3. To create remaining indexes, use the following syntax:

```
create index XCOM_HIST_DATEIDX on XCOM_HISTORY_TBL (
StartDate,
StartTime,
EndDate,
EndTime);

create index XCOM_REQ_IDX on XCOM_HISTORY_TBL (
reqno,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);

create index XCOM_RECP_IDX on XCOM_HISTORY_TBL (
luname,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);

create index XCOM_USER_IDX on XCOM_HISTORY_TBL (
user_name,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);

create index XCOM_IDNT_IDX on XCOM_HISTORY_TBL (
transfer_id,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);

create index XCOM_SYST_IDX on XCOM_HISTORY_TBL (
remote_system,
pds_member_number,
```

```
StartDate,
StartTime,
sysname,
sysid);
```

The history table now has 6 more indexes.

Step 4. Modify the XCOM Default Global Table

You need to modify the XCOM default global table to include certain parameters.

To modify the XCOM default global table

Add the following parameters:

SYSNAME and SYSID

Eight- and four-character values that uniquely identify a CA XCOM Data Transport system.

These are needed because history records can be accumulated from many different systems and system types. The values of these parameters are initially set during installation of CA XCOM but can be changed at any time.

XCOMHIST

The name that CA XCOM Data Transport will use to connect to ODBC.

XCOMHIST_TBL

The name of the table that was created.

The default value is xcom_history_tbl.

XCOMHIST_FILE

The name of the file that will contain SQL insert statements that could not be executed due to inaccessibility to the ODBC server. These records are also written if an attempt to insert the record fails.

The records may be used to insert history records directly into the database.

```
mysql-> use xcomhist
```

```
mysql-> source /xcomhist/history_records (pointed to by XCOMHIST_FILE)
```

XCOMHIST_SPLIT_FILE=Y/N

Specifies whether to split the INSERT statements created when ODBC fails into 72-byte segments. IBM's SPUFI, a DB2 utility, takes only 80-byte records (72 usable) as input, whereas MySQL takes as input the complete statement and does not handle 72-byte checks; DB2 does not handle complete statements (unless they are less than 72 bytes).

XCOMHIST_BACKSLASH=Y/N

Inserts to the named table include the name of the local file on the PC. Files are typically named c:\folder\folder\file.

MySQL interprets \\ sequences as a single backslash character both in the actual insert and in the statement created when an insert fails. IBM treats the characters differently. If your target system is DB2 then XCOM_BACKSLASH=Y creates file names as c:\folder\folder\file.

If N is specified then XCOM creates the path as c:\\folder\\folder\\file. Note that when the data is actually inserts into the table, the \\ is interpreted as a single slash.

XCOMHIST_USER

Names the user ID that is used to establish a connection to the database.

XCOMHIST_PASSWORD[.ENCRYPTED]

Specifies the password of XCOMHIST_USER that is used to establish a connection to the database.

XCOMHIST_OWNER

The ID of the creator of the History Table. This value can be omitted if the owner is XCOMHIST_USER.

Step 5. Install the Appropriate ODBC Driver

Multiple Windows PCs and multiple Linux/UNIX workstations can share the same XCOM History Table. Therefore the database only needs to be installed once. However the ODBC and JDBC drivers must be installed on the same machines that XCOM is installed on.

The 32-bit version of ODBC must be installed. To obtain the ODBC software visit www.unixODBC.org. To install ODBC follow the instructions that come with the ODBC software.

In addition to the ODBC software, the database software has its own ODBC driver that must be installed.

The MySQL ODBC driver can be downloaded from: http://dev.mysql.com/downloads/connector/odbc/

The DB2 ODBC driver is part of IBM's DB2 product

Step 6. Configure ODBC

Note: Configuration is different for MySQL and DB2. DB2 configuration is accomplished by using the DB2/UDB configuration tool. The following describes one method of configuring the ODBC driver for MySQL. Note that the instructions may vary depending on the release installed. The steps needed for DB2 are similar. For more information refer to the documentation provided by the database vendor.

To configure ODBC

1. Use the odbcinst-j command to find the location of the odbcinst.ini and odbc.ini files.
2. Edit the odbcinst.ini file and set the DRIVER= parameter to the MySQL 32-bit ODBC driver. For example:

   ```
   DRIVER=/opt/MySQL/mysql-connector-odbc-5.1.7-linux-glibc2.3-x86-32bit/lib/libmyodbc5.so
   ```

3. Edit the odbc.ini file and add the following parameters:

   [Data Source Name]

   DESCRIPTION={option}

   DRIVER=MySQL ODBC 5.1 Driver name

   SERVER=XCOM_MACHINE

   PORT=3306

   DATABASE=XCOM History Database Name

   UID=xcomuser

   PWD=mypassword

   Fill in the form as follows:

   Data Source Name

   XCOMHIST or another name defined to be the data source name.

   Description

   Optional; can contain whatever you like.

   Driver

   The name of the ODBC driver as defined in the odbcinst.ini file.

   Server

   The name of the machine containing the database.

   Database

   The name of the database created in Step 2, Create a Database.

   Port

The port number defined when MySQL was installed. The default port is 3306.

UID

The name of the user that will be connecting to the database. On the machine in which the database is installed the user should be set to root. On all other machines connecting to this database the name should be something common, like XCOMUSER.

PWD

Should be set on the machine that owns the database and must match the password that was set when MySQL was installed.

4. Verify the ODBC connection.

   Issue the following command:

   ```
   -isql –v {DSN}
   ```

   Where {DSN} is the datasource name in the ODBC.ini file that was added in step 3.

   Example:

   ```
   isql –v XCOMHIST
   ```

   A connected message will be displayed if the connection is successful.

5. Database authorization must be established for the user defined in the XCOMHIST_USER parameter within the XCOM global parameters.

   Enter the following:

   ```
   CREATE user 'xcomhist_user'@'%';
   ```

   The % indicates that you will accept xcomhist_user from any host.

   To be more restrictive, code the following (for example):

   ```
   CREATE user 'xcomhist_user'@'user01-xp.ca.com';
   ```

   Grant each user accessing your database with certain privileges:

   ```
   GRANT {all/select,insert}  on xcomhist.* to 'xcomhist_user'@'{%/machine}';
   ```

6. You are now ready for XCOM to create history records.

Configuring JDBC Database Access

The standalone UI in CA XCOM Data Transport is a Java based application that uses JDBC to communicate with the database for accessing the trusted transfer tables. If you do not plan on using trusted transfers then you can skip configuring the JDBC drivers.

To configure the DB2 JDBC drivers:

1. Locate the DB2 database driver jar files db2jcc.jar and db2jcc_license_cu.jar. In most installations these files are in the DB2 java directory.
2. Copy the database driver jar files to the XCOM_HOME/lib directory.
3. Add the directory path of the database driver jar files as an external directory (-java.ext.dirs) in the XCOM_HOME/bin/StandaloneUI.sh file.

Example:

```
-Djava.ext.dirs=$XCOM_HOME/lib:/opt/DB2path/java
```

To configure the MySQL JDBC drivers:

1. Locate the MySQL database driver jar file mysql-connector-java-x.x.x-bin.jar, where x.x.x is the release of MySQL. If the file is not part of your MySQL installation, it can be found on the mysql.com site by looking for the mysql connector/j feature.
2. Manually copy the database driver jar file to the XCOM_HOME/lib directory.
3. Add the directory path of the database driver jar file as an external directory (-Djava.ext.dirs) in the XCOM_HOME/bin/StandaloneUI.sh file.

Example:

```
-Djava.ext.dirs=$XCOM_HOME/lib:/opt/mysqlpath/java
```

Security Considerations for History Records

CA XCOM Data Transport checks security by ensuring that the user requesting history records is a member of the xcomadm group and/or the xcomsadm group. CA XCOM Data Transport returns records as follows:

- If the user is not a member of either xcomadm or xcomsadm, then the transfers returned are limited to those containing that user's user ID.
- If the user is a member of xcomadm, then CA XCOM Data Transport does *both* of the following:
  - Returns transfers in the queue for *all* users and allows operator functions against those transfers.
  - Returns history records for *all* transfers for the target machine (SYSNAME, SYSID), regardless of the owner.
- If the user is also a member of xcomsadm, then CA XCOM Data Transport returns history records for *any* machine included in the history table.

Instructions to Upgrade Existing R11.5 History Database

A sample SQL file with the commands to upgrade the existing r11.5 History database and tables is provided in the following file:

```
$XCOM_HOME/config/histupdateDB.sql
```

Note: This file will need to be updated to match the database management system being used at your site.

To upgrade the XCOM R11.5 History Table

Upgrade the history table using the following code (for MySQL):

```
USE XCOMHIST;
ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN msg_count                   SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN login_connection            SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN access_connection           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN num_of_new_dir              SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN xcomlib_count               SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN xcomlib_checkpoint_count    SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN encryptionkeybits           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN starting_state              SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN conversation_type           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN header_length               SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN ovrlength                   SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN recsize                     SET DEFAULT 0;
```

```
ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN version                    SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN silent                     SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN trace                      SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN buffsize                   SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN maxreclen                  SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN checkpoint_count           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN num_of_retries             SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN retry_time                 SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN state                      SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN next_state                 SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN record                     SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN bytes                      SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN record_count               SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN byte_count                 SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN block_count                SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN bytes_trans                SET DEFAULT 0;
```

```
ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN exec_priority              SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN sched_priority             SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN vtam_sends                 SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN vtam_recvs                 SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN file_output_count          SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN file_input_count           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN file_error_count           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN session_error_count        SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN vtam_error_count           SET DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ALTER COLUMN total_error_count          SET DEFAULT 0;
ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN invoking_jobname    char(8)   DEFAULT ' ' AFTER `local_unit`;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN create_delete_flag  char(1)  DEFAULT ' ' AFTER `invoking_jobname`;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN pds_compress_flag  char(1)  DEFAULT ' ' AFTER `create_delete_flag`;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN pds_member_number   smallint  DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN cpu_time            bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN tcb_time            bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN srb_time            bigint    DEFAULT 0;
```

```
ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN ziip_time_ziip      bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN ziip_time_cpu       bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN ziip_eligible_time  bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN record_count2       bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN vtam_send_count     bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN vtam_receive_count  bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN file_output_count2  bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN file_input_count2   bigint    DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN primary_allocation2          char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN secondary_allocation2        char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN directory_allocation2        char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN allocation_record_units      char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN extended_attributes          char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_ciphers                char(40)  DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN negotiated_cipher            char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN transfer_protocol            char(1)   DEFAULT ' ';
```

```
ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_charset                char(60)  DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN remote_charset               char(60)  DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_ccsid                  int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN remote_ccsid                 int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN source_ccsid                 int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN target_ccsid                 int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_conversion_technique   char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN remote_conversion_technique  char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN mbcs_input_error_action      char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN mbcs_conversion_error_action char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_delim_encoding         char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN remote_delim_encoding        char(1)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN mbcs_input_repl_char         char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN mbcs_conversion_repl_char    char(8)   DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN local_delim_options          char(16)  DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN remote_delim_options         char(16)  DEFAULT ' ';

ALTER TABLE  XCOM_HISTORY_TBL
```

```
   ADD COLUMN mbcs_input_repl_count        int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN mbcs_conversion_repl_count   int       DEFAULT 0;

ALTER TABLE  XCOM_HISTORY_TBL
   ADD COLUMN transmission_format          char(1)   DEFAULT ' ';
```

Create the Index on newly added fields using the below syntax:

```
create index XCOM_REQ_IDX on XCOM_HISTORY_TBL (
reqno,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);
create index XCOM_RECP_IDX on XCOM_HISTORY_TBL (
luname,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);
create index XCOM_USER_IDX on XCOM_HISTORY_TBL (
user_name,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);
create index XCOM_IDNT_IDX on XCOM_HISTORY_TBL (
transfer_id,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);
create index XCOM_SYST_IDX on XCOM_HISTORY_TBL (
remote_system,
pds_member_number,
StartDate,
StartTime,
sysname,
sysid);
```

Interdependent Transfers

CA XCOM Transfer Control, referred to as XTC processing, makes use of transfer parameters allowing interdependent transfers to be defined as a single group. Transfers in the same group can be held, purged, or released depending on the successful or unsuccessful completion of other transfers in the group.

For important information about XTC processing, see the following sections:

- Transfer Control (XTC) Parameters
- Coding Interdependent Transfers.

Coding Interdependent Transfers

This section explains how to code a set of interdependent transfers for CNF configuration files.

NEWXFER

The parameter NEWXFER must be used at the beginning of each of the multiple transfer requests, as shown in the following diagram:

```
                ...
                ...
                ...
transfer request 1 parameters
                ...
                ...
                ...
#============================
CONTROL=NEWXFER
#============================
                ...
                ...
                ...
transfer request 2 parameters
                ...
                ...
                ...
#============================
CONTROL=NEWXFER
#============================
                ...
                ...
                ...
transfer request 3 parameters
                ...
                ...
                ...
```

XTCNET

The parameter XTCNET is used to indicate which transfer requests belong to the same group (are dependent on each other). Suppose, for example, that the transfers XFER1, XFER2 and XFER3 need to be grouped together. This can be accomplished simply by assigning the name of the group (say, GROUP1) to XTCNET and including XTCNET=GROUP1 in the definition of each of the three transfer requests.

XTCJOB

Each interdependent transfer request must have a unique name, which is assigned to the XTCJOB parameter. Thus, if XFER1, XFER2 and XFER3 are the names of three interdependent transfer requests, their transfer definitions must contain the parameter XTCJOB=XFER1, XTCJOB=XFER2, and XTCJOB=XFER3, respectively.

Other XTC Parameters

XTC parameters other than XTCNET and XTCJOB are used to indicate the dependencies obtaining between the transfer requests. The outcome of a transfer request can affect as many as eight other transfer requests. In addition to the XTC parameters, other configuration parameters can (and some, for example, the local file name, must) be used when defining interdependent transfer requests.

HOLD_TRANSFER

HOLD_TRANSFER=YES must be coded in each transfer request that is dependent on the successful/unsuccessful completion of another transfer request (see Example 1).

HOLDCOUNT

However, HOLD_TRANSFER=YES need not be coded if the holding/releasing of the transfer request is controlled by the HOLDCOUNT parameter (see Example 2).

Multiple Jobs

You can also define multiple jobs (see Example 3).

Password Protection by Encryption

CA XCOM Data Transport always protects the password by using encryption. CA XCOM Data Transport encrypts the password in the configuration file and during transmission. A parameter is used to select the cipher that is used for encrypting the password.

How Transmission Password Encryption Works

The cipher that is used to encrypt the password during transmission is controlled by using the TRNENCRL_CIPHER/STCTRNENCRL_CIPHER and TRNENCRR_CIPHER parameters. Each of these parameters provides a list of ciphers. TRNENCRL_CIPHER/STCTRNENCRL_CIPHER provides the list of requested ciphers for locally initiated connections. TRNENCRR_CIPHER provides a ranked list of permitted ciphers for remotely initiated connections. See the description of the TRNENCRL_CIPHER and TRNENCRR_CIPHER parameters in the XCOM.GLB Parameters section for the full list of the supported ciphers. See the description of the TRNENCRL_CIPHER and STCTRNENCRL_CIPHER parameters in Appendix A: Parameters for the full list of supported ciphers.

When a connection is started to the remote system, the requested list of ciphers for the (STC)TRNENCRL_CIPHER parameter is sent to the remote system. The remote system compares that list with its list of permitted ciphers from the TRNENCRR_CIPHER parameter. The common cipher with the highest ranking on the permitted list is selected to encrypt the password fields during transmission.

Once the cipher is selected a key is generated, the generated key is valid for the current connection only. For the CA XCOM proprietary cipher, a CA proprietary technique is used to exchange the encryption key. For all other ciphers, the key exchange is done using a DH (Diffie-Hellman) key exchange.

When using DH key exchange, the TRNENCRR_DHBITS parameter on the remote system controls the size of the prime number that is used during the key exchange. Consider the following factors when determining the value for the TRNENCRR_DHBITS parameter:

- The more bits used for the prime number the more secure the key exchange.
- The more bits used for the prime number the more CPU overhead is required to negotiate a secret value that is based on the prime number.
- Each CA XCOM Data Transport connection generates a unique secret value.

When using the DH key exchange, the secret value that is negotiated is then hashed using the SHA1 hash function to generate the encryption key that is used for the CA XCOM Data Transport connection.

Note: The CA XCOM proprietary cipher is compatible with releases of CA XCOM Data Transport before r11.6. However, releases of CA XCOM Data Transport before r11.6 do not have the capability to negotiate the cipher that is used for password encryption and assumes the use of the CA XCOM proprietary cipher. To allow incoming connections from CA XCOM Data Transport releases before r11.6, include the COMPAT option on the TRNENCRR_CIPHER parameter.

The cipher negotiation can be enabled or disabled by using the COMPAT option of (STC)TRNENCRL_CIPHER.

Security of Passwords in Traces and XCOM Files

CA XCOM Data Transport automatically encrypts passwords in transfers, traces, and queue files.

Transfer Control (XTC) Parameters

The parameters described in this section form a feature of CA XCOM Data Transport known as CA XCOM Data Transport Transfer Control (XTC). These parameters are used to handle dependencies between multiple transfers. For example, one transfer must complete in a certain way before another can start. They provide the means by which interdependent transfer requests can be defined and processed as a single group. A transfer request belonging to such a group can cause another transfer request in the same group to be held, purged, or released (either conditionally or unconditionally) from the transfer request queue.

For further discussion of interdependent transfers, see the section Coding Interdependent Transfers that follows the list of XTC parameters.

HOLD_TRANSFER

Specifies whether a queued transfer is to be held.

Valid only when QUEUE=YES.

Y

Causes the queued transfer to be held.

N

Causes the queued transfer not to be held.

Default: N

HOLDCOUNT

Associates a number with a transfer request that is incremented or decremented by the successful or unsuccessful completion of other transfer requests. As long as the number is greater than 0, the transfer is not released.

0 to 255

Specifies a value that controls the holding/releasing of a transfer request. The transfer is released when the value of the parameter reaches 0.

Note: See the description of the parameters XTCERRDECR, XTCERRINCR, XTCGOODDECR, and XTCGOODINCR, which can decrement and increment the value of the HOLDCOUNT parameter.

Default: 0

XTCERRDECR

Specifies the transfer requests for which the HOLDCOUNT parameter value is decremented when the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if a transfer completes unsuccessfully.

Note:

- VTAM errors are not considered failed file transfers, because they are retried later.
- If the current file transfer fails, the value of the HOLDCOUNT parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the HOLDCOUNT value has already reached zero.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRINCR

Specifies the transfer requests for which the HOLDCOUNT parameter value is incremented if the current file transfer fails.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if the current file transfer fails.

Notes:

- A VTAM communications error does not count as a failure.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRPURGE

Specifies the transfer requests to be purged if the transfer concludes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be purged if the current file transfer concludes unsuccessfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRREL

Specifies the transfers to be released if the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be released if the current transfer completes unsuccessfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODDECR

Indicates an XTCNET job whose hold count decrements if the file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if the current transfer completes successfully.

Notes:

- If the current file transfer completes successfully, the value of the HOLDCOUNT parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the HOLDCOUNT value has already reached zero.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODINCR

Indicates the transfer requests whose HOLDCOUNT parameter is incremented when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODPURGE

Specifies the transfer requests to be purged when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be purged if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODREL

Specifies the transfer requests to be released if the current file transfer concludes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be released if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCJOB

Defines the name of a transfer request belonging to the group of interdependent transfer requests named through the XTCNET parameter.

xxxxxxxx

Specifies the name of a transfer request in a group of interrelated transfer requests.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCNET

Defines the name of a group of interdependent transfer requests.

xxxxxxxx

Specifies up to eight alphanumeric characters representing the name of the XTC network running this transfer.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

Chapter 3: The Command Line Interface

The command line interface allows you to enter commands from the system prompt to control CA XCOM Data Transport.

Make sure you are familiar with parameter usage before using the commands in this chapter; for more information, see the chapter "Operating Environment." The daemon must also be running before you can perform any transfers.

This section contains the following topics:

Using Configuration Files (see page 142)
Symbolic Parameters in Configuration Files (see page 144)
How to Use Symbolic Parameters (see page 146)
Parameters Supporting Symbolic Variables (see page 148)
Transferring Files (see page 148)
Transferring Files Using SNA/APPC Protocols (see page 152)
Transferring Files Using TCP/IP Protocols (see page 154)
Setting Data Transfer Attribute Parameters (see page 158)
Using the Send File Command (see page 161)
Creating a File on an IBM Mainframe (see page 166)
Using the Retrieve File Command (see page 181)
Using the Send Report Command (see page 185)
Using the Send Job Command (see page 192)
How to Send a Metatransfer (see page 193)
How to Inquire on the Status of Metatransfers (see page 196)
How to Retrieve History Records (see page 199)
Command Examples for Metatransfers and Inquire on Status (see page 219)
How to Send a Ping Request (see page 220)
Tracing Problems (see page 224)
Setting Up Log Files (see page 226)
Setting File Type Conversion (see page 228)
Creating Custom Character Sets for File Conversion (see page 230)
Unicode and Multi-Byte Character Sets Support for Data Transfer (see page 234)
Run Reports on History Records (see page 238)
Using Store and Forward (see page 242)

Using Configuration Files

Use configuration files to set default parameters for CA XCOM Data Transport transfers. Use xcom.cnf to change default settings or to set transfer-specific variables for an individual locally-initiated transfer. You may have any number of different .cnf files with different names.

CA XCOM Data Transport reads the .cnf parameter file specified, or it reads $XCOM_HOME/config/xcom.cnf if no configuration file is specified. It then reads any parameters specified on the command line, and then executes the specified file transfer. CA XCOM Data Transport will also automatically use the defaults from the xcom.glb file to supply any parameter values not specified elsewhere. For each transfer, CA XCOM Data Transport compiles a full set of parameter values. To collect these values, the program follows a search sequence. The values are collected in the following order:

1. The CA XCOM Data Transport program defaults.
2. Parameters defined in xcom.glb at the time the CA XCOM Data Transport daemon was started.
3. Parameters defined in the specified .cnf file.

   If a copy of the configuration file is not found in the current directory, then the file $XCOM_HOME/config/xcom.cnf is used. If a .cnf file is specified, the parameters are not picked up from the xcom.cnf.
4. Parameters defined in the command line following the -f option.

Note: Use a text editor to modify the values in xcom.cnf. All modifications go into effect the next time a transfer is initiated.

For a description of how parameter values take precedence, see Using CA XCOM Parameters in the chapter "Operating Environment." For a complete list of CA XCOM Data Transport parameters see the appendix "Parameters."

Creating Configuration Files

To create configuration files of your own for different types of transfers

1. From the command line, copy xcom.cnf to a new file. Enter the following:
2. cp $XCOM_HOME/config/xcom.cnf *configfilename.cnf*

   The file xcom.cnf is renamed to *configfilename.cnf* and resides in your local directory

   Note: The file name *configfilename.cnf* is used only as an example here.
3. From your local directory, load *configfilename.cnf* into vi or another editor.

   The *configfilename.cnf* file is loaded in the editor.
4. Modify the parameters in the new file to your specific needs.

   It is not necessary to list all available parameters in a configuration file, because any unspecified parameters will use the defaults that are set in xcom.glb or the CA XCOM Data Transport daemon.

   If you specify a configuration file, then xcom.cnf is not used. If you do not specify a configuration file, then CA XCOM Data Transport looks in xcom.cnf for parameter values.
5. Save the file and exit the editor.

Example 1

In the following example, CA XCOM Data Transport is invoked from the command line and a configuration file is specified.

```
xcom62 [ -cnumber] [ -f [configfilename]

[PARAMETER_NAME=value ...] ]
```

Replace *number*, *configfilename*, and *PARAMETER_NAME=value* with the desired transfer value, configuration file name and/or parameter name and values.

For the complete syntax and options for using xcom62 or xcomtcp, see Transferring Files in this chapter.

Example 2

In the following example, a regional office transfers its accounting records daily to the mainframe headquarters.

For this transfer, create a configuration file named *account.cnf* that contains the following parameter values:

```
FILEOPTION=REPLACE
LOCAL_FILE=/usr/bills/july
```

```
REMOTE_FILE=BILLS.JULY
```

Then specify this configuration file to use for the transfer as follows:

```
xcom62 -c1 -f account.cnf
```

Symbolic Parameters in Configuration Files

Symbolic parameters let you store transfer parameters in configuration files with variable data that is resolved to other values at schedule time.

Standard symbolic parameters are supplied with CA XCOM Data Transport.

No setup is required to use the pre-defined variables that are integrated as part of CA XCOM Data Transport. If you place these variables in the parameter data set for a transfer, CA XCOM Data Transport invokes them when scheduling the transfer.

&DATE(format-code)

Causes the current date to be substituted dynamically in the current keyword value. The format of the date depends on the format code that is selected as a sub-parameter. If no format is specified, MMDDYYYY is used. Valid format-codes and examples of their output are:

- *MMDDYYYY*—12312012
- *DDMMYYYY*—31122012
- *YYYYMMDD*—20121231
- *YYMMDD*—121231
- *YYYY*—2012
- *MM*—12
- *YY*—12
- *DD*—31
- *YYDDD*—12366
- *YYYYDDD*—2012366
- *DDMONYYYY*—31DEC2012
- *MON*—DEC
- *MONTH*—DECEMBER

&ID

Causes the value entered for ID to be substituted dynamically in the current keyword value.

&IPNAME

Causes the value entered for IPNAME to be substituted dynamically in the current keyword value.

&LU

Causes the value entered for LU to be substituted dynamically in the current keyword value.

&LUSER

Causes the current local user ID (or the USERID of the current job) to be substituted dynamically in the current keyword value.

&TIME(format-code)

Causes the current time to be substituted dynamically in the current keyword value. The format of the time depends on the format code that is selected as a sub-parameter. Valid format-codes and examples of their output are:

- *HHMMSSTH*—15312811
- *HHMMSS*—153128
- *HHMM—1531*
- *MMSS*—3128
- *HH*—15
- *MM*—31
- *SS*—28
- *TH*—11
- *TH3*—110

&USERID

Causes the current remote user ID (or the USERID of the current job) to be substituted dynamically in the current keyword value.

How to Use Symbolic Parameters

You can place symbolic variables in any combination in SYSIN01 control statements to create composite parameter values. You can use a period (.) as a terminating character for the symbolic variable, but this is not required. If a period is present, it is removed from the resultant field content.

Example

```
REMOTE_FILE=C:\REMOTE_FILE-&TIME(HH)-&TIME(MM)-&TIME(SS)
```

Important! The symbolic variable expression cannot exceed the maximum length allowed by the parameter. In the case of REMOTE_FILE, the maximum length is 256 characters.

This results in the value for REMOTE_FILE being set to:

```
C:\REMOTE_FILE-15-31-28
```

Important! If you specify a symbolic parameter on the command line, you need to enclose it in double quotes, so that Windows does not interpret the ampersand character (&) as a special command line character. So this example would become *one* of the following:

- REMOTE_FILE=C:\REMOTE_FILE-"&TIME(HH)"-"&TIME(MM)"-"&TIME(SS)"
- REMOTE_FILE=”C:\REMOTE_FILE-&TIME(HH)-&TIME(MM)-&TIME(SS)"

If you specify a symbolic parameter in a configuration file, there is no need for double quotes.

Assigning Values

If the symbolic variable does not have a system value, you must assign a value before using the symbolic variable. You can also specify subscripts on symbolic variables that do not have a format-code. There are two formats for subscripted symbolic variables. If only one numeric value is present, a starting position of 1 is assumed. A second subscript format allows for a starting position as well as a length to be entered:

Example (one subscript)

```
LU=L784000

ID=LU#&LU(4)
```

This results in the value for ID being set to:

```
LU#L784
```

Example (two subscripts)

```
LU=LU250021

ID=LU&LU(5,4)
```

This results in the value for ID being set to:

```
LU0021
```

Parameters Supporting Symbolic Variables

You can use symbolic parameters with the following parameters:

- DESTINATION
- FORM
- LOCAL_FILE, LOCAL_FILE_RF, LOCAL_FILE_SJ, LOCAL_FILE_SR, LOCAL_NOTIFY, LUSER
- NOTIFY_NAME
- OEDATE, OETIME, OFLMAX, OFLMIN, OID, OLMSG, OREQ, OSDATE, OSTIME, OTNAME, OUSER
- REMOTE_FILE, REMOTE_FILE_RF, REMOTE_SYSTEM, REMOTE_SYSTEM_RF, REMOTE_SYSTEM_SJ, REMOTE_SYSTEM_SR, REPORT_TITLE
- TRANSFER_ID, TRANSFER_USR_DATA
- UNIT, UNIT_RF, USER_DATA, USERID
- VOLUME, VOLUME_RF
- XCOM_CONFIG_SSL, XTCERRDECR, XTCERRINCR, XTCERRPURGE, XTCERRREL, XTCGOODDECR, XTCGOODINCR, XTCGOODPURGE, XTCGOODREL, XTCJOB, XTCNET

Transferring Files

This section contains information about performing file transfers from the command line using xcom62 or xcomtcp. It describes the syntax and options for these commands, and also contains information about multiple transfers, using semicolons, and wildcard characters.

- Use xcom62 or xcomtcp to initiate file transfers from the command line.
- Use xcom62 for transfers that use SNA/APPC protocols.
- Use xcomtcp for transfers that use TCP/IP protocols.

Specifying Protocols

When using the xcom62 or xcomtcp commands, the choice of protocol to use is indicated by the PROTOCOL parameter. This can be specified at the command line, in a configuration file, or in the xcom.glb file, depending upon your installation's needs. If the protocol is not specified at the command line, the defaults specified in the configuration file or in xcom.glb are used.

Queuing Transfers

Transfers can be queued or not queued. When queued, failed transfers are retried automatically, depending on the parameters set in the xcom.glb. Multiple transfers can be sent to a partner simultaneously using SNA or TCP/IP.

When not queued, transfers execute immediately, single-threaded and are not retried if they fail.

Using xcom62

For performing transfers using SNA/APPC protocols, xcom62 allows you to initiate the transfer of a file, a job or a report from the command line, and to specify the parameters to be used for that transfer.

Using xcomtcp

For performing transfers using TCP/IP, xcomtcp allows you to initiate the transfer of a file, a job or a report from the command prompt, and to specify the parameters to be used for that transfer

Multiple Transfers

You may initiate multiple transfers in two ways. You can use the NEWXFER option of the CONTROL parameter or you can use separate commands from the command line by placing a semicolon in between each transfer request. For information on the CONTROL parameter, see the appendix "Parameters."

Using Semicolons

You can do several transfers from the command line by using the semicolon (;), the general shell command separator.

Example

In the following example, two files are sent to the remote system, using semicolons to separate the transfers. The file *test1* is sent into file *testA* and file *test2* is sent into file *testB*.

```
xcom62 -c1 -f LOCAL_FILE=test1 REMOTE_FILE=testA; xcom62 -c1
-f LOCAL_FILE=test2 REMOTE_FILE=testB
```

Wildcard Characters

CA XCOM Data Transport for UNIX and Linux supports the use of wildcards for local files and remote files.

Syntax

The syntax for using xcom62 and xcomtcp is identical, except for the choice of command. The syntax for each is shown below.

Syntax for xcom62

The syntax for using xcom62 is as follows:

```
xcom62 [-cnumber] [option] [-f [configfilename]
[PARAMETER_NAME=value ...] ]
```

Syntax for xcomtcp

The syntax for using xcomtcp is as follows:

```
xcomtcp [-cnumber] [option] [-f [configfilename]
[PARAMETER_NAME=value ...] ]
```

Options

The following list explains the options available for xcom62 and xcomtcp:

Note: When the shell encounters an option that performs an action and exits xcom62 or xcomtcp, it will perform the command and exit xcom62 or xcomtcp without reading the rest of the command line.

-c*number*

The type of transfer to be attempted. Valid values for number are:

1—Send file to remote system (default).

2—Send report to be printed on remote system.

3—Send job to be executed on remote system.

4—Receive file from remote system.

5—Metatransfer to a remote system.

6—Metatransfer to inquire on the status of metatransfers.

7—Metatransfer to retrieve history records.

-ping

CA XCOM Data Transport tests reachability of the remote CA XCOM Data Transport Server and displays information about CA XCOM Data Transport Server.

Note: All other options are ignored.

-h

CA XCOM Data Transport displays help. The usage message is displayed and exits xcom62 or xcomtcp.

Note: All other options are ignored.

-r

CA XCOM Data Transport displays the release level. The release level of the software is displayed and exits xcom62 or xcomtcp.

-q

CA XCOM Data Transport queues the transfer. This option works like the QUEUE=YES parameter in xcom.cnf.

-s

CA XCOM Data Transport sets silent mode, which turns off all output to stderr.

-t

CA XCOM Data Transport sends trace output to stderr.

-i

CA XCOM Data Transport ignores configuration errors.

-f *configfilename*

CA XCOM Data Transport reads *configfilename* to set the parameters necessary for the transfer. The syntax for setting parameters within the configuration file is as follows:

```
PARAMETER1=value
PARAMETER2=value
```

Note: Though you can have CA XCOM Data Transport read parameters from a file as well as from the command line, the parameters on the command line take precedence and must come after the file name.

If you specify -f, you must specify *configfilename* or PARAMETERNAME=value, or you can specify both.

PARAMETERNAME=*value...*

Instead of setting parameters in a file, you can set them on the command line. The syntax is as follows:

```
PARAMETER1=value PARAMETER2=value
```

The value of the parameters must be in uppercase where appropriate, but the parameter names themselves, with an underscore character where indicated, are not required to be in uppercase.

Note: Parameters entered on the command line take precedence over those in the parameter file.

no options

If you invoke CA XCOM Data Transport without options, the defaults that are in xcom.cnf are used for the transfer.

-v

Sets the transfer version type. Valid values are 1 or 2.

-x

Sets the trace level. Valid values are 0 to 10.

Transferring Files Using SNA/APPC Protocols

Use the information below as examples when performing transfers that use SNA/APPC protocols.

Note: The syntax and options for using xcom62 are identical to those for xcomtcp. For more information, see Transferring Files in this chapter.

Using SNA/APPC Protocols

Your computer and the remote CA XCOM Data Transport system must be configured for the appropriate SNA/APPC for you to use SNA/APPC protocols with CA XCOM Data Transport.

The choice of protocol to use is indicated by the PROTOCOL parameter. This can be specified at the command line, in a configuration file, or in the xcom.glb file, depending upon your installation's needs. If the protocol is not specified at the command line, the defaults specified in the configuration file or in xcom.glb are used.

For more information about your platform's SNA/APPC configuration, see the *CA XCOM Data Transport Installation Guide for UNIX and Linux* and the SNA/APPC vendor documentation.

Example 1

In the following example, the xcom62 command is used to queue a file for a send file transfer, using SNA/APPC protocols. The configuration file is *myconfig.cnf*, the local file is named *testa.aaa*, and the remote file is named *testb.bbb*. The remote system is indicated by the value *XCOMDEVT2* in the REMOTE_SYSTEM parameter. QUEUE is set to YES, so the transfer is queued. The protocol is determined by the value of the PROTOCOL parameter in the xcom.glb file, which should be set to SNA for an SNA/APPC transfer.

```
xcom62 -c1 -f myconfig.cnf LOCAL_FILE=testa.aaa
REMOTE_FILE=testb.bbb REMOTE_SYSTEM=XCOMDEVT2
QUEUE=YES
```

Example 2

In the following example, the xcom62 command is used to send a file directly, without sending it to the queue, using SNA/APPC protocols. The configuration file is *myconfig.cnf*, the local file is named *testa.aaa*, and the remote file is named *testb.bbb*. The remote system is indicated by the value *XCOMDEVT2* in the REMOTE_SYSTEM parameter. QUEUE is set to NO, so the transfer starts immediately after you press Enter. The choice of protocols is indicated by the value *SNA* in the PROTOCOL parameter. If the default value in the configuration file or the xcom.glb file is SNA, you do not need to specify this value here.

```
xcom62 -c1 -f myconfig.cnf LOCAL_FILE=testa.aaa
REMOTE_FILE=testb.bbb REMOTE_SYSTEM=XCOMDEVT2
QUEUE=NO PROTOCOL=SNA
```

Transferring Files Using TCP/IP Protocols

Use IP addresses, host names, or domain names when performing transfers that use TCP/IP protocols.

Note: The syntax and options for using xcomtcp are identical to those for xcom62. For more information, see Transferring Files in this chapter.

Using IP Addresses and Names

Your system and the remote CA XCOM Data Transport system must be configured for TCP/IP for you to use TCP/IP protocols with CA XCOM Data Transport.

Before performing a file transfer, you must know the IP address, the host name, or the domain name of the remote system. Check with the network administrator of the remote system for these values.

If your system is not configured for TCP/IP, check with your network administrator for further information.

The formats of the IP address, host name, and domain name are as follows:

IP address

A unique number for a particular system, used to identify the system on the TCP/IP network. IP addresses are in the dotted decimal notation format.

Example:

123.123.78.91

Host Name

The host name of a particular system.

Example:

goodsys

Domain Name

The Domain Name Service (DNS) name. Identifies the system's group in the DNS hierarchy. The host name and the domain name make up the fully qualified domain name of the system.

Example:

goodsys.goodsite.com

goodsys is the name of the computer, and it is in the goodsite.com domain.

Specifying the Remote System

When using TCP/IP, the remote system can be specified in different ways. For example, you can use the following forms:

By host name:

REMOTE_SYSTEM=goodsys

By fully qualified domain name:

REMOTE_SYSTEM=goodsys.goodsite.com

By IP address:

REMOTE_SYSTEM=123.123.78.91

Using the IP address is the most efficient method to use when specifying a remote system location.

Note: The examples above use the REMOTE_SYSTEM parameter, but the usage also applies to REMOTE_SYSTEM_RF, REMOTE_SYSTEM_SJ, and REMOTE_SYSTEM_SR.

TCP/IP Name Resolution

If using a host name or domain name, your system must have a way to resolve that name to an IP address.

You can use any symbolic name that can be mapped to an IP address, such as a host name or a domain name. Your system must be set up to resolve the name to an IP address. Check with your network administrator for further information about your system's use of names.

If you are relying on name resolution to resolve names to IP addresses, this capability must be installed and configured on your system. There are many ways to implement name resolution, including the Domain Name Service (DNS), and the use of host files.

Using TCP/IP Protocols with CA XCOM Data Transport

To use TCP/IP protocols for a CA XCOM Data Transport transfer, CA XCOM Data Transport must have the port specification for the remote system and an indication that you wish to use TCP/IP protocols.

The port is specified by the PORT parameter. The default value in the xcom.glb file should be valid for most remote hosts. However, when executing transfers that specify an IPv6 remote address and/or the parameter SECURE_SOCKET=YES for secure transfers using OpenSSL, then the PORT parameter value would need to be changed accordingly. For more information about configuring CA XCOM Data Transport for secure transfers, see the chapter “Generate SSL Certificates” later in this guide.

If you need to change the port value of the local system, see the *CA XCOM Data Transport Installation Guide for UNIX and Linux*.

The choice of protocol to use is indicated by the PROTOCOL parameter. This can be specified at the command line, in a configuration file, or in the xcom.glb file, depending upon your installation's needs. If the protocol is not specified at the command line, the defaults specified in the configuration file or in xcom.glb are used.

In addition, CA XCOM Data Transport uses other parameters for TCP/IP functionality. These parameters are of interest to system administrators for tuning and performance considerations. They are described in the section on xcom.glb parameters in the chapter "Operating Environment" and are also listed in the appendix "Parameters."

Example 1:

In this example, the xcomtcp command is used to queue a file for a send file transfer, using TCP/IP. The configuration file is *myconfig.cnf*, the local file is named *testa.aaa*, and the remote file is named *testb.bbb*. The remote system is indicated by the host name *goodsys* in the REMOTE_SYSTEM parameter. QUEUE is set to YES, so the transfer is queued. TCP/IP is indicated by the value *TCPIP* in the PROTOCOL parameter. The port is determined by the value of the PORT parameter in the xcom.glb file.

```
xcomtcp -c1 -f myconfig.cnf LOCAL_FILE=testa.aaa
REMOTE_FILE=testb.bbb REMOTE_SYSTEM=goodsys
QUEUE=YES PROTOCOL=TCPIP
```

Example 2:

In this example, the xcomtcp command is used to submit a transfer directly, without sending it to the queue, using TCP/IP. The configuration file is *myconfig.cnf*, the local file is named *testa.aaa*, and the remote file is named *testb.bbb*. The remote system is indicated by the fully qualified domain name *goodsys.goodsite.com* in the REMOTE_SYSTEM parameter. QUEUE is set to NO, so the transfer starts immediately after you press Enter. TCP/IP is indicated by the value *TCPIP* in the PROTOCOL parameter. (If the default value in the configuration file or the xcom.glb file is TCPIP, you do not need to specify this value here.) The port is indicated by the value 8044 in the PORT parameter.

Note: The default value of PORT should be valid for most hosts, and you should only specify a value for this parameter when needed.

```
xcomtcp -c1 -f myconfig.cnf LOCAL_FILE=testa.aaa
REMOTE_FILE=testb.bbb REMOTE_SYSTEM=goodsys.goodsite.com
QUEUE=NO PROTOCOL=TCPIP PORT=8044
```

PORT

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 1 to 65535

Default: 8044

PROTOCOL

The type of communication protocols to use.

SNA

For transfers using SNA/APPC communication protocols

TCPIP

For transfers using TCP/IP communication protocols

Default: SNA

Note: If only the TCP/IP component is installed with the base components of CA XCOM Data Transport, the default value is TCPIP.

SECURE_SOCKET

This parameter specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and must connect to an SSL listener on the remote partner.

NO

Performs a non-secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

XCOM_CONFIG_SSL

This parameter specifies the configssl.cnf file path and file name.

Range: 1 to 256 characters

Default: $XCOM_HOME%/config/configssl.cnf

Note:

- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.
- $XCOM_HOME is an environment variables.

Setting Data Transfer Attribute Parameters

Use the following parameters in the xcom.cnf file or in any CA XCOM Data Transport configuration files to configure CA XCOM Data Transport data transfer attribute parameters. These parameters do not change for the different types of file transfers. We recommend modifying these values in xcom.cnf during the initial configuration of CA XCOM Data Transport.

List of Parameters

The data transfer attribute configuration parameters are shown in this section.

COMPRESS

Indicates the compression type. When communicating with an IBM mainframe, if the data file contains any empty lines, COMPRESS can only be set to YES, LZSMALL, LZMEDIUM, or LZLARGE.

Notes:

- Not all compression types are supported on all platforms. For supported compression types, see the partner platform documentation.
- The LZ values enable LZ (Lempel-Ziv) compression to replace sequences of data bytes that occur more than once in a data stream with a code value.

COMPACT

RJE compaction algorithm optimized for uppercase English text.

LCOMPACT

RJE compaction algorithm optimized for lowercase English text.

LZLARGE

Activates LZ compression to search back 32K in the datastream for a matching string.

LZMEDIUM

Activates LZ compression to search back 16K in the datastream for a matching string.

LZRW3

General-purpose algorithm that runs fast and gives reasonable compression.

LZSMALL

Activates LZ compression to search back 4K in the datastream for a matching string.

NO

Indicates no compression.

RLE

Run length encoding.

YES

Indicates run length encoding of binary zeros and blanks only.

ZLIB*n*

Greater compression than LZRW3 but less than LZSMALL, LZMEDIUM, and LZLARGE. The *n* value can be 1 through 9.

Default: YES

MAXRECLEN

For Windows, UNIX, and Linux systems, the locally initiating CA XCOM Data Transport system determines the values for MAXRECLEN, TRUNCATION, and LRECL, for send and receive operations. When the local CA XCOM Data Transport system initiates a transfer of a text file, this parameter designates the length, in bytes, of the largest record that can be transferred. If a record length is longer than this value, CA XCOM Data Transport uses the value in the TRUNCATION parameter on the initiating side to determine whether to terminate the transfer or to truncate the record and continue the transfer. When CA XCOM Data Transport transfers binary files, this value indicates the length of the records that are transferred. On a receive operation, MAXRECLEN is set to whatever the LRECL value is on the initiating side.

Range: 1 to 32767

Default: 1024

TRUNCATION

Indicates whether CA XCOM Data Transport truncates excess characters in the source file if the record exceeds the maximum record length as indicated by the MAXRECLEN parameter. If NO is selected, and the maximum record length is exceeded, CA XCOM Data Transport aborts the transfer. This parameter is ignored if CARRIAGE_FLAG=NO.

Note: Truncation is not valid for BINARY data or for non-text data received on the UNIX or Linux platform.

Range: YES or NO

Default: NO

VERSION

Indicates the version of the CA XCOM Data Transport protocol to be used for this transfer. For TCP/IP, only a value of 2 is valid.

Range: 1 or 2

Default: 2

Note: This is a Version 2 parameter.

Using the Send File Command

Use the Send File command to send a copy of a file on the local system to a file on a remote system.

The -c1 option specifies that this is a Send File transfer. The local file is indicated by the LOCAL_FILE parameter. The file on the remote system is indicated by the REMOTE_FILE parameter.

Example 1

In the following example, the xcom62 command is used. The configuration file is */myconfig.cnf*, the local file is named */testa.aaa*, the remote file is named */testb.bbb*, and the transferred file replaces a file on the remote system that already exists. The PROTOCOL parameter specifies the protocol as SNA. All other necessary parameters are read from the default configuration file, XCOM.CNF.

```
xcom62 -c1 -f /myconfig.cnf LOCAL_FILE=/testa.aaa
REMOTE_FILE=/testb.bbb FILE_OPTION=REPLACE PROTOCOL=SNA
```

Example 2

In the following example, the xcomtcp command is used. The configuration file is */myconfig.cnf*, the local file is named */testa.aaa*, the remote file is named */testb.bbb*, and the transferred file replaces a file on the remote system that already exists. The PROTOCOL parameter specifies the protocol as TCP/IP. All other necessary parameters are read from the default configuration file, XCOM.CNF.

```
xcomtcp -c1 -f /myconfig.cnf LOCAL_FILE=/testa.aaa
REMOTE_FILE=/testb.bbb FILE_OPTION=REPLACE
PROTOCOL=TCPIP
```

Parameters

The Send File parameters are shown in this section.

CREATE_DIR

Indicates whether to create the specified directory if it does not exist.

YES

Create the directory if it does not exist.

NO

Do not create the directory if it does not exist.

Default: YES

FILE_OPTION

Indicates how the transferred data is to be processed by the receiving system. For file transfers only.

For most file transfers, the parameter values are as follows:

CREATE

Create a new file on the receiving system.

APPEND

Append the transferred data to an existing file on the receiving system.

REPLACE

Replace an existing file on the receiving system.

For wildcard transfers, the parameter values are as follows:

CREATE

Create the PDS/Directory and add the transferred members. If the PDS/Directory already exists, the transfer fails with an error.

APPEND

Add transferred members/files. If the PDS/Directory does not exist or the member/file already exists, the transfer fails with an error.

REPLACE

Add or replace transferred members/files. If the PDS/Directory does not exist, the transfer fails with error XCOMN0403E Cannot open output file—No such file or directory.

Note: When creating a file on an IBM mainframe system, some additional information may be necessary. For more information, see the explanations for RECORD_FORMAT, BLKSIZE, VOLUME, and UNIT parameters.

Default: CREATE

LOCAL_FILE

The name of the file on the local system that is being transferred. At the command prompt or in a script, if this variable is null or unset, standard input is read. In this manner, CA XCOM Data Transport commands can be used in a pipeline or with redirection. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk (*) as a file name to indicate that all files within the specified directory are to be transferred. For example, the statement LOCAL_FILE=/NAMES/* indicates that all files under the NAMES directory are to be transferred.

When a prefix is followed by an asterisk, all members beginning with a specific prefix are to be transferred. For example, LOCAL_FILE=/NAMES/AL* requests that files AL, ALEX, and ALICE are all to be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated. However, when sending files to a mainframe PDS, any file name over 8 characters in length is truncated. These systems do not recognize file extensions. For example, a file called *longfilename.txt* will be truncated to *longfile* and a file called *file.txt* will be truncated to *file*.

Note: If QUEUE=YES, the full path name must be specified.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_FILE

Indicates the file on the remote computer to which the transferred data is being written. If you are creating the file (FILE_OPTION=CREATE), the file name must be consistent with the file naming conventions of the remote system. The local CA XCOM Data Transport system does not validate this name. The remote I/O system determines whether the file name is valid.

For wildcard transfers, use an asterisk (*) as a file name to indicate and to inform the receiving partner that multiple files will be sent.

Example

```
REMOTE_FILE=/PAYROLL/*.
```

If multiple files are sent and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

An asterisk used to send files to an IBM mainframe system shows that all files are to be transferred to a partitioned data set (PDS).

For platforms that support it, you can specify a common file extension to append to each file name.

Example

```
REMOTE_FILE=/PAYROLL/*.TXT.
```

Note: For send file transfers only.

Range: 1 to 256 characters

Default: None

REMOTE_SYSTEM

The name of the remote system that receives a file, job, report, or ping request.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, the name is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote systems IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

Creating a File on an IBM Mainframe

The IBM mainframe file system, unlike the UNIX or Linux system, may require that you specify a variety of attributes in order to create a file on the mainframe. To determine the appropriate values for these parameters, check with your IBM mainframe applications group.

Parameters

The following parameters are used when FILE_OPTION=CREATE when creating a file on an IBM mainframe from a local system.

ALLOCATION_TYPE

This parameter indicates the unit of storage allocation for a data set created on an IBM mainframe.

CYL

Cylinders

TRK

Tracks

BLK

Blocks

REC

Records

Default: CYL

Note: This parameter is a Version 2 parameter.

AVGREC

For a data set created on an IBM mainframe, Specifies the multiplier for Primary and Secondary allocation units when allocating based on the number of records. The record size is based on the value of the LRECL parameter.

U

Indicates that the PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records to allocate for.

K

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in thousands (so it would be the number specified multiplied by 1024).

For example, specifying 3 would be stating 3 K or 3072 records.

M

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in millions (so it would be the number specified multiplied by 1048576).

For example, specifying 2 would be stating 2M or 2097152 records.

Default: None

Note: This parameter applies only when the SPACE parameter specifies a value of “REC”. The parameter indicates that a file is being allocated based on a specific number of records.

BLKSIZE

Specifies the block size of a data set created on an IBM mainframe.

Note: Used when FILE_OPTION=CREATE

Range: 0 to 32767

If the format is…	Then the block size must be…
Fixed or fixed block record	A multiple of the record length
Variable record	Four bytes larger than the record length
Undefined record	Larger than the largest record length

Default: 800

COMPRESS_PDS

The compress_pds parameter controls when a PDS dataset gets compressed for CA XCOM Data Transport z/OS partners that support this functionality. The value of the CMPRS_PDS_ALLOW parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMDFLT) determines if PDS compression is allowed. When this parameter is set to YES, then PDS compression can occur on the CA XCOM Data Transport z/OS partner.

NONE

Suppresses the compression of an output PDS dataset as part of a CA XCOM Data Transport transfer.

BEFORE

The output PDS dataset is compressed before the transfer of user data begins.

AFTER

The output PDS dataset is compressed after the transfer of user data has completed.

BOTH

The output PDS dataset is compressed both before and after the transfer of user data.

Default: NONE

Note: COMPRESS_PDS applies only to PDS data sets that are opened for output as the target of a CA XCOM Data Transport transfer.

CREATEDELETE

Specifies whether an existing data set can be deleted and a new data set allocated at the start of a FILEOPT=CREATE transfer. This functionality is for CA XCOM Data Transport z/OS partners with support for CREATEDELETE. The CREATEDELETE parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMCNTL) specified by the z/OS CA XCOM Data Transport Administrator affects the functionality of this parameter.

When z/OS has CREATEDELETE=ALLOW:

YES

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

NO

If FILEOPT=CREATE and the data set exists, then the transfer fail with a catalog/file error.

When z/OS has CREATEDELETE=YES:

YES or NO

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

When z/OS has CREATEDELETE=NO:

YES or NO

If FILEOPT=CREATE and the data set exists, then the transfer fails with a catalog/file error.

Default: NO

Notes:

- The attributes of the existing dataset are deleted and the new data set is allocated with the attributes specified in the transfer when CREATEDELETE=YES.
- CREATEDELETE applies only if the target data set is a sequential data set or an entire PDS/PDSE. CREATEDELETE is ignored for other types of data sets (such as PDS members, PDSE members, VSAM, and USS files).
- CREATEDELETE applies to relative GDGs when the dataset is specified using the fully qualified GxxxxVxx name.

DATACLAS

Specifies the name of the data class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

DSNTYPE

This parameter specifies the data set definition.

Note: This parameter applies only to mainframe SMS data sets.

LIBRARY

Defines a PDSE.

PDS

Defines a partitioned data set.

GATEWAY

Defines a Gateway file.

BASIC

Defines a legacy sequential dataset.

LARGE

Defines a large format sequential dataset.

EXTREQ

Defines an extended format dataset.

EXTPREF

Specifies an extended format is preferred. If the extended format is not possible, a basic format is used.

<blank>

Defines a partitioned or sequential data set based on the data set characteristics entered.

Note: These values are IBM standards for SMS processing.

Range: One to eight characters

Default: None

EATTR

This parameter identifies if the dataset can have extended attributes when the dataset is allocated on an Extended Address Volume (EAV).

OPT

Specifies that a dataset can optionally have extended attributes.

NO

Specifies that a dataset cannot have extended attributes.

Default: None

Note: This parameter is applicable only for data set creation on an IBM mainframe.

LRECL

Specifies the actual or maximum length, in bytes, of a logical record. Corresponds to the JCL LRECL subparameter.

Range: 0 to 32767

If the format is…	Then the maximum length of a logical record must be equal to the…
Variable blocked record	Maximum record length plus four
Fixed or fixed block record	Constant record length

Default: 160

MGMTCLAS

Specifies the name of the management class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

NUM_OF_DIR_BLOCKS

This parameter specifies the number of directory blocks to allocate for a data set created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

PRIMARY_ALLOC

This parameter identifies primary storage allocation for a data set created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 1

Note: This parameter is a Version 2 parameter.

RECORD_FORMAT

Specifies the record format of a data set created on an IBM mainframe. This corresponds to the JCL RECFM subparameter.

Range: The range of values is listed in the following table:

Value	Description	Record Length	Comment
F	Fixed unblocked	The same length as the data set	
FA	Fixed unblocked ANSI	The same length as the data set	Contains ISO/ANSI/FIPS control characters
FB	Fixed blocked	Fixed	Fixed record length with multiple records per block
FBA	Fixed blocked ANSI	Fixed	Multiple records per block where these records contain ISO/ANSI/FIPS control characters
FBS	Fixed blocked spanned	Fixed	Multiple records per block written as standard blocks

Value	Description	Record Length	Comment
FM	Fixed unblocked machine	The same length as the data set	Contains machine code control characters
FS	Fixed unblocked spanned	The same length as the data set	Written as standard blocks where these records do not contain any truncated blocks or unfilled tracks
U	Undefined	Undefined	
V	Variable unblocked	Variable	
VA	Variable unblocked ANSI	Variable	Contains ISO/ANSI/FIPS control characters
VB	Variable blocked	Variable	Multiple records per block
VBA	Variable blocked ANSI	Variable	Multiple records per block where these records contain ISO/ANSI/FIPS control characters
VBM	Variable blocked machine	Variable	Multiple records per block where these records contain machine code control characters
VBS	Variable blocked spanned	Variable	May have multiple records per block where these records can span more than one block
VM	Variable unblocked machine	Variable	Contains machine code control characters
VS	Variable unblocked spanned	Variable	A record can span more than one block

Default: VB

SECONDARY_ALLOC

This parameter identifies secondary storage allocation for a data set created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

SPACE

The unit of storage allocation for the remote file.

CYL

Cylinders

TRK

Tracks

BLK

Blocks

REC

Records

Specify by:

- Primary allocation space for the remote file
- Secondary allocation for the remote file
- Directory blocks for the remote file

Default: CYL

STORCLAS

Specifies the name of the storage class for a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

UNIT

Specifies the unit on which a data set is to be created on an IBM mainframe.

Note: Used when FILE_OPTION=CREATE.

Range: Zero to six characters

Default: None

VOLUME

Specifies the volume on which a data set is to be created on an IBM mainframe.

Range: Zero to six characters

Default: None

Tape Parameters for an IBM Mainframe

Use the following parameters at the command prompt and in configuration files when IBM Mainframe tape drives are involved in the transfer. Allowable values for these parameters are the same as for their IBM JCL counterparts except where noted. You can also refer to this section for parameter information when using the xcomtool.

Note: When tape drives are involved in the transfer, it may be necessary to also use some of the parameters discussed in the previous section, Creating a File on an IBM Mainframe, such as UNIT and VOLUME.

DEN

Specifies the density to be used in creating a tape on the remote system. Valid values are the same as those for the DEN parameter in JCL.

Range: 1 to 4

Default: None

EXPDT

Specifies the expiration date to be placed on the tape data set being created.

yyddd

Specifies an expiration date for the tape data set as a two-digit designation for the year and a three-digit designation for the day of the year. For example, in the expiration date 11021, 11 is the year (namely, 2011) and 021 is the 21st day of that year. The tape data set will expire on January 21, 2011.

yyyy/ddd

Specifies an expiration date for the tape data set in terms of a four-digit designation for the year and a three-digit designation for the day of the year. For example, in the expiration date 2011/021, 2011 is the year and 021 is the 21st day of that year. The tape data set will expire on January 21, 2011.

Note: EXPDT and RETPD are mutually exclusive; specify one or the other.

LABEL

Indicates the type of label associated with a tape data set.

Processing type: AL, AUL, BLP, LTM, NL, NSL, SL, SUL

Represents the type of processing to be applied to data sets on tape.

Note: CA XCOM Data Transport for z/OS supports only standard label tapes.

Example:

LABEL=BLP

The type of processing to be applied to this data set is BLP.

Default: AL

LABELNUM

Indicates the sequence number of the data set on the tape.

Sequence number (0001 to 9999)

This value identifies the sequence number of a data set on tape.

Example:

LABELNUM=2

This specification refers to the second data set on the tape.

Default: 0001

RETPD

Specifies the number of days (1 to 9999) that the tape data set being created is to be retained.

Range: 1 to 9999

Default: None

Note: RETPD and EXPDT are mutually exclusive; specify one or the other.

TAPE

Indicates to the remote system whether the transfer is to a tape volume or a disk file.

YES

The transfer is to a tape volume. Mounts are allowed when performing dynamic allocation.

NO

The transfer is to a disk file.

Default: NO

TAPEDISP

Specifies the disposition value for MVS tape data sets.

1

New

2

Old

3

Mod

Default: 1

UNITCT

Specifies the number of units to be allocated on the remote system.

This is a tape parameter and is used when the partner is an IBM mainframe.

Range: 1 to 20

Default: None

VOLCT

Specifies the maximum number of volumes to be used in processing a multi-volume output tape data set on the remote system.

Range: 1 to 255

Default: None

VOLSQ

Specifies the sequence number of the first volume of a multi-volume remote data set to be used.

Range: 1 to 255

Default: None

Using the Retrieve File Command

Use the Retrieve File command to retrieve a copy of a file from a remote system and write it to a specified file on the local system.

The -c4 option specifies that this is a Retrieve File transfer. The remote file to be copied is indicated by the REMOTE_FILE_RF parameter. The file on the local system is indicated by the LOCAL_FILE_RF parameter.

Example 1

In the following example, the xcom62 command is used. The remote file *customer* is retrieved into the local file */accounts*. All other necessary parameters are read from the default configuration file, xcom.cnf.

```
xcom62 -c4 -f REMOTE_FILE_RF=customer LOCAL_FILE_RF=/accounts
```

Example 2

In the following example, the xcomtcp command is used. The remote file *customer* is retrieved into the local file */accounts*. All other necessary parameters are read from the default configuration file, xcom.cnf.

```
xcomtcp -c4 -f REMOTE_FILE_RF=customer LOCAL_FILE_RF=/accounts
```

Parameters

The Retrieve File parameters are shown below.

FILE_OPTION_RF

Indicates how the transferred data is to be processed by the receiving system (that is, the local system). Used when the transfer type is Retrieve File. If a value is not specified, then the value of FILE_OPTION is used. If no default is specified in FILE_OPTION or FILE_OPTION_RF, then the value defaults to CREATE.

For most file transfers, the parameter values are as follows:

CREATE

Create a new file on the receiving system.

APPEND

Append the transferred data to an existing file on the receiving system.

REPLACE

Replace an existing file on the receiving system.

For wildcard transfers, the parameter values are as follows:

CREATE

Create the PDS/Directory and add the transferred members. If the PDS/Directory already exists, the transfer fails with an error.

APPEND

Add transferred members/files. If the PDS/Directory does not exist or the member/file already exists, the transfer fails with an error.

REPLACE

Add or replace transferred members/files. If the PDS/Directory does not exist, the transfer fails with error XCOMN0403E Cannot open output file-No such file or directory.

Default: CREATE

LOCAL_FILE_RF

The file name that is created, appended, or replaced on the local system when it receives a file. At the command prompt or in a script, if this value is null or unset, then CA XCOM Data Transport writes to the stdout. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk (*) as a file name to indicate that multiple files will be received.

Example

```
LOCAL_FILE_RF=/PAYROLL/*.
```

If multiple files are received and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

For platforms that support it, you can specify a common file extension to be appended to each file name.

Example

```
LOCAL_FILE_RF=/PAYROLL/*.TXT.
```

Note: If QUEUE=YES, the full path name must be specified.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_FILE_RF

Indicates the name of the file to be retrieved from the remote system.

For wildcard transfers, use an asterisk (*) as a file name to indicate that all files within the specified PDS/Directory should be transferred. For example, the statement REMOTE_FILE_RF=/NAMES/* indicates that all files under the NAMES directory should be transferred.

When a prefix is followed by an asterisk, all members beginning with a specific prefix are transferred. For example, REMOTE_FILE_RF=/NAMES/AL* requests that files AL, ALEX, and ALICE should be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated.

Note: For retrieve file transfers only.

Range: 1 to 256 characters

Default: None

REMOTE_SYSTEM_RF

The name of the remote system that sends a file on a receive file operation. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

UNIT_RF

Specifies the unit from which a data set is to be retrieved from an IBM mainframe.

Range: Zero to six characters

Default: None

VOLUME_RF

Specifies the volume from which a data set is to be retrieved from an IBM mainframe.

Range: Zero to six characters

Default: None

Using the Send Report Command

Use the Send Report command to send a copy of a report file from a local system to a remote system for printing.

The -c2 option specifies that this is a Send Report transfer. The report file on the local system is specified by the LOCAL_FILE_SR parameter.

Example 1:

In the following example, the xcom62 command is used. The configuration file is */myconfig.cnf*, the local report file sent is named */tmp/myfile.rpt*, and it is placed on HOLD status on the remote system until released by the remote system.

```
xcom62 -c2 -f /myconfig.cnf LOCAL_FILE_SR=/tmp/myfile.rpt
HOLDFLAG=YES
```

Example 2:

In the following example, the xcomtcp command is used. The configuration file is */myconfig.cnf*, the local report file sent is named*/tmp/myfile.rpt*, and it is placed on HOLD status on the remote system until released by the remote system.

```
xcomtcp -c2 -f /myconfig.cnf LOCAL_FILE_SR=/tmp/myfile.rpt
HOLDFLAG=YES
```

Parameters

The Send Report parameters are shown below.

CARRIAGE_CONTROL_CHARACTERS

Indicates the type of printer carriage-control codes, if any, that are included in the report file.

Note: For report transfers only.

ASA

ASA control codes in column 1.

If CARRIAGE_CONTROL_CHARACTERS=ASA, then digest generation is not supported.

IBM

IBM Machine Characters (valid only for IBM mainframes).

BYPASSASA

If data is already in ASA format, bypass conversion.

OTHER

No carriage-control codes are used.

Default: OTHER

CLASS

The print class assigned to a report transferred to a remote system.

If the remote system is an IBM mainframe, this field designates the JES SYSOUT class.

Note: For report transfers only.

Example:

Enter **B** to print the report through SYSOUT=B.

Range: 1 character

Default: None

COPIES

The number of copies that are to be sent. If this parameter is not specified, the remote system queues one copy of the report to the system's default printer. For report transfers only.

Range: 1 to 999

Default: 1

DESTINATION

Identifies the printer or other device on the remote system where the report is to be sent. If this parameter is not specified, the remote system sends the report to the system's default printer. For report transfers only.

0 to 16 characters

For indirect transfers and for Version 1.

0 to 21 characters

For transfers that are not indirect and for Version 2.

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

DISPOSITION

Indicates what the remote system does with the report file after the report has been printed. For report transfers only.

Note: This field is not used when the remote system is an IBM mainframe.

DELETE

After printing the report is deleted.

KEEP

After printing the report is kept.

HOLD

After printing the report is held.

Default: DELETE

FCB

Identifies the FCB JCL parameter when sending the report file to an IBM mainframe, defining print density, lines per page, and so on. For report transfers only.

Range: Zero to four characters

Default: None

FORM

The type of form that should be used to print the report. Because CA XCOM Data Transport places the print job in the remote system's print queue, the print control functions depend on the remote system. The user must verify beforehand that the requested form is available at the remote site. For report transfers only.

Note: When sending a report to a VAX computer, leave this parameter blank unless you are certain that this is a valid form type. VMS interprets this to mean that no special form is being requested.

Range: 0 to 10 characters

Default: None

HOLDFLAG

Indicates whether a transferred report file is to be placed on HOLD on the remote system or is to be printed immediately. For report transfers only.

Range: YES or NO

Default: NO

LOCAL_FILE_SR

Local file name to be sent as a report to the remote system. If this value is null or unset, then CA XCOM Data Transport reads the standard input file. For report transfers only.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM_SR

The name of the remote system to which a report is sent. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

REPORT_TITLE

This field contains the report title and job number that will be printed on the report. The field has the following format:

8 characters	2 characters	8 characters
Job Name	Blanks	Job Number

The job is optional and can be skipped. The job name can also be skipped, but if you skip the job name and want to use the job number, you must pad the number with 10 blanks.

Note: For report transfers only.

Examples

```
REPORT_TITLE="Salary94   Job12345"
REPORT_TITLE="           Job23456"
```

Non-example

```
REPORT_TITLE="      Job34567"
```

This is not a valid REPORT_TITLE because the job number spans both subfields.

This parameter is used by CA XCOM Data Transport on remote systems in the following ways:

System	Uses the REPORT_TITLE...
z/OS	To interpret a non-blank value in this field as specifying the generation of a separator (banner) page for this value.
VAX/VMS	To print with the report.
UNIX/Linux	To allow CA XCOM Data Transport to pass this field to the LP spooler as a title field.
Other systems	As a descriptive comment only and does not print it as part of the report.

Range: 0 to 21 alphanumeric or blank characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

SPOOL_FLAG

Indicates whether the report is to be spooled to disk or printed immediately. For report transfers only.

Note: If the remote system is an IBM mainframe, this parameter has no effect on the transfer.

Range: YES or NO

Default: YES

Support for IBM Machine Code Characters

IBM Machine Code characters will be converted to the proper combination of form feed and line feed characters when the report is received from a machine code data set on the mainframe, such as FBM, or the partner specifies that the report has machine code controls.

Using the Send Job Command

Use the Send Job Command to send an executable file to a remote system to perform a job.

The -c3 option specifies that this is a Send Job transfer. The local job file is specified by the LOCAL_FILE_SJ parameter.

The executable file or command script that contains the job must have in it the control statements that are needed to execute the job on the remote system. These control statements must be in a form that is recognized by the remote system. For example, if you are sending to an IBM z/OS system, this file could contain JCL statements. If you are sending to a UNIX or Linux system, this file could contain a shell script. For sending to a Windows system, the file would be a DOS .bat file.

Example 1

In the following example, the xcom62 command is used. The job sent for execution on the remote system is in the file */tmp/myfile*.

```
xcom62 -c3 -f LOCAL_FILE_SJ=/tmp/myfile
```

All other necessary parameters are read from the default configuration file xcom.cnf.

Example 2

In the following example, the xcomtcp command is used. The job sent for execution on the remote system is in the file */tmp/myfile*.

```
xcomtcp -c3 -f LOCAL_FILE_SJ=/tmp/myfile
```

All other necessary parameters are read from the default configuration file xcom.cnf.

Parameters

The Send Job parameters are shown below.

LOCAL_FILE_SJ

Indicates the name of the file on the local system to be sent as a job. All the UNIX or Linux file naming conventions apply. If this value is null or unset, then CA XCOM Data Transport reads the standard input file.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM_SJ

The name of the remote system to which a job is sent. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

How to Send a Metatransfer

Use this command at the command prompt to send a metatransfer from a local system to a remote system.

Note: If the remote system is z/OS, it can be running release r11 or higher of CA XCOM Data Transport..

The **-c5** option specifies a metatransfer. The INQ_FILE command line parameter specifies the file containing information about the transfers being performed in this request.

Example:

```
XCOMTCP -c5 -f <CNF or XML configuration file> INQ_FILE=MYSCHEDULE STCIP=?? STCPORT=??
LUSERID=?? LPASSWORD=?? LDOMAIN=?? SECURE_SCHEDULE=<Y or N> STCTRNENCRL_CIPHER=??
```

The Send a Metatransfer (-c5) parameters:

INQ_FILE (see page 194)
LDOMAIN (see page 194)
LPASSWORD (see page 195)
LUSERID (see page 195)
SECURE_SCHEDULE (see page 195)
STCIP (see page 195)
STCPORT (see page 195)
STCTRNENCRL_CIPHER (see page 452)

Parameters

This section describes the metatransfer (-c5) parameters.

-f <configfilename>

The name of the CNF or XML configuration file name for a metatransfer.

Range: 1 to 256 characters

Default: None

INQ_FILE

Specifies the complete path information of the file to contain the information required when doing an inquire metatransfer (-c6) about a metatransfer (-c5).

Range: 1 to 256 characters

Default: None

LDOMAIN

The domain associated with LUSERID and LPASSWORD, if the target system is Windows, when handling a metatransfer (-c5).

Range: 1 to 15 characters

Default: None

LPASSWORD

Specifies the password of the local user to be validated on the CA XCOM Data Transport server handling a metatransfer (-c5).

Range: 1 to 31 characters

Default: None

LUSERID

The user ID to use on the CA XCOM Data Transport system receiving a -c5 transfer.

Range: 1 to 12 characters

Default: None

SECURE_SCHEDULE

Specifies whether the -c5, -c6, or -c7 metatransfer uses SSL.

Y

The metatransfer uses SSL.

N

The metatransfer does not use SSL.

Default: N

STCIP

Specifies the IP address or name of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 64 characters

Default: None

STCPORT

Specifies the TCP/IP port number of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 65535

Default: 8044

STCTRNENCRL_CIPHER

See Appendix A: Parameter section for more details.

How to Inquire on the Status of Metatransfers

Use this command at the command prompt to inquire on the status of metatransfers.

The **-c6** option specifies that this is a metatransfer status inquiry. The file containing information about the transfers being performed by the –c5 metatransfer request is specified by the INQ_FILE command line parameter.

Example

```
xcomtcp -c6 stcip=?? stcport=?? inq_file=?? inq_wait=?? hist_file=??
```

The parameters for inquiring on the status of metatransfers:

HIST_FILE (see page 196)
INQ_FILE (see page 194)
INQ_WAIT (see page 197)
STCIP (see page 195)
STCPORT (see page 195)

Parameters

This section describes the parameters for inquiring on the status of metatransfers.

HIST_FILE

Specifies the complete path information of the file to contain the history records returned by an inquire metatransfer (-c6) or a get history record retrieval metatransfer (-c7).

Range: 0 to 256 characters

Default: None

INQ_FILE

Specifies the complete path information of the file to contain the information required when doing an inquire metatransfer (-c6) about a metatransfer (-c5).

Range: 1 to 256 characters

Default: None

INQ_WAIT

Specifies how long CA XCOM Data Transport should wait for the metatransfer (-c5) to complete when doing an inquire metatransfer (-c6).

hhmmss

Specifies in hours (*hh*), minutes (*mm*), and seconds (*ss*) the length of the time that CA XCOM Data Transport should wait for the metatransfer request to complete.

Default: 001000 (10 minutes)

Note: This parameter's value is expressed as a number of up to six digits (for example, 010000 for 1 hour).

STCIP

Specifies the IP address or name of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 64 characters

Default: None

STCIP

Specifies the IP address or name of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 64 characters

Default: None

STCPORT

Specifies the TCP/IP port number of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 65535

Default: 8044

Return Codes

The inquire return codes are as follows:

0

All transfers in INQ_FILE have been completed.

4

INQ_FILE is empty.

8

OPEN failed for INQ_FILE or HIST_FILE.

36

At least one transfer in the INQ_FILE has not been completed.

40

One or more transfers specified in the INQ_FILE did not complete successfully.

44

At least one transfer in the INQ_FILE has not been completed *and* one or more transfers in the INQ_FILE did not complete successfully.

48

No history records were returned.

This condition can be caused by the STCIP= parameter on the –c6 (inquire) metatransfer pointing to a different CA XCOM system than the STCIP= on the –c5 metatransfer request. In other words that CA XCOM system is the wrong CA XCOM system.

How to Retrieve History Records

Use the command, as shown in the example, at the command prompt to retrieve history records..

This command accepts a history filter file in XML format or CNF format and creates a file of the matching history records in a format that a CA Easytrieve report can be generated from.

The **-c7** option specifies that this is a history transaction. The history filter file is generated in XML format by using the Export button on the Get History Records page of the GUI or is created in a CNF configuration file format by using the supported history record filter parameters. For a list of the parameters that can be used in a CNF configuration file for a -c7 metatransfer, see History Record Filter Parameters.

For more information about the history filter fields in the GUI, see the *CA XCOM Data Transport GUI Online Help*.

Example:

```
xcomtcp -c7 -f <configfilename> STCIP=<ip-value> STCPORT=<port-value>
SECURE_SCHEDULE=<YES | NO> HIST_FILE=<history-file-name>
```

The Retrieve History Records (-c7) parameters:

SECURE_SCHEDULE (see page 195)
STCIP (see page 195)
STCPORT (see page 195)

Parameters

This section describes the history retrieval (-c7) parameters

-f <configfilename>

The name of the CNF or XML configuration file name for a metatransfer.

Range: 1 to 256 characters

Default: None

SECURE_SCHEDULE

Specifies whether the -c5, -c6, or -c7 metatransfer uses SSL.

Y

The metatransfer uses SSL.

N

The metatransfer does not use SSL.

Default: N

STCIP

Specifies the IP address or name of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 64 characters

Default: None

STCPORT

Specifies the TCP/IP port number of the CA XCOM Data Transport server to handle the -c5, -c6, or -c7 metatransfer request.

Range: 1 to 65535

Default: 8044

History Record Filter Parameters

The history record filter parameters that can be used with a –c7 metatransfer.

The History Record Filter parameters:

OEDATE (see page 202)
OETIME (see page 203)
OFILE (see page 204)
OFILECASE (see page 205)
OFILETYPE (see page 205)
OFLMAX (see page 206)
OFLMIN (see page 207)
OID (see page 208)
OINIT (see page 209)
OJOB (see page 208)
OLIMIT (see page 209)
OLMSG (see page 210)
OLOCATN (see page 211)
OLU (see page 211)
OREQ (see page 212)
OSDATE (see page 213)
OSTIME (see page 214)
OSYSID (see page 215)
OSYSNAME (see page 215)
OTNAME (see page 216)
OTYPE (see page 217)
OTYPEREQ (see page 218)
OUSER (see page 218)
OVOL (see page 219)

OEDATE

Limits the history request to only those file transfers that were scheduled or completed on or before the end date.

YYYYMMDD

The end date used to limit the history request to only those file transfers that were scheduled or completed on or before the end date specified.

YYYY

The four-digit year

MM

The two-digit number of the month, as follows:

```
01 = January      02 = February     03 = March
04 = April        05 = May          06 = June
07 = July         08 = August       09 = September
10 = October      11 = November     12 = December
```

DD

The two-digit day of the month (01 to 31)

Default: Current date

Notes:

- OEDATE and OETIME form an end date and time used to limit the history request to only those file transfers that were scheduled or completed on or before the date and time specified.
- See OSDATE and OSTIME for the start date and time.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OETIME

Limits the history request to only those file transfers that were scheduled or completed on or before the end time.

HHMMSS

The end time used to limit the history request to only those file transfers that were scheduled or completed on or before the end time specified.

HH

The two-digit hour (00 through 23)

MM

The two-digit minute (00 through 59)

SS

The two-digit second (00 through 59)

Default: 235959

Notes:

- OEDATE and OETIME form an end date and time used to limit the history request to only those file transfers that were scheduled or completed on or before the date and time specified.
- See OSDATE and OSTIME for the start date and time.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OFILE

Specifies the file name, local, or remote, to match for a history request.

You can use the following wildcard characters when you specify the file name:

*** or %**

Represents a string of zero or more characters.

_

Represents any single character.

Range: 1 to 256

Default: None

Example:

An OFILE value of %MASTER.FIL_.G* tells CA XCOM Data Transport to locate a file with following attributes:

- Starting with anything
- Ending with anything
- With the characters MASTER.FIL found in the name, followed by any single character and .G.

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

OFILECASE

Specifies whether the specified file name (OFILE parameter) search is case-sensitive.

YES

The value that is specified for the OFILE parameter is case-sensitive.

NO

The value that is specified for the OFILE parameter is not case-sensitive.

Default: NO

Notes:

- For a case-sensitive search, case-sensitive collation is set to the history database table. Such collation can also be set to the 'file' and 'lfile' columns in the history database table.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

OFILETYPE

Limits the history request to those transfers with the specified FILETYPE.

JOB

Restrict the search for file transfers to only FILETYPE JOB transfers.

REPORT

Restrict the search for file transfers to only FILETYPE REPORT transfers.

FILE

Restrict the search for file transfers to only FILETYPE FILE transfers.

Default: All file types

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer are authorized to access.

OFLMAX

Limits the history request to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

***NNNNNNNNN*(*N*|*X*)**

A 1- to 10-digit number, where the last digit can be either another numeric digit or a 1-character qualifier. This parameter is used to restrict the search for file transfers to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

X

One of the following qualifiers (default B):

- B = Bytes
- K = Kilobytes
- M = Megabytes
- G = Gigabytes
- T = Terabytes (maximum allowed value is 8388607T)
- P = Petabytes (maximum allowed value is 8191P)
- E = Exabytes (maximum allowed value is 7E)

Default: 7E

Notes:

- Use OFLMIN and OFLMAX to specify a range that can be used to limit the history request by the number of bytes transferred.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OFLMIN

Limits the history request to only those file transfers where the number of bytes transferred is equal to or greater than the value specified.

***NNNNNNNNN*(*N*|*X*)**

A 1- to 10-digit number, where the last digit can be either another numeric digit or a 1-character qualifier. This parameter is used to restrict the search for file transfers to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

X

One of the following qualifiers (default B):

- B = Bytes
- K = Kilobytes
- M = Megabytes
- G = Gigabytes
- T = Terabytes (maximum allowed value is 8388607T)
- P = Petabytes (maximum allowed value is 8191P)
- E = Exabytes (maximum allowed value is 7E)

Default: None

Notes:

- Use OFLMIN and OFLMAX to specify a range that can be used to limit the history request by the number of bytes transferred.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OID

Limits the history request to only those file transfers with a specific transfer ID. The transfer ID is a user-defined identifier for file transfer requests.

xxxxxxxxxx

A 1- to 10-character transfer ID used to limit the history request to only those file transfers that contain the specified transfer ID.

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **TRANSFER01** returns the same results as specifying **transfer01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OJOB

Specifies the invoking job name to match for a history request.

Range: 1 to 8 characters

Default: None

Note:

- Supported in –c7 CNF metatransfers only.
- The wildcard character, *, can be used for this parameter only when specified as the last character.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

OINIT

Limits the history request to only locally initiated transfers or only remotely initiated transfers.

LOCAL or L

Restrict the search for file transfers to only locally initiated transfers.

REMOTE or R

Restrict the search for file transfers to only remotely initiated transfers.

Default: All local and remote transfers

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OLIMIT

Sets the maximum number of history records that can be returned.

Range: 0 to 9999

Notes:

- OLIMIT=0 means that all records will be returned.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OLMSG

Limits the history request by the transfer's last message. The format to use for CA XCOM Data Transport messages is as follows:

XCOM*XNNNNS*

A 1- to 10-character name used to restrict the search for file transfers to those where the last message matches the value specified.

XCOM

Indicates that the message is from CA XCOM Data Transport.

X

Identifies the system.

NNNN

Is the message number.

S

Is the message severity:

- I = Informational
- W = Warning
- E = Error

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is case insensitive.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OLOCATN

Limits the history request to transfers in one of the following locations:

- Transfers that have been completed and have been stored in a database
- Transfers still in the queue

DATABASE

Restrict the search for file transfers to those stored in the database. This applies only if a history database has been set up at your site.

QUEUE

Restrict the search for file transfers to those still in the queue.

Default: Transfers located in both the QUEUE and DATABASE.

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OLU

Limits the history request to only those file transfers with a specific remote LU name.

xxxxxxxx

A one- to eight-character LU name used to limit the history request to only those file transfers for the specified remote LU.

Default: None

Notes:

- The wildcard character (*) can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **LU01** returns the same results as specifying **lu01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OREQ

Limits the history request to only those file transfers that contain this specific request number.

NNNNNN

A one- to six-character request number used to limit the history request to a specific request number.

Default: All request numbers

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OSDATE

Limits the history request to only those file transfers that were scheduled or completed on or after the start date.

YYYYMMDD

The start date used to limit the history request to only those file transfers that were scheduled or completed on or after the start date specified.

YYYY

The four-digit year

MM

The two-digit number of the month, as follows:

```
01 = January      02 = February     03 = March
04 = April        05 = May          06 = June
07 = July         08 = August       09 = September
10 = October      11 = November     12 = December
```

DD

The two-digit day of the month (01 to 31)

Default: Current date

Notes:

- OSDATE and OSTIME form a start date and time used to limit the history request to only those file transfers that were scheduled or completed on or after the date and time specified.
- See OEDATE and OETIME for the end date and time.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OSTIME

Limits the history request to only those file transfers that were scheduled or completed on or after the start time.

HHMMSS

The start time used to limit the history request to only those file transfers that were scheduled or completed on or after the start time specified.

HH

The two-digit hour (00 through 23)

MM

The two-digit minute (00 through 59)

SS

The two-digit second (00 through 59)

Default: 000000

Notes:

- OSDATE and OSTIME form a start date and time used to limit the history request to only those file transfers that were scheduled or completed on or after the date and time specified.
- See OEDATE and OETIME for the end date and time.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OSYSID

Limits the history request to only those file transfers executed on the specified system ID.

Note: OSYSNAME and OSYSID together uniquely identify a CA XCOM Data Transport r11.5 and higher server.

Range: 1 to 4 characters

Default: None

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OSYSNAME

Limits the history request to only those file transfers executed on the specified system name.

Note: OSYSNAME and OSYSID together uniquely identify a CA XCOM Data Transport r11.5 and higher server.

Range: 1 to 8 characters

Default: None

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OTNAME

Limits the history request to only those file transfers with a specific remote TCP/IP name or TCP/IP address.

xxxxxxxx...x

A 1- to 64-character TCP/IP name or address used to limit the history request to only those file transfers for the specified TCP/IP name or address.

Default: None

Notes:

- The wildcard character (*) can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **TCPIP01** returns the same results as specifying **tcpip01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OTYPE

Specifies if the history request should include inactive transfer requests, active transfer requests, or completed transfers.

I

Selecting this option retrieves history records for file transfers whose execution is still pending.

A

Selecting this option retrieves history records for file transfers that are currently in progress.

C

Selecting this option retrieves history records for file transfers that have been successfully or unsuccessfully completed.

ALL|AIC|*

Selecting this option retrieves history records for all file transfers, independent of their status.

Default: AIC

Notes:

- You can also specify values in combination; for example, specify AI to request history records for file transfers whose execution status is inactive and active.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OTYPEREQ

Limits the history request to only send transfers or only receive transfers.

SEND or S

Restrict the search for file transfers to only send transfers.

RECEIVE or R

Restrict the search for file transfers to only receive transfers.

Default: All send and receive transfers

Notes:

- This parameter is case sensitive.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OUSER

Limits the history request to only those file transfers submitted by a specific user.

xxxxxxxxxxxx

A 1- to 12-character user name used to limit the history request to only those file transfers submitted by the specified user.

Default: None

Notes:

- The wildcard character (*) can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **USER01** returns the same results as specifying **user01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OVOL

Specifies the volser (local or remote) to match for a history request.

Range: 1 to 6 characters

Default: None

Note:

- Supported in –c7 CNF metatransfers only.
- The wildcard character, *, can be used for this parameter only when specified as the last character.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

Command Examples for Metatransfers and Inquire on Status

This is a sample script file for performing a metatransfer and an inquire, and producing a CA Easytrieve report:

```
xcomtcp -c5 -f c:\temp\xcomcnf\asm2.cnf stcip=xcomwinca.ca.com
stcport=48316 inq_file=/tmp/xcom.inq.file
xcomtcp -c6 stcip=xcomwinca.ca.com stcport=48316
inq_file=/tmp/xcom.inq.file inq_wait=000005
hist_file=/tmp/xcomhist
echo Process returned $?
$XCOM_HOME/ezt/xcomehrp /tmp/xcomhist
```

Note: The –c5 metatransfer parameters LUSERID and LPASSWORD are not shown in the example above but are required and can be specified in the configuration file.

Suppose this script file is called xmeta.script and is located in /xcomcnf.

To execute this batch file

1. Submit the following command to change to the directory where xcomtcp is located:

   ```
   cd /opt/CA/XCOM/bin
   ```

   Note: This step is not required if the CA XCOM Data Transport directory is in the system path.

2. Submit the following command to run the script file:

   ```
   /xcomcnf/meta.script
   ```

How to Send a Ping Request

Use the Ping command at the command prompt to send a request from the Local System to the Remote System to determine whether the system is available to XCOM.

Information about the remote CA XCOM Data Transport Server is returned as a result of this command.

Example:

```
XCOMTCP -ping REMOTE_SYSTEM=?? {PORT=??} {SECURE_SOCKET=YES|NO}
{TRNENCRL_CIPHER=xx}
```

The Ping request parameters:

TRNENCRL_CIPHER (see page 221)

Parameters

This section describes the parameters of Ping request.

REMOTE_SYSTEM

The name of the remote system that receives the Ping request.

For SNA/APPC protocols, the name is the symbolic destination name from the CPI C side information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 64 characters

Default: None

PORT

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 1 to 65535

Default: 8044

SECURE_SOCKET

Specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and connects to an SSL listener on the remote partner.

NO

Performs a non-secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

TRNENCRL_CIPHER

TRNENCRL_CIPHER specifies the requested list of ciphers which are used to encrypt the password fields for locally initiated transfers.

The cipher list consists of one or more ciphers from the table that colons separate. The characters (!) or (-) can precede each cipher in the cipher list.

If (!), is used then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-), is used then the ciphers are deleted from the list, but some or all of the ciphers are added again using later options.

Default: COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining
DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining

3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining
RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.

COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To request all ciphers except for any of the DES ciphers, use the following command:

```
TRNENCRL_CIPHER=ALL:!DES
```

To request only a 3DES or AES cipher, use the following command:

```
TRNENCRL_CIPHER=3DES:AES
```

To disable the cipher negotiation or remain backward compatible with the earlier releases of CA XCOM Data Transport, use the following command:

```
TRNENCRL_CIPHER=COMPAT
```

Command Example of Ping Request

```
xcomtcp -ping REMOTE_SYSTEM=XX PORT=XX TRNENCRL_CIPHER=ALL
XCOMN0882I PING INFO FOR <SYSTEMNAME>
XCOMN0882I RELEASE=r11.6 SP00 GEN LEVEL <LEVEL> SYSTEM NAME=<SYSNAME> SYSTEM
ID=<SYSID>
```

Tracing Problems

The tracing parameters are used to collect information at the request of CA XCOM Data Transport Customer Support in order to diagnose a problem. The trace output is primarily intended to provide information to Customer Support, and they usually request a level of tracing for you to use. The -t option of the xcom62 or xcomtcp command will direct trace output to stderr, the standard error output.

This trace facility collects information about a transfer from the point where a transfer is queued or submitted and a connection is established. The resulting trace data is stored in a file named *tidnumber*.tra, where *tidnumber* = the transfer ID.

Example 1

In the following example, the xcom62 command is used. The configuration file is */myconfig*, the XTRACE parameter is used to set the trace level to 10 to provide full tracing information and the trace information is written to a trace file. The protocol is specified by the PROTOCOL parameter.

```
xcom62 –c1 -f /myconfig XTRACE=10 PROTOCOL=SNA
```

This section contains information about collecting information to diagnose problems when requested by CA XCOM Data Transport Customer Support.

Example 2

In the following example, the xcomtcp command is used. The configuration file is */myconfig*, the XTRACE parameter is used to set the trace level to 10 to provide full tracing information and the trace information is written to a trace file. The protocol is specified by the PROTOCOL parameter.

```
xcomtcp –c1 -f /myconfig XTRACE=10 PROTOCOL=TCPIP
```

Return Codes

For queued transfers, the return code indicates success or failure in queuing the transfer.

For non-queued transfers the return codes from XCOMTCP and XCOM62 are as follows:

0

Transfer successful

Non-zero

Transfer failed

The CA XCOM Data Transport return code is the same as the error message number, which is usually a three-digit number. In order to pass back the error message number, CA XCOM Data Transport must make sure the error code is 256 or less, because of system restrictions. To do this, CA XCOM Data Transport subtracts 256 from the error message number if it is greater than 256. In other words, the return code is modulo 256.

Example:

Message XCOMU0298E would give 42 as the return code (298 - 256 = 42).

Refer to $XCOM_HOME/api/include/xcomerr.h. These return code**s** are not modulo 256.

Note: $XCOM_HOME is an environment variable.

Parameter

The tracing parameter is shown below.

XTRACE

Indicates the level of desired execution tracing.

0

No tracing.

1 to 8

Levels of tracing from 1 (lowest) to 8 (highest).

9

Show the contents of the data buffers.

10

Includes levels 1 to 9 with additional detailed technical information.

Default: 0

Note: Turning on the trace can seriously degrade performance.

Setting Up Log Files

CA XCOM Data Transport automatically collects information about transfers and stores it in a log file. Read the log file to see information about transfers. The log indicates information such as when a transfer was started, stopped, completed, deleted, suspended, or aged off of the queue. For each transfer, it shows the time, date, transfer ID and number of blocks and bytes transmitted.

Example 1

In the following example, the xcom62 command is used. The configuration file is /tmp/myconfig and the log file is */home/phil/xcom/mylog*.

```
xcom62 –c1 -f /tmp/myconfig XLOGFILE=/home/phil/xcom/mylog
```

Example 2

In the following example, the xcomtcp command is used. The configuration file is */tmp/myconfig* and the log file is */home/phil/xcom/mylog*.

```
xcomtcp –c1 -f /tmp/myconfig XLOGFILE=/home/phil/xcom/mylog
```

Parameter

The logging parameter is shown below.

XLOGFILE

The name of the file where CA XCOM Data Transport logs activity. If you do not specify this parameter, the systemwide log file $XCOM_HOME/xcom.log is used. If you specify this parameter with a different file name, the logging information is only sent to the specified file.

Note:

- If QUEUE=YES, specify the full path name.
- $XCOM_HOME is an environment variable.

Range: 0 to 256 characters

Setting File Type Conversion

CA XCOM Data Transport uses internal character conversion sets to handle different file formats when files are sent or received. These default conversion sets are stored in files as conversion tables. When executing a non-binary file transfer that requires an ASCII/EBCDIC conversion, CA XCOM Data Transport uses these tables by default (INTERNAL_CONVERSION_TABLES=YES).

The default conversion tables are as follows:

ASCII to EBCDIC

$XCOM_HOME/convtab/atoe.tab

EBCDIC to ASCII

$XCOM_HOME/convtab/etoa.tab

Note: $XCOM_HOME is an environment variable.

For conversions, when transferring text files for transfers that are initiated on a UNIX or Linux system, CARRIAGE_FLAG is set to YES and CODE_FLAG is set to EBCDIC.

Example 1

In the following example, the xcom62 command is used. An EBCDIC text file is being transferred:

```
xcom62 –c1 -f CARRIAGE_FLAG=YES CODE_FLAG=EBCDIC
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Example 2

In the following example, the xcomtcp command is used. An EBCDIC text file is being transferred:

```
xcomtcp –c1 -f CARRIAGE_FLAG=YES CODE_FLAG=EBCDIC
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Parameters

Use the parameters shown below to specify the characteristics of the file and the data involved in the transfer. For information on using custom character conversion sets rather than the CA XCOM Data Transport defaults, see the topic, Creating Custom Character Sets for File Conversion, later in this chapter.

The file type parameters are listed next.

CARRIAGE_FLAG

Specifies the type of file being transferred and some special characteristics of the conversion done during the transfer.

YES

Indicates that the transferred file is a text file and a newline character should be added to the end of incoming records. Also, newline characters are removed from the ends of lines before an outgoing record is sent.

NO

Indicates no special processing.

MPACK

Indicates a text file with record packing. Uses 2K pack buffer.

VLR

Indicates a binary file of variable-length records with a field of four bytes preceding each record. Applies to locally initiated transfers only.

XPACK

Indicates a text file with record packing. Uses 31K pack buffer.

Note: MPACK does not support a MAXRECLEN (actual record length) over 2K. XPACK does not support a MAXRECLEN (actual record length) over 31K.

Default: YES

CODE_FLAG

Used to identify the type of data being transferred.

ASCII

An ASCII file is being transferred. This transfer indicates that the incoming file is assumed to be ASCII format, and is not translated. Therefore the file on the remote system has to be in ASCII format before it is transferred.

BINARY

A binary file, such as an executable file, is being transferred. This file indicates to a remote system that it is not to translate the data it is exchanging with your system.

EBCDIC

An EBCDIC file is being transferred. The data is translated from EBCDIC to ASCII when the local system receives the data. From ASCII to EBCDIC, when the local system sends the data.

UTF8

A Unicode file which is based on the UTF8 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

UTF16

A Unicode file which is based on the UTF16 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

Default: ASCII

Creating Custom Character Sets for File Conversion

The custom character sets are stored in standard ASCII files as conversion tables. Each original character and the character to which it is translated are defined by a single line in the file. The line number of the file itself indicates the decimal code value of the original character.

The character conversion table must be the only contents of the conversion file. Do not insert any comments or additional numbers. Each conversion file must contain 256 lines. If your text editor or word processor provides a line counter, you can use it to help you keep track of the line numbers. The procedures and examples below assume that the editor's line counter begins at 1.

When to Use Character Conversion

Use conversion processing when the following situations occur:

- When a non-USA character set is being used.
- When the transferred file is to be converted to all uppercase or lowercase characters.
- When CA XCOM Data Transport's default character conversion sets would not be appropriate.

For information on using CA XCOM Data Transport's default character conversion sets, see Setting File Type Conversion in this chapter.

Procedure

To create a custom character translation table

1. From the command line, use vi or another editor to create and open a file.
2. Define each character on a separate line.

CA XCOM Data Transport for UNIX or Linux performs the EBCDIC to ASCII translation as follows:

- When an EBCDIC code 0 is received, CA XCOM Data Transport substitutes the ASCII value from the first line (line 1) of the table.
- For EBCDIC code 1, CA XCOM Data Transport takes the value from the second line (line 2), and so on.

Because the first line is for code 0, you need to add 1 to the line number when determining which line of the table to modify.

Example 1

A question mark (?) must be sent to the mainframe in EBCDIC. Set up the translation table to translate it from ASCII to EBCDIC.

The ASCII code for ? is 63 (decimal). The EBCDIC code for ? is 111 (decimal). Take the ASCII code and add 1. This is 64. Go to line 64 of the ATOE table and type the value 111.

Example 2

An A is received from the mainframe in EBCDIC. Set up the translation table to translate it to ASCII.

The EBCDIC code for A is 193 (decimal). The ASCII code for A is 65. Take the EBCDIC code for A and add 1. This is 194. Go to line 194 of the ETOA table and type the value 65.

Hexadecimal Numbers in Conversion Tables

An EBCDIC character can be represented by either its decimal code or its hexadecimal equivalent. Hexadecimal numbers should be prefixed by 0*x*.

Example

In the following example, to translate the EBCDIC character A to the equivalent ASCII code using a hexadecimal number, line 194 of the conversion file would contain only the information below:

```
0x41
```

Customizing Default Character Sets

You can build a conversion file by customizing a copy of the atoe.tab or etoa.tab character sets and changing the character translations only where necessary.

To specify external character conversions tables for use by transfers, the CODETABL parameter (one to three characters with no default value). The CODETABL parameter is a prefix to the name of the files containing the external ASCII-to-EBCDIC and EBCDIC-to-ASCII character conversion tables.

Note: The CODETABL parameter is valid (resulting with the user-defined conversion tables loaded) only if the xcom.glb parameter INTERNAL_CONVERSION_TABLES=NO. Failure to specify INTERNAL_CONVERSION_TABLES=NO results in the source or target system not loading or using the defined conversion tables.

When creating external custom ASCII-to-EBCDIC and EBCDIC-to-ASCII translation tables, create the ATOE/ETOA tables using the *xxx*atoe.tab and *xxx*etoa.tab naming convention.

- All custom translation tables must be in the $XCOM_HOME/convtab directory. The following files are supplied on your distribution media and contain the tables that are used to map ASCII-to-EBCDIC and EBCDIC-to-ASCII translations:$XCOM_HOME/convtab/atoe.tab
- $XCOM_HOME/convtab/etoa.tab

Note: $XCOM_HOME is an environment variable.

Procedure

To translate all lower case ASCII characters to upper case EBCDIC characters

1. a. To create a custom character set, do the following:

   From the command line, type

   `cp $XCOM_HOME/convtab/atoe.tab conversionfilename`

   Note:

   - *conversionfilename* can be any name you choose.
   - $XCOM_HOME is an environment variable.

   $XCOM_HOME/convtab/atoe.tab is copied to a new file and named conversionfilename.

   or

   b. To create a customer character set in order to use the CODETABL parameter, do the following:

   From the command line, type

   `cp $XCOM_HOME/convtab/atoe.tab $XCOM_HOME/convtab/tstatoe.tab`

   Note:

   - The prefix used above is tst but it can be any 1 to 3 character prefix you choose.
   - $XCOM_HOME is an environment variable.

   $XCOM_HOME/convtab/atoe.tab is copied to a new file and named tstatoe.tab.

2. Use vi or another editor to open the file that was created in Step 1 and copy the values from lines 66 to 91 to lines 98 to 123.

   Note: The decimal ASCII code for a is 98 and the decimal ASCII code for z is 123.

   The values in lines 98-123 are changed to represent uppercase EBCDIC characters.

3. Save the file and exit the editor.

Specifying a Custom Character Set

To use a custom character set you must change the parameter values in the xcom.glb file for the INTERNAL_CONVERSION_TABLES parameter and then either the CODETABL value, or the ETOA_FILENAME and/or ATOE_FILENAME values, depending on the custom character set changes that were made. To activate the changes in xcom.glb, you must restart xcomd.

To specify a custom character set in xcom.glb

1. From the command line, enter the following:

   ```
   vi xcom.glb
   ```

   The xcom.glb file is opened for editing.
2. Set INTERNAL_CONVERSION_TABLES=NO.
3. a. Change the value of CODETABL to the 1 to 3 character prefix specified for the xxxatoe.tab/xxxetoa.tab file names you created.

   or

   b. Change the values for ETOA_FILENAME and/or for ATOE_FILENAME to the file names containing the customized files you created.
4. Save xcom.glb and exit the editor.
5. Restart xcomd to activate the changes made.

Unicode and Multi-Byte Character Sets Support for Data Transfer

Before the advent of Unicode, a significant number of character sets were devised to permit the representation of symbols used in the Chinese, Japanese, Korean, and Taiwanese (CJK) languages. Today, Unicode is favored and there is an ongoing transition from these legacy character sets to Unicode encodings, most notably UTF-8 and UTF-16. Many CJK legacy Multi-Byte character sets are ASCII based, as is the case for the most commonly used Unicode encodings (i.e.UTF-8, UTF-16).

In the IBM mainframe (predominantly EBCDIC) world however composite character sets are commonly employed, involving a Shift-in/Shift-out encoding method. This encoding mechanism enables a single byte ASCII or EBCDIC character set to be used for the representation of Latin characters, in tandem with a multi byte character set for the representation of non-Latin characters. Shift-in and shift-out control characters are then inserted in the data stream to signal a switch between the two embedded character sets. For example, the CCSID 937 character set combines an EBCDIC single byte character set with a ‘Traditional Chinese’ multi-byte character set, whilst the CCSID 938 character set combines an ASCII single byte character set with the same ‘Traditional Chinese’ multi-byte character set.

CA XCOM Data Transport allows for transmission of text files that are encoded using Multi-Byte characters sets, including in-flight conversion of data between different character sets.

CA XCOM Data Transport utilizes the ICU (International Components for Unicode) toolkit to perform data conversion functions. For information on the ICU toolkit, please refer to the ICU website http://site.icu-project.org/.

Using Unicode Transfer

CA XCOM Data Transport is capable of transmitting data using either the UTF-8 or UTF-16 Unicode encodings. They can be specified with the CODE_FLAG parameter to allow for conversion of files from one character encoding to Unicode and back.

When a file's data comprises a low ratio of non-Latin characters versus Latin characters, UTF-8 encoding will consume fewer bytes and therefore result in a faster transfer. In contrast, when a file's data comprises a high ratio of non-Latin characters versus Latin characters, UTF-16 encoding will produce the best result.

The Unicode transfer Data Formats parameter:

CODE_FLAG

Specifying Charset

The CA XCOM Data Transport sending server converts the input encoding to UTF-8 or UTF-16, while the CA XCOM Data Transport receiving server converts to the required output encoding. This divides the conversion workload between the two CA XCOM Data Transport servers.

Use the LOCAL_CHARSET and REMOTE_CHARSET parameters in order to choose the local file and remote file character encoding. If not specified for the transfer, they default to the value specified for the DEFAULT_CHARSET global parameter in the XCOM.GLB file.

For a list of supported Charsets see Appendix C.

The Unicode transfer Charset parameters:

- DEFAULT_CHARSET
- LOCAL_CHARSET
- REMOTE_CHARSET

Handling Conversion Errors

Not all characters can be converted between Unicode and other charsets or vice versa. In most cases, Unicode is a superset of the characters supported by any given charset.

Use MBCS_INPUTERROR and MBCS_CONVERROR to specify what action CA XCOM Data Transport should take in the event of a character being encountered cannot be converted.

When erroneous data is encountered during conversion then the following actions are possible:

- Skip the erroneous data and continue.
- Replace the erroneous data with the charsets default substitution character.
- Replace the erroneous data with the supplied Unicode character.
- Fail the transfer.

The CA XCOM Data Transport sending server uses MBCS_INPUTERROR whereas the CA XCOM Data Transport receiving server uses MBCS_CONVERROR to specify the action. If not specified the value of the DEFAULT_INPUTERROR and DEFAULT_CONVERROR global parameters in XCOM.GLB will be used.

The Unicode transfer Conversion error handling parameters:

- DEFAULT_INPUTERROR
- DEFAULT_CONVERROR
- MBCS_INPUTERROR
- MBCS_CONVERROR

Record Processing

CA XCOM Data Transport uses a newline also known as line break or end-of-line (EOL) marker special character to identify records in the text files. With the advent of Unicode, all the end of the line delimiters are supported irrespective of the platform.

Use the LOCAL_DELIM or REMOTE_DELIM parameters to choose the delimiter depending on the charset used in a Unicode transfer. If these parameters are not specified then the value of the DEFAULT_DELIM global parameter in the xcom.glb file will be used.

The Unicode transfer Delimiter handling parameters:

- DEFAULT_DELIM
- LOCAL_DELIM
- REMOTE_DELIM

Examples

Example1:

In the following example, the XCOMTCP command is used to run a Unicode transfer by specifying CODE_FLAG=UTF8.

The local file input.txt encoded in CP949 is converted to EUC-KR.

```
XCOMTCP –c1 -f MYCONFIG.CNF LOCAL_FILE=input.txt CODE_FLAG=UTF8 LOCAL_CHARSET=CP949
REMOTE_CHARSET=EUC-KR
```

Example2:

In the following example, the conversion errors are handled.

If any erroneous character is found while reading the input file, the transfer will be FAILED. If any erroneous character is found while converting to the remote charset, the transfer will continue by substituting the malformed character with a default substitution character.

```
XCOMTCP –c1 -f MYCONFIG.CNF LOCAL_FILE=input.txt CODE_FLAG=UTF16 LOCAL_CHARSET=CP949
REMOTE_CHARSET=EUC-KR MBCS_INPUTERROR=FAIL MBCS_CONVERROR=REPLACE
```

Example3:

In the following example, the EBCDIC conversion is handled.

The ASCII based text file is converted to EBCDIC encoding. The ASCII LF (Line Feed) delimiter is used to detect the end of a record. The NL is added as a line delimiter in the output file.

```
XCOMTCP –c1 -f MYCONFIG.CNF LOCAL_FILE=input.txt CODE_FLAG=UTF8
LOCAL_CHARSET=ISO-8859-1 REMOTE_CHARSET=CCSID#37 LOCAL_DELIM=ASCII:LF
REMOTE_DELIM=EBCDIC:NL
```

Run Reports on History Records

You can run CA Easytrieve reports on CA XCOM Data Transport history records. Sample code is distributed with CA XCOM Data Transport for UNIX and Linux.

The CA Easytrieve execution programs xcomehrp and xcomesrp are installed in the $XCOM_HOME/ezt directory.

You can run the following reports:

- Transfer Information Report
- Statistical Data Transfer Report

Generate a Transfer Information Report

To generate a general Transfer Information Report

1. Go to the $XCOM_HOME/ezt directory.
2. Run *one* of the following commands:
   - To display the output report on the screen:

     ```
     xcomehrp {input_filename}
     ```

     Example:

     ```
     cd $XCOM_HOME/ezt
     xcomehrp
     ```

     Use this command to produce a report from xcomhist.histf (located in the $XCOM_HOME/ezt directory) and display the output report on the screen.
   - To place the output report into a file, where it can be reviewed with a text editor:

     ```
     xcomehrp {input_filename} > {PATH}/xcomehrp.out
     ```

     Example:

     ```
     xcomehrp /home/tester/xcom_history_record/XCOMHIST > /home/tester/
     xcom_history_record/output.rpt
     ```

     Use this command to produce a report from the XCOMHIST file (located somewhere other than $XCOM_HOME/ezt) and generate an output report file.

   {input_filename}

   This is an optional parameter. Use it to pass an input file of CA XCOM Data Transport history records (other than xcomhist.histf) which is located in the current directory.

   Default: xcomhist.histf

Generate a Statistical Data Transfer Report

To generate a Statistical Data Transfer Report

1. Go to the $XCOM_HOME/ezt directory.
2. Run the following command:

   `xcomesrp [time_interval] [flag_of_calculated_base] [input_filename]`

   time_interval

   This is a *mandatory* parameter.

   1

   One-hour interval to display data

   0.5

   Half-hour interval to display data

   Default: None

   flag_of_calculated_base

   This is an *optional* parameter.

   S

   The calculation base for start time.

   E

   The calculation base for end time.

   Default: E

   input_filename

   Use this optional parameter to pass an input file of CA XCOM Data Transport history records to the command. By default, it looks for a file with the name of xcomhist.histf in the current directory. This parameter allows you to specify an input file with a different path and/or file name.

   Default: xcomhist.histf

Example 1:

```
cd $XCOM_HOME/ezt

xcomesrp 1
```

Example 2:

```
cd $XCOM_HOME/ezt

xcomesrp 0.5 S
```

Example 3:

```
cd $XCOM_HOME/ezt

xcomesrp 0.5 /home/tester/xcom_history_record/XCOMHIST1
```

Example 4:

```
cd $XCOM_HOME/ezt

xcomesrp 1 E /home/tester/xcom_history_record/XCOMHIST2
```

Using Store and Forward

About Store and Forward

A store and forward transfer can be used to transfer a file indirectly when a destination system is not available. You can send a file to the mainframe, where it is stored temporarily. When the destination system becomes available, the file is automatically forwarded from the mainframe to the destination system.

This method is also used when you have two PU 2.0 devices that cannot communicate directly. They must use the mainframe as an intermediate device, because a UNIX or Linux system is not an intermediate device.

Note: A PU 2.1 device can handle direct transfers with another PU 2.1 device.

Example 1

In the following example, the xcom62 command is used. The file *mytest.tst* is sent to the intermediate z/OS node XCOMMVS2, which forwards the transfer to the remote Windows system defined in the *endest* destination member and into the file named *MYTEST.TST*.

```
xcom62 -c1 -f XIDEST=XCOMMVS2 REMOTE_SYSTEM=endest
LOCAL_FILE=mytest.tst REMOTE_FILE=MYTEST.TST
```

There must be a destination member created and enabled for the remote system in the z/OS Dynamic Control Library in order for CA XCOM Data Transport to forward the transfer successfully.

Example 2

In the following example, the xcomtcp command is used. The file *mytest.tst* is sent to the intermediate z/OS node XCOMMVS2, which forwards the transfer to the remote Windows system defined in the *endest* destination member and into the file named *MYTEST.TST.*.

```
xcomtcp -c1 -f XIDEST=XCOMMVS2 REMOTE_SYSTEM=endest
LOCAL_FILE=mytest.tst REMOTE_FILE=MYTEST.TST
```

There must be a destination member created and enabled for the remote system in the z/OS Dynamic Control Library in order for CA XCOM Data Transport to forward the transfer successfully.

When to Use

Use the XIDEST parameter with the REMOTE_SYSTEM parameter for store and forward transfers through an IBM mainframe system. Use store and forward to send an indirect transfer or to transfer files between two PU 2.0 devices.

Parameter

The store and forward parameter is shown below.

XIDEST

Specifies the name of the remote system on the intermediate destination that is designated for store-and-forward transfers. If this variable is null or unset, then a direct connection to a remote system is attempted.

Note: For store-and-forward transfers only.

Range: 0 to 14 characters

Default: None

Setting Password and User ID Security

In general, a remote system requires a password and user ID before CA XCOM Data Transport can perform transfers. The access rights of the user ID on the remote system determine the actions permitted for the transfer.

Example 1

In the following example, the xcom62 command is used. *myname* is the USERID and *mypassword* is the PASSWORD.

```
xcom62 -c1 -f USERID=myname PASSWORD=mypassword
```

All other parameters are read from the default configuration file, xcom.cnf.

Example 2

In the following example, the xcomtcp command is used. *myname* is the USERID and *mypassword* is the PASSWORD.

```
xcomtcp -c1 -f USERID=myname PASSWORD=mypassword
```

All other parameters are read from the default configuration file, xcom.cnf.

Parameters

The security parameters are shown in this section.

DOMAIN

The Windows domain name for use in authenticating the user ID and password when accessing a Windows-based machine that has sharable disks and drives that belong to that domain. This allows users to access these sharable drives without having to have a local user ID or password defined to the machine.

Range: 1 to 15 characters

Default: None

PASSWORD

The password associated with the user ID on a remote system.

Range: 0 to 31 characters

Default: None

TRUSTED

Allows the user to request a trusted transfer and the partner's CA XCOM Data Transport TRUSTED database to be searched to verify the user's credentials. This eliminates the need for the user to specify a USERID and PASSWORD. If XCOM_TRUSTED_OVR is set to NO or no USERID is specified, the USERID of the process that initiated the transfer is used.

Note: TRUSTED=YES cannot be specified with indirect transfers, because this is not supported.

Range: YES, NO, Y, N

Default: NO

USERID

The user ID that the security system on the remote system checks before granting access for the file transfer.

Range: 0 to 12 characters

Default: None

Encrypting Parameter Values in Existing Configuration Files

Use XCOMENCR to encrypt selected parameter values in an existing configuration file. When XCOMENCR encounters a line with #!ENCRYPT, it changes the next non-comment line from

PARAMETER=VALUE

to

PARAMETER.ENCRYPTED=ENCRYPTEDVALUE

Example

Before encryption:

PASSWORD=*ENIGMA*

After encryption:

PASSWORD.ENCRYPTED=*12 0F 36 79 65 AB D0 37 ...*

Default: Up to 32 hexadecimal characters

Syntax

The syntax for using XCOMENCR is as follows:

xcomencr *input_file output_file*

Options

The options for XCOMENCR are as follows:

- (minus sign)

Send output to stdout.

Example: xcomencr *input_file* -

+ (plus sign)

Replace input_file.

Example: xcomencr *input_file* +

no options

Displays help text.

Procedure

To encrypt a parameter value using XCOMENCR

1. Using a text editor, open the configuration file you want to modify, go to the parameter you want to encrypt, create a blank line above it, and enter the following:

   ```
   #!ENCRYPT
   ```

   Repeat as necessary for each parameter you want to encrypt.

   Note: Because # denotes a comment line in the .cnf file, any line beginning with #!ENCRYPT is ignored by CA XCOM Data Transport and is used only by XCOMENCR.

2. Save the configuration file as an ASCII text file.

3. At the command prompt, enter the following:

   `xcomencr` *`input_file output_file`*

   input_file

   Is the name of the configuration file from Step 1.

   The parameter value in the first non-comment line after each occurrence of the #!ENCRYPT statement is changed to the encrypted parameter value format.

   Note: If you open an encrypted configuration file with a text editor, you would not be able to see the values of the encrypted parameters. In the GUI, when you open an encrypted configuration file, you can see the encrypted values, except for the user ID and password security parameters.

Changing an Encrypted Value

To change an encrypted value that is already specified

Delete all text after the parameter name and specify it again.

Example

If the line you want to change is

```
PASSWORD.ENCRYPTED=12 0F 36 79 65 AB D0 37 ...
```

you would have to delete

```
.ENCRYPTED=12 0F 36 79 65 AB D0 37 ...
```

Then type in an equal sign and the new parameter value, (in an unencrypted form) replacing *NEWVALUE* with your desired value, as follows:

```
PASSWORD=NEWVALUE
```

Then save the file as ASCII text and encrypt it as in the encryption procedure above.

Queuing Locally Initiated Transfers

The queue contains the information that CA XCOM Data Transport uses to perform a locally initiated transfer.

Note: The xcomqm command provides access to information about the transfers in the queue. For information about using xcomqm, see Managing the CA XCOM Data Transport Queue in the chapter "Operating Environment."

Example 1

In the following example, the xcom62 command is used. The transfer request goes into the queue to execute on July 28, 2012 at one minute after noon.

```
xcom62 -c1 -f QUEUE=YES START_DATE=07/28/12 START_TIME=12:01:00
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Example 2

In the following example, the xcomtcp command is used. The transfer request goes into the queue to execute on July 28, 2012 at one minute after noon.

```
xcomtcp -c1 -f QUEUE=YES START_DATE=07/28/12 START_TIME=12:01:00
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Parameters

The queue management parameters are shown below.

PRIORITY

Indicates the priority that CA XCOM Data Transport uses for scheduling a transfer. If two transfers are scheduled for the same time, the one with the high priority is processed before one with a normal or low priority.

HIGH

Set high priority.

NORMAL

Set medium priority.

LOW

Set low priority.

Default: NORMAL

QUEUE

Indicates whether to execute the transmission request immediately or to allow the request to be queued. If the user does not specify a .cnf file, and has not changed a .cnf file, the default value is YES.

Note: If NO is specified and the remote system is unavailable, the request aborts. If YES is specified, START_TIME and START_DATE are read.

YES

The transfer request goes into a queue and executes depending on the traffic in the queue and START_DATE and START_TIME.

NO

The transfer starts immediately.

Default: YES

Important! If you load a .cnf file into the CA XCOM Data Transport GUI, the value (QUEUE=YES or QUEUE=NO) specified in the .cnf file is shown correctly in the generated xml file. However, when you submit the transfer from the CA XCOM Data Transport GUI, it is always treated as though QUEUE=YES.

START_DATE

Indicates the date on which the transfer becomes eligible for execution. The format is mm/dd/yy. If this field is blank, the current date is used.

Example:

A value of 02/28/13 indicates February 28, 2013 as the start date.

Format: mm/dd/yy

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

START_TIME

Indicates the time when the transfer becomes eligible for execution. The military format of *hh*:*mm*:*ss* is used. If this field is blank, then the current time is used.

Example

A value of 14:00:00 indicates 2 p.m. as the start time.

Format: *hh*:*mm*:*ss*

Default: None

Notifying of Transfer Completion

Using notification parameters, CA XCOM Data Transport automatically notifies a local system, a remote system, or a particular user that a transfer is complete without the user having to monitor the queue manually. Notification parameters allow you to specify one of the following levels of notification: Complete, Warning, or Error.

Example 1

The following example uses the xcom62 command. When the transfer completes, local user user1 is notified with a mail message, and remote user *USER2* is notified with a TSO message if the transfer receives an error.

```
xcom62 -c1 -f LOCAL_NOTIFY=user1 NOTIFYL=MAIL
NOTIFY_NAME=USER2 NOTIFYR=TSO RMTNTFYL=E LCLNTFYL=A
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Example 2

The following example uses the xcomtcp command. When the transfer completes, local user *user1* is notified with a mail message, and remote user *USER2* is notified with a TSO message if the transfer receives an error.

```
xcomtcp -c1 -f LOCAL_NOTIFY=user1 NOTIFYL=MAIL
NOTIFY_NAME=USER2 NOTIFYR=TSO RMTNTFYL=E LCLNTFYL=A
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Parameters

The notification parameters are shown below.

LOCAL_NOTIFY

Specifies the user on the local system who is to be notified that CA XCOM Data Transport has completed a transfer. CA XCOM Data Transport uses the NOTIFYL parameter to determine the type of notification to use.

Range: 0 to 64 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LCLNTFYL

Specifies the local user notification level.

A or ALL

Notify on transfer completion.

W or WARN

Notify only if the transfer received a warning or error.

E or ERROR

Notify only if the transfer received an error.

Default: ALL

NOTIFYL

The local user notification flag.

WRITE

A message is displayed on the workstation where the user is logged in.

MAIL

A mail message will be sent to the user.

NONE

No notification is sent.

ALL

A message is displayed on all workstations attached to the server.

Default: None

Note: The L in NOTIFYL indicates that the local system governs the processing of the resulting notification on that system.

NOTIFY_NAME

The user on the remote system who is to be notified when CA XCOM Data Transport completes a transfer.

Note: If the remote system is an IBM mainframe, CA XCOM Data Transport uses the value of NOTIFYR to determine the type of notification to deliver.

If the remote system is a UNIX or Linux system, the user receives a mail message.

Range: 0 to 12 characters

Default: None

NOTIFY_TERM

Specifies which terminals to write to if NOTIFYL=WRITE. If NOTIFY_TERM is not set, all users specified in LOCAL_NOTIFY are notified at the first terminal where they are logged in, as found in the system table.

Range: 0 to 256 characters

Default: None

NOTIFYR

Specifies the remote user notification type when sending data to a remote system.

WRITE

A message is displayed on the screen.

MAIL

A mail message is sent to the user.

TSO

The specified TSO user is notified.

WTO

CA XCOM Data Transport writes to the log only (WTO).

CICS

The specified CICS user is notified.

LU

The specified Logical Unit is notified.

ROSCOE

Notify CA Roscoe user.

NONE

No notification is sent.

ALL

Write to all users.

Default: None

Note: The R in NOTIFYR indicates that the remote system governs the processing of the resulting notification on that system.

RMTNTFYL

Specifies the remote user notification level when sending data to a remote system.

A (ALL)

Notify on transfer completion.

W (WARN)

Notify only if the transfer received a warning or error.

E (ERROR)

Notify only if the transfer received an error.

Default: ALL

Checkpointing and Restarting Transfers

By setting the parameters described in this section, a transfer is marked for checkpointing and automatic restart if an interruption occurs during the transfer attempt. After an interruption the daemon scans its table of transfers and automatically attempts to restart eligible transfers.

The CHECKPOINT_COUNT parameter indicates where to restart the transfer after it has been interrupted. If a transfer is interrupted, it is resumed from a checkpoint rather than starting over from the beginning of the file. When CHECKPOINT_COUNT=1000, CA XCOM Data Transport performs a check after every 1000 records sent, to ensure that the records are stored on the destination system. Then the next 1000 records are transmitted.

The restart parameters automatically restart a transfer after a transmission error, based on the number of retries and the retry time specified. Transmission conditions that permit restarts generally include most transmission errors. Restarts will also be attempted when a transfer is suspended by the remote system, and when a link goes down. Conditions that are not considered for restart include errors such as invalid passwords and "data set not cataloged" messages.

Note: Restart parameters can be specified without setting a checkpoint.

Important! Checkpointing impacts performance. Set the CHECKPOINT_COUNT as high as possible, or turn off checkpointing if it is not needed.

Example 1

In the following example, the xcom62 command is used. If the transfer is interrupted, it restarts from the most recent checkpoint of every 2000 records, and up to three attempts are made to retry the transfer at intervals of 30 seconds.

```
xcom62 -f CHECKPOINT_COUNT=2000 NUMBER_OF_RETRIES=3
RESTART_SUPPORTED=YES RETRY_TIME=30
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Example 2

In the following example, the xcomtcp command is used. If the transfer is interrupted, it restarts from the most recent checkpoint of every 2000 records, and up to three attempts are made to retry the transfer at intervals of 30 seconds.

```
xcomtcp -f CHECKPOINT_COUNT=2000 NUMBER_OF_RETRIES=3
RESTART_SUPPORTED=YES RETRY_TIME=30
```

All other necessary parameters are read from the default configuration file, xcom.cnf.

Parameters

The checkpoint and restart parameters are shown in this section.

CHECKPOINT_COUNT

Defines how often (based on record count) the sending system requests a checkpoint to be taken. The value 0000 indicates no checkpointing.

Range: 0 to 9999

Default: 1000

Notes:

- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX
- This is a Version 2 parameter.

NUMBER_OF_RETRIES

Maximum number of retries before a transfer is logged as failed and taken out of the transfer queue. If the value is 0, no retries are attempted.

Range: 0 to 255

Default: 1

Note: This is a Version 2 parameter.

RESTART_SUPPORTED

Specifies whether automatic restart is to be supported on a transfer.

Range: YES or NO

Default: YES

Note: This is a Version 2 parameter.

RETRY_TIME

The number of seconds between retries of unsuccessful transfers.

Range: 0 to 99999

Default: 1

Note: This is a Version 2 parameter.

Chapter 4: Application Programming Interface

The Application Programming Interface (API) is intended to give programmers the capability of developing their own CA XCOM Data Transport applications.

There are two CA XCOM Data Transport APIs, as follows:

- XcomAPI
- XcomQAPI

The XcomAPI provides the ability to submit transfers to CA XCOM Data Transport. The XcomQAPI provides the same functionality as the XCOMQM command line, but through an API.

There are also two utility APIs that work with these two APIs, as follows:

- msgstring
- XsetSystemPriv

This section contains the following topics:

XcomAPI (see page 259)
XcomQAPI (see page 260)
Utility Functions (see page 260)
Using API Member Names (see page 260)
Upgrade from Previous Releases (see page 320)
Using XcomQAPI Option Parameters (see page 326)
API Examples (see page 329)
XcomQAPI Return Codes (see page 339)

XcomAPI

The XcomAPI routine uses C structures defined in the xcomapi.h file to pass information to CA XCOM Data Transport. The CA XCOM Data Transport XcomAPI function call takes a starting state and a parameter block structure as its arguments. For transfers using SNA, before invoking XcomAPI, an SNA session must be active with the desired remote system.

xcomapi.h can be located at $XCOM_HOME/api/include

XcomQAPI

The XcomQAPI routine uses C structures defined in the xcomapi.h file to return information from the CA XCOM Data Transport Queue. The CA XCOM Data Transport XcomQAPI function call takes a parameter field as its argument, which allows you to delete entries from the queue, suspend a transfer, resume a suspended transfer, display the list of entries in the queue, and display detailed information about the queue entries.

Utility Functions

msgstring

The msgstring API is a utility API. It converts a return code to an appropriate message. The CA XCOM Data Transport msgstring API() function call takes a return code and sense as its arguments. The return code is the return value of XcomAPI or XcomQAPI.

Example:

Suppose XcomAPI or XcomQAPI returns a code other than zero. This return code is passed to the msgstring API, which returns a message corresponding to the return code.

XsetSystemPriv

XsetSystemPriv is a utility API that allows xcomtcp and xcom62 to return to system privileges (user=root, group=xcomadm) after a transfer has been completed. The CA XCOM Data Transport XsetSystemPriv API() function call takes a type of privilege as its argument.

This function needs to be called at the end of transfer. Refer to API example to know usage of this function.

Using API Member Names

The member names below are for use by applications programmers to pass information in a C structure to the XcomAPI routine.

Member Names

The member names in the following list are used to create C structures that specify aspects about a transfer. The appropriate information is then passed to CA XCOM Data Transport by the XcomAPI routine.

The XcomAPI member names are listed below in alphabetical order.

allocation_type

This parameter indicates the unit of storage allocation for a dataset that was created on an IBM mainframe.

CYL

Cylinders

TRK

Tracks

BLK

Blocks

REC

Records

Default: CYL

Note: This parameter is a Version 2 parameter.

avgrec

For a data set created on an IBM mainframe, this parameter specifies the multiplier for Primary and Secondary allocation units when allocating based on the number of records. The record size is based on the value of the LRECL parameter.

U

Indicates that the PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records to allocate for.

K

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in thousands (so it would be the number specified multiplied by 1024).

For example, specifying 3 would be stating 3 K or 3072 records.

M

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in millions (so it would be the number specified multiplied by 1048576).

For example, specifying 2 would be stating 2M or 2097152 records.

Default: None

Note: This parameter applies only when the SPACE parameter specifies a value of “REC”. The parameter indicates that a file is being allocated based on a specific number of records.

blksize

Specifies the block size of the data set created on an IBM mainframe.

Range: 0 to 32767

If the record format is	Then the block size must be
Fixed or fixed block	A multiple of the record length
Variable	Four bytes larger than the record length
Undefined	Larger than the largest record length. A maximum of five digits can be entered

Default: 800

carriage_control_characters

Indicates the type of printer carriage control codes, if any, that are included in the report file.

Note: For report transfers only.

ASA

ASA control codes in column 1.

IBM

IBM Machine Characters (valid only for IBM mainframes).

BYPASSASA

If data is already in ASA format, bypass conversion.

OTHER

No carriage-control codes are used.

Default: OTHER

carriage_flag

Specifies the type of file being transferred and some special characteristics of the conversion done during the transfer.

YES

Indicates that the transferred file is a text file and a newline character should be added to the end of incoming records. Also, newline characters are removed from the ends of lines before an outgoing record is sent.

NO

Indicates no special processing.

MPACK

Indicates a text file with record packing. Uses 2K pack buffer.

VLR

Indicates a binary file of variable-length records with a field of 4 bytes preceding each record. Applies to local initiated transfers only.

XPACK

Indicates a text file with record packing. Uses 31K pack buffer.

Note: MPACK does not support a MAXRECLEN (actual record length) over 2K. XPACK does not support a MAXRECLEN (actual record length) over 31K.

Default: YES

checkpoint_count

Defines how often (based on record count) the sending system requests a checkpoint to be taken. The value 0000 indicates no checkpointing.

Range: 0 to 9999

Default: 1000

Notes:

- CA XCOM Data Transport automatically sets checkpoint_count to 0 if any of the following parameter values are set:
  - (l)ear_cipher=XXX-OFB
  - (l)ear_cipher=RC4
  - (l)ear_hash=XXX
- This is a Version 2 parameter.

class

The print class assigned to a report transferred to a remote system.

If the remote system is an IBM mainframe, this field designates the JES SYSOUT class.

Note: For report transfers only.

Example:

Enter **B** to print the report through SYSOUT=B.

Range: 1 character

Default: None

code_flag

Used to identify the type of data being transferred.

ASCII

An ASCII file is being transferred. This indicates that the incoming file is assumed to be ASCII format, and is not translated. Therefore the file on the remote system should be in ASCII format before it is transferred.

BINARY

A binary file, such as an executable file, is being transferred. This indicates to a remote system that it is not to translate the data it is exchanging with your system.

EBCDIC

An EBCDIC file is being transferred. The transferred data is translated from EBCDIC to ASCII format when the local system receives the data, and from ASCII to EBCDIC format when the local system sends the data.

UTF8

A Unicode file which is based on the UTF8 encoding system is being transferred. LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

UTF16

A Unicode file which is based on the UTF16 encoding system is being transferred. LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

Default: ASCII

codetabl

Specifies the prefix to the file names, atoe.tab and etoa.tab, that contain the external ASCII-to-EBCDIC and EBCDIC-to-ASCII custom character conversion tables. These custom character conversion tables determine which external translation tables will be used by the transfer.

This parameter is valid only if INTERNAL_CONVERSION_TABLES=NO.

Range: 0 to 12 characters

Default: None

compress

Indicates the compression type.

Notes:

- When communicating with an IBM mainframe, if the data file contains any empty lines, COMPRESS can be set to YES, LZSMALL, LZMEDIUM, or LZLARGE only.
- Not all compression types are supported on all platforms. For supported compression types, see the partner platform documentation.
- The LZ values enable LZ (Lempel-Ziv) compression to replace sequences of data bytes that occur more than once in a datastream with a code value.

COMPACT

RJE compaction algorithm optimized for uppercase English text.

LCOMPACT

RJE compaction algorithm optimized for lowercase English text.

LZLARGE

Activates LZ compression to search back 32K in the datastream for a matching string.

LZMEDIUM

Activates LZ compression to search back 16K in the datastream for a matching string.

LZRW3

General-purpose algorithm that runs fast and gives reasonable compression.

LZSMALL

Activates LZ compression to search back 4K in the datastream for a matching string.

NO

Indicates no compression.

RLE

Run length encoding.

YES

Indicates run length encoding of binary zeros and blanks only.

ZLIB*n*

Greater compression than LZRW3 but less than LZSMALL, LZMEDIUM, and LZLARGE. The value *n* can be 1 through 9.

HUFFMAN

Huffman compression technique will be used.

Default: YES

compress_pds

The compress_pds parameter controls when a PDS dataset gets compressed for CA XCOM Data Transport z/OS partners that support this functionality. The value of the CMPRS_PDS_ALLOW parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMDFLT) determines if PDS compression is allowed. When this parameter is set to YES, then PDS compression can occur on the CA XCOM Data Transport z/OS partner.

NONE

Suppresses the compression of an output PDS dataset as part of a CA XCOM Data Transport transfer.

BEFORE

The output PDS dataset is compressed before the transfer of user data begins.

AFTER

The output PDS dataset is compressed after the transfer of user data has completed.

BOTH

The output PDS dataset is compressed both before and after the transfer of user data.

Default: NONE

Note: COMPRESS_PDS applies only to PDS data sets that are opened for output as the target of a CA XCOM Data Transport transfer.

configssl

This parameter specifies the configssl.cnf file path and file name.

Range: 1 to 256 characters

Default: $XCOM_HOME\config\configssl.cnf

Note: $XCOM_HOME is an environment variable.

convert_classes

A character string containing print classes for which EBCDIC-to-ASCII conversions are performed. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

copies

The number of copies to be sent. If this parameter is not specified, the remote system queues one copy of the report to the system's default printer. For report transfers only.

Range: 1 to 999

Default: 1

createdelete

Specifies whether an existing data set can be deleted and a new data set allocated at the start of a FILEOPT=CREATE transfer. This functionality is for CA XCOM Data Transport z/OS partners with support for CREATEDELETE. The CREATEDELETE parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMCNTL) specified by the z/OS CA XCOM Data Transport Administrator affects the functionality of this parameter.

When z/OS has CREATEDELETE=ALLOW:

YES

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

NO

If FILEOPT=CREATE and the data set exists, then the transfer fails with a catalog/file error.

When z/OS has CREATEDELETE=YES:

YES or NO

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

When z/OS has CREATEDELETE=NO:

YES or NO

If FILEOPT=CREATE and the data set exists, then the transfer fails with a catalog/file error.

Default: NO

Notes:

- The attributes of the existing dataset are deleted and the new data set is allocated with the attributes specified in the transfer when CREATEDELETE=YES.
- CREATEDELETE applies only if the target data set is a sequential data set or an entire PDS/PDSE. CREATEDELETE is ignored for other types of data sets (such as PDS members, PDSE members, VSAM, and USS files).
- CREATEDELETE applies to relative GDGs when the dataset is specified using the fully qualified GxxxxVxx name.

dataclas

Specifies the name of the data class to be used when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

debug_flag

This parameter specifies whether a trace will be output to the file /tmp/<tid> instead of to the default path $XCOM_HOME/trace/<tid>.TRA. In addition to the transfer trace information, the initialization will also be traced. The trace level is still taken from the variable XTRACE.

Note:

- <tid> refers to the six-digit transfer ID number.
- $XCOM_HOME is an environment variable.

YES

The trace is output to /tmp/<tid>

NO

The trace is output to $XCOM_HOME/trace/<tid>.TRA.

Default: NO

den

Specifies the density to be used in creating a tape on the local or remote system. Valid values are the same as those for the DEN parameter in JCL.

Range: 1 to 4

Default: None

destination

Identifies the printer or other device on the remote system where the report is to be sent. If this parameter is not specified, the remote system sends the report to the system's default printer. For report transfers only.

0 to 16 characters

For indirect transfers and for Version 1.

0 to 21 characters

For transfers that are not indirect and for Version 2.

Default: None

disposition

Indicates what the remote system does with the report file after the report has been printed. For report transfers only.

Note: This field is not used when the remote system is an IBM mainframe.

DELETE

After printing the report is deleted.

KEEP

After printing the report is kept.

HOLD

After printing the report is held.

Default: DELETE

domain

The Windows domain name for use in authenticating the user ID and password when accessing a Windows based machine that has sharable disks and drives that belong to that domain. This allows users to access these sharable drives without having to have a local user ID or password defined to the machine.

Range: 1 to 64 characters

Default: None

dsntype

This parameter specifies the data set definition.

Note: This parameter applies only to mainframe SMS data sets.

LIBRARY

Defines a PDSE.

PDS

Defines a partitioned data set.

BASIC

Defines a legacy sequential dataset.

LARGE

Defines a large format sequential dataset.

EXTREQ

Defines an extended format dataset.

EXTPREF

Specifies an extended format is preferred. If the extended format is not possible, a basic format is used.

<blank>

Defines a partitioned or sequential data set based on the data set characteristics entered.

Note: These values are IBM standards for SMS processing.

Range: One to eight characters

Default: None

ear_cipher

Indicates which encryption algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
DES-CBC DES-ECB DES-CFB DES-OFB DES	YES/NO	Algorithm uses fixed key size (8 bytes)
3DES-CBC 3DES-ECB 3DES-CFB 3DES-OFB 3DES	YES/NO	Algorithm uses fixed key size (24 bytes)
AES128-CBC AES128-ECB AES128-CFB AES128-OFB AES128	YES/NO	Algorithm uses fixed key size (16 bytes)
AES192-CBC AES192-ECB AES192-CFB AES192-OFB AES192	YES/NO	Algorithm uses fixed key size (24 bytes)
AES256-CBC AES256-ECB AES256-CFB AES256-OFB AES256	YES/NO	Algorithm uses fixed key size (32 bytes)
RC2-CBC RC2-ECB RC2-CFB RC2-OFB RC2	NO	16-byte key size supported
RC4-STREAM RC4	NO	16-byte key size supported

Example:

```
ear_cipher=3DES-CBC
```

Notes:

- This is one of a pair of parameters, as follows:
  - lear_cipher (local)
  - ear_cipher (remote)
- The following values are the same as each other:
  - RC2 is the same as RC2-CBC
  - RC4 is the same as RC4-STREAM
  - AES128 is the same as AES128-CBC
  - AES192 is the same as AES192-CBC
  - AES is the same as AES256 and AES256-CBC
  - DES is the same as DES-CBC
  - 3DES is the same as 3DES-CBC

ear_digest

Indicates that the digest value matched with the generated digest, when the transfer type is a receive file. You must have already generated a digest value on the input file.

A send file transfer, CA XCOM Data Transport generates the digest on the remote file. CA XCOM Gateway uses the generated digest to ensure the integrity of the file in the staging area.

Range: 0 to 128 characters

Default: None

Example:

```
ear_digest=CEA700113B151ECF63312EE6C6CFC2260AAE5FAB
```

Notes:

- The digest value specified in the example is the digest that is generated using the SHA1 digest algorithm.
- XCOM accepts the hexadecimal value of the actual digest.
- This value is one of a pair of parameters, as follows:
  - lear_digest (local)
  - ear_digest (remote)

ear_hash

Indicates which hash algorithm is used.

Default: None

Value	FIPS MODE	Comments
SHA1	YES	Secure Hash Algorithm 1
MD5	NO	Message Digest-5 hash algorithm
MD4	NO	Message Digest-4 hash algorithm

Note:

- This value is one of a pair of parameters, as follows:
  - lear_hash (local)
  - ear_hash (remote)
- CA XCOM Data Transport automatically sets checkpoint_count to 0 if any of the following parameter values are set:
  - (l)ear_cipher=XXX-OFB
  - (l)ear_cipher=RC4
  - (l)ear_hash=XXX

ear_key

Indicates the symmetric encryption key that is used to encrypt/decrypt.

Range: 0 to 128 characters

Default: None

Example:

```
ear_key=025BD5549BF743D99D86A2752019BCF48CC14AD5A16E0110
```

Notes:

- The key value specified in this example is a key for the 3DES encryption algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual key and the specified value must be in upper case.

eattr

This parameter identifies if the dataset can have extended attributes when the dataset is allocated on an Extended Address Volume (EAV).

OPT

Specifies that a dataset can optionally have extended attributes.

NO

Specifies that a dataset cannot have extended attributes.

Default: None

Note: This parameter is applicable only for data set creation on an IBM mainframe.

eol_classes

A character string containing print classes for which an ASCII newline character will be appended to each record. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

expdt

Specifies an expiration date for the tape data set in terms of a two-digit designation for the year and a three-digit designation for the day of the year. For example, in the expiration date 13021, 13 is the year, that is, 2013, and 021 is the 21st day of that year when the tape data set expires.

Format: *yyddd*

Default: None

Note: EXPDT and RETPD are mutually exclusive; specify one or the other.

fips_mode

Specify the mode to use for encryption/decryption.

YES

Indicates that encryption/decryption will be running in FIPS mode.

NO

Indicates that encryption/decryption will not be running in FIPS mode.

Default: NO

fcb

Identifies the FCB JCL parameter when sending the report file to an IBM mainframe, defining print density, lines per page, and so on. For report transfers only.

Range: Zero to four characters

Default: None

file_option

Indicates how the transferred data is to be processed by the receiving system. For file transfers only.

CREATE

Create a new file on the receiving system.

APPEND

Append the transferred data to an existing file on the receiving system.

REPLACE

Replace an existing file on the receiving system.

Default: CREATE

Note: When creating a file on an IBM mainframe system, some additional information may be necessary. For more information, see the explanations for RECORD_FORMAT, BLKSIZE, VOLUME, and UNIT parameters.

file_type

Indicates the type of transmission.

SEND_FILE

File transfer

SEND_REPORT

Report to be printed

SEND_JOB

Job

RECEIVE_FILE

Retrieval of a file

form

The type of form that should be used to print the report. Since CA XCOM Data Transport places the print job in the remote system's print queue, the print control functions depend on the remote system. Verify beforehand that the requested form is available at the remote site. For report transfers only.

Note: When sending a report to a VAX computer, leave this parameter blank unless you are certain that this is a valid form type. VMS interprets this to mean that no special form is being requested.

Range: 0 to 10 characters

Default: None

gatewayguid

Identifies the remote file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. This is a unique value that identifies each CA XCOM Gateway file.

Note: When the CA XCOM Gateway GUID is not known, the keyword ANY can be used to identify the remote file as a CA XCOM Gateway file.

Range: 0 to 36 characters

Default: None

hold

Indicates whether the transferred report file is to be placed on HOLD status on the remote system or is to be readied for immediate printing. Specify Y to request that it be held (spooled) on an MVS system.

Range: YES or NO

Default: NO

hold_transfer

This parameter prevents a TYPE=SCHEDULE transfer from starting until explicitly released.

YES

This transfer is not initiated until it is released in one of the following ways:

- Through the CA XCOM Data Transport GUI
- By an operator with a RELEASE command
- With the command, xcomqm -Annnnnn where nnnnnn is the TID number (Transaction ID).

Note: The XCOMQM command provides access to information about the transfers in the queue. For information about using XCOMQM, see the chapter "Operating Environment" earlier in this guide.

- Through the XTC facility

NO

This transfer is not held.

Default: NO

label

Indicates the type of label associated with a tape data set.

Processing type: AL, AUL, BLP, LTM, NL, NSL, SL, SUL

Represents the type of processing to be applied to data sets on tape.

Note: CA XCOM Data Transport for z/OS supports only standard label tapes.

Default: AL

labelnum

Indicates the sequence number of the data set on the tape.

Sequence number (0001-9999)

This value identifies the sequence number of a data set on tape.

Default: 0001

lclntfyl

Specifies the local user notification level.

A or ALL

NOTIFY on transfer completion.

W or WARN

NOTIFY only if the transfer received a warning or error.

E or ERROR

NOTIFY only if the transfer received an error.

Default: ALL

lear_cipher

Indicates which encryption algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
DES-CBC DES-ECB DES-CFB DES-OFB DES	YES/NO	Algorithm uses fixed key size (8 bytes)
3DES-CBC 3DES-ECB 3DES-CFB 3DES-OFB 3DES	YES/NO	Algorithm uses fixed key size (24 bytes)
AES128-CBC AES128-ECB AES128-CFB AES128-OFB AES128	YES/NO	Algorithm uses fixed key size (16 bytes)
AES192-CBC AES192-ECB AES192-CFB AES192-OFB AES192	YES/NO	Algorithm uses fixed key size (24 bytes)

Value	FIPS MODE	Comments
AES256-CBC AES256-ECB AES256-CFB AES256-OFB AES256	YES/NO	Algorithm uses fixed key size (32 bytes)
RC2-CBC RC2-ECB RC2-CFB RC2-OFB RC2	NO	16-byte key size supported
RC4-STREAM RC4	NO	16-byte key size supported

Example:

```
lear_cipher=3DES-CBC
```

Notes:

- This is one of a pair of parameters, as follows:
  - lear_cipher (local)
  - ear_cipher (remote)
- CA XCOM Data Transport automatically sets checkpoint_count to 0 if any of the following parameter values are set:
  - (l)ear_cipher=XXX-OFB
  - (l)ear_cipher=RC4
  - (l)ear_hash=XXX

lear_digest

Indicates the digest value that is matched with the generated digest, when the transfer type is a send file, job, or report. You must have already generated a digest value on the input file.

A receive file transfer; CA XCOM Data Transport generates the digest on the local file. CA XCOM Gateway uses the generated digest to ensure the integrity of the file in the staging area.

Range: 0 to 128 characters

Default: None

Example:

```
lear_digest=CEA700113B151ECF63312EE6C6CFC2260AAE5FAB
```

Note:

- The digest value specified in the example is the digest that is generated using the SHA1 digest algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual digest and the specified value must be in upper case.
- This value is one of a pair of parameters, as follows:
  - lear_digest (local)
  - ear_digest (remote)

lear_hash

Indicates which hash algorithm is used.

Default: None

Value	FIPS MODE	Comments
SHA1	YES	Secure Hash Algorithm 1
MD5	NO	Message Digest-5 hash algorithm
MD4	NO	Message Digest-4 hash algorithm

Note:

- This value is one of a pair of parameters, as follows:
  - lear_hash (local)
  - ear_hash (remote)
- CA XCOM Data Transport automatically sets checkpoint_count to 0 if any of the following parameter values are set:
  - (l)ear_cipher=XXX-OFB
  - (l)ear_cipher=RC4
  - (l)ear_hash=XXX

lear_key

Indicates the symmetric encryption key that is used to encrypt/decrypt.

Range: 0 to 128 characters

Default: None

Example:

```
lear_key=025BD5549BF743D99D86A2752019BCF48CC14AD5A16E0110
```

Notes:

- The key value specified in this example is a key for the 3DES encryption algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual key and the specified value must be in upper case.

lgatewayguid

Identifies the local file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. The CA XCOM Gateway GUID is a unique value that identifies each CA XCOM Gateway file. The keyword ANY can be used to identify the local file as a CA XCOM Gateway file when the CA XCOM Gateway GUID is not known.

Range: 0 to 36 characters

Default: None

local_charset

This parameter specifies the local character set CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

Default: The DEFAULT_CHARSET parameter specifies the default in the CA XCOM Data Transport Global Parameters.

local_delim

This parameter specifies an optional encoding for which the specified LOCAL_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list. If LOCAL_CHARSET is UTF based charset, UTF rules will be applied disregarding encoding specified in LOCAL_DELIM.

Additionally, specifies a colon separated list of record delimiters that are used to mark and detect the end of a record.

This parameter is used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

EBCDIC Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
FF	
NA	

NOTE: The NA is only used by XCOM z/OS to indicate that the file format attribute should not be set.

ASCII Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: The DEFAULT_DELIM parameter specifies the default in the CA XCOM Data Transport Global Parameters specifies the default.

The tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which the LOCAL_CHARSET is based, it should not be used.

Example: If LOCAL_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if LOCAL_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in LOCAL_DELIM, delimiters will be derived as explained below.

If LOCAL_DELIM is not specified or doesn't contain at least one valid delimiter, the DEFAULT_DELIM parameter in the CA XCOM Data Transport Global Parameters is considered. If DEFAULT_DELIM is also not specified or doesn't contain at least one valid delimiter, platform default values are used as shown:

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

Note: During Unicode conversion some of the delimiters may fail to convert to some code pages.

For example, in the case of a Receive File transfer, when LOCAL_CHARSET is set to an ASCII based charset and LOCAL_DELIM is set to NL/CRNL, then the transfer may fail. The Unicode conversion fails as ASCII based code pages may not have a mapping character for NL/CRNL.

local_file

The name of the file on the local system that is being transferred. If this variable is null or unset, standard input is read. In this manner, CA XCOM Data Transport commands can be used in a pipeline or with redirection. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk as a file name to indicate that all files within the specified directory are to be transferred. For example, the statement "LOCAL_FILE=/NAMES/*" indicates that all files under the NAMES directory are to be transferred.

When a prefix is followed by an asterisk (*), all members beginning with a specific prefix are to be transferred. For example, LOCAL_FILE=/NAMES/AL* requests that files AL, ALEX, and ALICE are all to be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated. However, when sending files to a mainframe PDS, any file name over 8 characters in length is truncated. These systems do not recognize file extensions. For example, a file called *longfilename.txt* will be truncated to *longfile* and a file called *file.txt* will be truncated to *file*.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

local_file_rf

The file name that is created, appended, or replaced on the local system when it receives a file. At the command prompt or in a script, if this value is null or unset, then CA XCOM Data Transport writes to the stdout. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk (*) as a file name to indicate that multiple files will be received. For example, LOCAL_FILE_RF=/PAYROLL/*.

If multiple files are received and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

For platforms that support it, you can specify a common file extension to be appended to each file name. For example, LOCAL_FILE_RF=/PAYROLL/*.TXT.

Note: If QUEUE=YES, the full path name must be specified.

Range: 0 to 256 characters

Default: None

local_file_sj

Indicates the name of the file on the local system to be sent as a job. All the UNIX or Linux file naming conventions apply. If this value is null or unset, then CA XCOM Data Transport reads the standard input file.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

local_file_sr

Specifies the local file name to be sent as a report to the remote system. If this value is null or unset, then CA XCOM Data Transport reads the standard input file. For report transfers only.

Note: If QUEUE=YES, user must specify the full path name.

Range: 0 to 256 characters

Default: None

local_notify

Specifies the user on the local system who is to be notified that CA XCOM Data Transport has completed a transfer. CA XCOM Data Transport uses the NOTIFYL parameter to determine the type of notification to use.

Range: 0 to 64 characters

Default: None

lrecl

Specifies the actual or maximum length, in bytes, of a logical record. Corresponds to the JCL LRECL subparameter.

Range: 0 to 32767

If the format is	Then the maximum length of a logical record must be equal to the
Variable blocked record	Maximum record length plus 4
Fixed or fixed block record	Constant record length

Default: 160

maxreclen

For Windows, UNIX, and Linux systems, the locally initiating CA XCOM Data Transport system determines the values for MAXRECLEN, TRUNCATION, and LRECL, for send and receive operations. When the local CA XCOM Data Transport system initiates a transfer of a text file, this parameter designates the length, in bytes, of the largest record that can be transferred. If a record length is longer than this value, CA XCOM Data Transport uses the value in the TRUNCATION parameter on the initiating side to determine whether to terminate the transfer or to truncate the record and continue the transfer. When CA XCOM Data Transport transfers binary files, this value indicates the length of the records that are transferred. On a receive operation, MAXRECLEN is set to whatever the LRECL value is on the initiating side.

Range: 1 to 32767

Default: 1024

mbcs_converror

This parameter identifies the action when the input file contains characters that cannot be converted as they are not included within the output character sets repertoire.

REPLACE

Replace each unconvertible character with the default substitution character defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each unconvertible character with the Unicode character that the decimal value nnnnnnn identifies. (If the specified replacement character cannot be represented in the output character set, then the transfer is failed). This option is not supported for z/OS systems and is treated as REPLACE.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of skipped characters. This option is not supported for z/OS systems and is treated as REPLACE.

FAIL

The transfer terminates with an error condition.

Default: The DEFAULT_CONVERROR parameter in the CA XCOM Data Transport Global Parameters specifies the default.

mbcs_inputerror

This parameter identifies the appropriate action when the input file contains data that is not consistent with the specified input character set.

REPLACE

Replace each piece of erroneous data with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each piece of erroneous data with the Unicode character that the decimal value nnnnnnn identifies. This option is not supported for z/OS systems and is treated as REPLACE.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of ignored bytes. This option is not supported for z/OS systems and is treated as REPLACE.

FAIL

The transfer terminates with an error condition.

Default: The DEFAULT_INPUTERROR parameter in the CA XCOM Data Transport Global Parameters specifies the default.

metacode_classes

Classes of print jobs that are saved in metacode format, a variable length record format. For incoming report transfers only.

Range: 0 to 64 characters

Default: None

mgmtclas

Specifies the name of the management class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

notify_name

The user on the remote system who is notified when CA XCOM Data Transport completes a transfer.

Note: If the remote system is an IBM mainframe, CA XCOM Data Transport uses the value of NOTIFYR to determine the type of notification to deliver.

If the remote system is a UNIX or Linux system, the user receives an email message.

Range: 0 to 12 characters

Default: None

notifyl

The local user notification flag.

WRITE

A message is displayed on the workstation where the user is logged in.

MAIL

A mail message is sent to the user.

NONE

No notification is sent.

ALL

A message is displayed on all workstations attached to the server.

Default: None

notifyr

Specifies the remote user notification type when sending data to a remote system.

WRITE

A message is displayed on the screen.

MAIL

A mail message is sent to the user.

TSO

The specified TSO user is notified.

WTO

CA XCOM Data Transport writes to the log only (WTO).

CICS

The specified CICS user is notified.

LU

The specified Logical Unit is notified.

ROSCOE

Notify CA Roscoe user.

NONE

No notification is sent.

ALL

Write to all users.

Default: None

notify_term

Specifies terminals to write if NOTIFYL=WRITE. If NOTIFY_TERM is not set, all users specified in LOCAL_NOTIFY are notified at the first terminal where they are logged in as found in the system table.

Range: 0 to 256 characters

Default: None

number_of_retries

Maximum number of retries before a transfer is logged as failed and taken out of the transfer queue. If the value is 0, no retries are attempted.

Range: 0 to 255

Default: 1

Note: This is a Version 2 parameter.

num_of_dir_blocks

This parameter specifies the number of directory blocks to allocate for a dataset that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

password

The password associated with the user ID on a remote system.

Range: 0 to 31 characters

Default: None

primary_alloc

This parameter identifies primary storage allocation for a data set that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 1

Note: This parameter is a Version 2 parameter.

print_class

The print class assigned to a report transferred to a remote system. If the remote system is an IBM mainframe, this field designates the JES SYSOUT class.

Note: For report transfers only.

Range: 1 character

Default: None

Example:

Enter **B** to print the report through SYSOUT=B.

port

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 0 to 65535

Default: 8044

protocol

The type of communication protocols to use.

SNA

For transfers using SNA/APPC communication protocols

TCPIP

For transfers using TCP/IP communication protocols

Default: SNA

queue

Indicates whether to execute the transmission request immediately or to allow the request to be queued. If the user does not specify a .cnf file, and has not changed a .cnf file, the default value is YES.

Note: If NO is specified and the remote system is unavailable, the request aborts. If YES is specified, start_time and start_date are read.

YES

The transfer request goes into a queue and executes depending on the traffic in the queue and START_DATE and START_TIME.

NO

The transfer starts immediately.

Default: YES

rmtntfyl

Specifies the remote user notification level when sending data to a remote system.

A or ALL

NOTIFY on transfer completion.

W or WARN

NOTIFY only if the transfer received a warning or error.

E or ERROR

NOTIFY only if the transfer received an error.

Default: ALL

record_format

Specifies the record format of a dataset created on an IBM mainframe. This corresponds to the JCL RECFM subparameter.

Range: The range of values is listed in the following table:

Value	Description	Record Length	Comment
F	Fixed unblocked	The same length as the data set.	

Value	Description	Record Length	Comment
FA	Fixed unblocked ANSI	The same length as the data set.	Contains ISO/ANSI/FIPS control characters.
FB	Fixed blocked	Fixed.	Fixed record length with multiple records per block.
FBA	Fixed blocked ANSI	Fixed.	Multiple records per block where these records contain ISO/ANSI/FIPS control characters.
FBS	Fixed blocked spanned	Fixed.	Multiple records per block written as standard blocks.
FM	Fixed unblocked machine	The same length as the data set.	Contains machine code control characters.
FS	Fixed unblocked spanned	The same length as the data set.	Written as standard blocks where these records do not contain any truncated blocks or unfilled tracks.
U	Undefined	Undefined.	
V	Variable unblocked	Variable.	
VA	Variable unblocked ANSI	Variable.	Contains ISO/ANSI/FIPS control characters.
VB	Variable blocked	Variable.	Multiple records per block.
VBA	Variable blocked ANSI	Variable.	Multiple records per block where these records contain ISO/ANSI/FIPS control characters.
VBM	Variable blocked machine	Variable.	Multiple records per block where these records contain machine code control characters.
VBS	Variable blocked spanned	Variable.	May have multiple records per block where these records can span more than one block.
VM	Variable unblocked machine	Variable.	Contains machine code control characters.
VS	Variable unblocked spanned	Variable.	A record can span more than one block.

Default: VB

remote_file

Indicates the file on the remote computer to which the transferred data is being written. If you are creating the file (FILE_OPTION=CREATE), the file name must be consistent with the file naming conventions of the remote system. The local CA XCOM Data Transport system does not validate this name. The remote I/O system determines whether the file name is valid.

For wildcard transfers, use an asterisk (*) as a file name to indicate and to inform the receiving partner that multiple files will be sent. For example, REMOTE_FILE=/PAYROLL/*.

If multiple files are sent and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

An asterisk used to send files to an IBM mainframe system shows that all files are to be transferred to a partitioned data set (PDS).

For platforms that support it, you can specify a common file extension to append to each file name; for example, REMOTE_FILE=/PAYROLL/*.TXT.

Note: For send file transfers only.

Range: 1 to 256 characters

Default: None

remote_file_rf

Indicates the name of the file to be retrieved from the remote system.

For wildcard transfers, use an asterisk (*) as a file name to indicate that all files within the specified PDS/Directory should be transferred. For example, the statement REMOTE_FILE_RF=/NAMES/* indicates that all files under the NAMES directory should be transferred.

When a prefix is followed by an asterisk, all members beginning with a specific prefix are transferred. For example, REMOTE_FILE_RF=/NAMES/AL* requests that files AL, ALEX, and ALICE should be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated.

Note: For retrieve file transfers only.

Range: 1 to 256 characters

Default: None

remote_system

The name of the remote system that receives a file, job or report.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

remote_charset

This parameter specifies the remote character set CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

When the remote system is CA XCOM Data Transport for z/OS:

CCSID#nnnnn/tttttttt

nnnnnn - specifies the CCSID number that corresponds to the character set of the local file. Valid values are 1 - 65535.

tttttttt (optional) – specifies the technique search order IBM Unicode Services uses when performing conversion. From 1 to 8 characters are specified. Valid values to use are:

R - Roundtrip conversion

E - Enforced Subset conversion

C - Customized conversion

L - Language Environment Behavior conversion

M - Modified for special use conversion

B - Bidi transformation (Bi-directional) conversion

0-9 - User defined conversions

Note: If the technique search order is not specified, Unicode Services defaults to 'RECLM'.

When the remote system is CA XCOM Data Transport for Windows or CA XCOM Data Transport for Linux/Unix:

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

remote_delim

This parameter specifies an optional encoding for which the specified REMOTE_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list. If REMOTE_CHARSET is a UTF based charset, UTF rules will be applied disregarding the encoding specified in REMOTE_DELIM.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

When the remote system is CA XCOM Data Transport for z/OS:

Additionally it specifies the delimiter to use for USS-based output files when FILEDATA=TEXT.

Valid options:

EBCDIC – The specified character set is EBCDIC encoded.

NA – Not applicable, the system default delimiter is used.

NL – New line

CR – Carriage return

LF – Line feed

CRLF – Carriage return/Line feed

LFCR – Line feed/Carriage return

CRNL – Carriage return/New line

Note: The only valid encoding for CA XCOM Data Transport for z/OS is EBCDIC.

When the remote system is CA XCOM Data Transport for Windows or CA XCOM Data Transport for Linux/Unix:

Additionally, specifies a colon separated list of record delimiters that are used to mark and detect the end of a record.

EBCDIC Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF

CRNL	CR
FF	
NA	

NOTE: The NA option is only used by z/OS to indicate that the file format attribute should not be set.

ASCII Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: The DEFAULT_DELIM parameter specifies the default in the CA XCOM Data Transport Default Options Table/Global Parameters on the remote system.

The tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which the REMOTE_CHARSET is based, it should not be used.

Example: If REMOTE_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if REMOTE_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in REMOTE_DELIM, delimiters will be derived as explained below.

If REMOTE_DELIM is not specified or doesn't contain at least one valid delimiter, the DEFAULT_DELIM parameter in the CA XCOM Data Transport Default Options Table/Global Parameters on the remote system is considered. If DEFAULT_DELIM is also not specified or doesn't contain at least one valid delimiter, platform default values are used as shown:

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

Note: During Unicode conversion some of the delimiters may fail to convert to some code pages.

For example, In case of Send File transfer, when REMOTE_CHARSET set to ASCII based charset and REMOTE_DELIM set to NL/CRNL, then transfer may fail. The Unicode conversion fails as ASCII based code pages may not have mapping character for NL/CRNL.

remove_trail_blanks

Indicates whether or not to remove the blanks at the end of each record when receiving a text file.

Range: YES or NO

Default: NO

report_title

This field contains the report title and job number that will be printed on the report. The field has the following format:

8 characters	2 characters	8 characters
Job Name	Blanks	Job Number

The Job is optional and can be skipped. The Job Name can also be skipped, but if you skip the Job name and want to use the job number, you must pad the number with 10 blanks. For more information, see the appendix "Parameters."

Note: For report transfers only.

Range: 0 to 21 alphanumeric or blank characters

Default: None

restart_supported

Specifies whether automatic restart is to be supported on a transfer.

Range: YES or NO

Default: YES

Note: This is a Version 2 parameter.

retpd

Specifies the number of days (1 to 9999) that the tape data set being created is to be retained.

Range: 1 to 9999

Default: None

Note: RETPD and EXPDT are mutually exclusive; specify one or the other.

secondary_alloc

This parameter Identifies secondary storage allocation for a data set that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

secure_socket

Specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and must connect to a SSL listener on the remote partner.

NO

Performs a non secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

spool_flag

Indicates whether the report is to be spooled to disk or printed immediately. For report transfers only.

Note: If the remote system is an IBM mainframe, this parameter has no effect on the transfer.

Range: YES or NO

Default: YES

start_date

Indicates the date on which the transfer becomes eligible for execution. The format is *mm/dd/yy*. If this field is blank, the current date is used.

Format: *mm/dd/yy*

Default: None

start_time

Indicates the time when the transfer becomes eligible for execution. The military format of hh:mm:ss is used. If this field is blank, then the current time is used.

Format: *hh*:*mm*:*ss*

Default: None

stat_frequency

Indicates the frequency with which transfer statistics are made available to xcomqm. Intended for tuning high-speed links. Longer values help performance, but byte/record counts in xcomqm -D may be slightly behind the actual counts.

Range: 1 to 9999 records

Default: 10

storclas

Specifies the name of the storage class for a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

tape

Indicates to the remote system whether the volume is a tape volume or a disk file. Yes indicates a tape volume and that mounts are allowed when performing dynamic allocation. No indicates that the transfer is to a disk file.

Range: YES or NO

Default: NO

tempdir

Indicates the directory in which temporary files for jobs and reports can be created.

Range: 1 to 256 characters

Default: /tmp

trnencrl_cipher

This parameter specifies the requested list of ciphers which are used to encrypt the password fields for locally initiated transfers.

The cipher list consists of one or more ciphers from the table that colons separate. The characters (!) or (-) can precede each cipher in the cipher list.

If (!) is used then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-) is used then the ciphers are deleted from the list, but some or all of the ciphers are added again using later options.

Default: COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining
DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback

3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining
RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.

COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

truncation

Indicates whether CA XCOM Data Transport truncates excess characters in the source file if the record exceeds the maximum record length as indicated by the MAXRECLEN parameter. If NO is selected, and the maximum record length is exceeded, CA XCOM Data Transport aborts the transfer. This parameter is ignored if CARRIAGE_FLAG=NO.

Note: Truncation is not valid for BINARY data or for non-text data received on the UNIX or Linux platform.

Range: YES or NO

Default: NO

trusted

Allows the user to request a trusted transfer and the partner's CA XCOM Data Transport TRUSTED database to be searched to verify the user's credentials. This eliminates the need for the user to specify a USERID and PASSWORD. If XCOM_TRUSTED_OVR is set to NO or no USERID is specified, the USERID of the process that initiated the transfer will be used.

Note: TRUSTED=YES cannot be specified with indirect transfers, because this is not supported.

Range: YES, NO, Y, N

Default: NO

unit

Specifies the unit on which a dataset is to be created on an IBM mainframe.

Note: Used when FILE_OPTION=CREATE.

Range: Zero to six characters

Default: None

unitct

Specifies the number of units to be allocated on the remote system.

Range: 1 to 20

Default: None

userid

The user ID which the security system on a remote system checks before granting access for the file transfer.

Range: 0 to 12 characters

Default: None

version

Indicates the version of the CA XCOM Data Transport protocol to be used for this transfer. For using TCP/IP, only a value of 2 is valid.

Range: 1 or 2

Default: 2

Note: This is a Version 2 parameter.

volct

Specifies the maximum number of volumes to be used in processing a multi-volume output tape data set on the remote system.

Range: 1 to 255

Default: None

volsq

Specifies the sequence number of the first volume of a multivolume remote data set to be used.

Range: 1 to 255

Default: None

volume

Specifies the volume on which a dataset is to be created on an IBM mainframe.

Range: Zero to six characters

Default: None

xcomfullssl

This parameter specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and must connect to an SSL listener on the remote partner.

NO

Performs a non-secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

xcom_show_cipher

Specifies whether to display encryption algorithms in the CA XCOM Data Transport queue detailed information, which is used for transfers.

NO

Do not display encryption algorithms in the queue detail information.

YES

Display encryption algorithms in the queue detail information.

Default: NO

xidest

Specifies the name of the LU or TCP/IP address on the intermediate destination that is designated for store-and-forward transfers. If this variable is null or unset, then a direct connection to a remote system is attempted. If it contains a value, it is taken to be the name of an intermediate CA XCOM Data Transport that will handle traffic to and from the named remote system.

Note: For store-and-forward transfers only.

Range: 0 to 14 characters

Default: None

xlogfile

The name of the file where CA XCOM Data Transport logs activity. If you do not specify this parameter, the systemwide log file $XCOM_HOME/xcom.log is used. If you specify this parameter with a different file name, the logging information is only sent to the specified file.

Note: If QUEUE=YES, specify the full path name.

Range: 0 to 256 characters

Default: $XCOM_HOME/xcom.log

Note: $XCOM_HOME is an environment variable.

xluname

Indicates the name of the local LU. This name must match the remote LU name on the remote system.

Range: 1 to 17 characters

Default: None

Note: Not used in CA XCOM Data Transport for RS/6000 SNA Services or RS/6000 SNA Server.

xmode

Indicates the mode name associated with the SNA/APPC configuration for the XLUNAME-Remote LU name pair. This name must match the mode name defined on the remote system.

Range: One to eight characters

Default: XCOMMODE

Note: Not used in CA XCOM Data Transport for RS/6000 SNA Services or RS/6000 SNA Server.

xnodespec

Required for Brixton and SunLink APPCs. Indicates the name of the node that specifies the gateway to use for the transfer. Does not apply to other APPCs.

Range: 1 to 64 characters

Default: None

xtcerrdecr

This parameter specifies the transfer requests for which the xtchold_count parameter value is decremented when the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if a transfer completes unsuccessfully.

Note:

- VTAM errors are not considered failed file transfers, because they are retried later.
- If the current file transfer fails, the value of the xtchold_count parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the xtchold_count value has already reached zero.

xtcerrincr

Specifies the transfer requests for which the xtchold_count parameter value is incremented if the current file transfer fails.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if the current file transfer fails.

Notes:

- A VTAM communications error does not count as a failure.
- If the current file transfer fails, the value of the xtchold_count parameter of the transfers assigned to this parameter is increased by one, unless the next transfer has already started or the xtchold_count value has already reached zero.

xtcerrpurge

This parameter specifies the transfer requests to be purged if the transfer concludes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be purged if the current file transfer concludes unsuccessfully.

Note: xtcerrpurge overrides xtchold_count.

xtcerrrel

This parameter specifies the transfers to be released if the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be released if the current transfer completes unsuccessfully.

Note: xtchold_count overrides xtcerrrel.

xtcgooddecr

This parameter indicates an xtcnet job whose hold count decrements if the file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if the current transfer completes successfully.

Note: If the current file transfer completes successfully, the value of the xtchold_count parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the xtchold_count value has already reached zero.

xtcgoodincr

This parameter indicates the transfer requests whose xtchold_count parameter is incremented when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if a transfer completes successfully.

Note: If the current file transfer completes successfully, the value of the xtchold_count parameter of the transfers assigned to this parameter is increased by one, unless the next transfer has already started or the xtchold_count value has already reached zero.

xtcgoodpurge

This parameter specifies the transfer requests to be purged when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be purged if a transfer completes successfully.

Note: xtcgoodpurge overrides xtchold_count.

xtcgoodrel

This parameter specifies the transfer requests to be released if the current file transfer concludes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be released if a transfer completes successfully.

Note: xtchold_count overrides xtcgoodrel.

xtchold_count

This parameter associates a number with a transfer request that is incremented or decremented by the successful or unsuccessful completion of other transfer requests. As long as the number is greater than 0, the transfer is not released.

0 to 255

Specifies a value that controls the holding/releasing of a transfer request. The transfer is released when the value of the parameter reaches 0.

Note:

- See the description of the parameters xtcerrdecr, xtcerrincr, xtcgooddecr, and xtcgoodincr, which can decrement and increment the value of the xtchold_count parameter.
- hold_transfer=YES overrides xtchold_count.

xtcjob

This parameter defines the name of a transfer request belonging to the group of interdependent transfer requests named through the xtcnet parameter.

xxxxxxxx

Specifies the name of a transfer request in a group of interrelated transfer requests.

xtcnet

This parameter defines the name of a group of interdependent transfer requests.

xxxxxxxx

Specifies up to eight alphanumeric characters representing the name of the XTC network running this transfer.

xtrace

Indicates the level of desired execution tracing.

0

No tracing

1 to 8

Levels of tracing from 1 (lowest) to 8 (highest)

9

Show the contents of the data buffers

10

Includes levels 1 to 9 plus some additional detailed technical information

Default: 0

Upgrade from Previous Releases

The XcomAPI and XcomQAPI distributed with previous releases of CA XCOM Data Transport have changed for this release. If you have versions of XcomAPI and XcomQAPI that were distributed before this release, recompile your CA XCOM Data Transport applications with the current version.

The XcomAPI distributed with previous versions of CA XCOM Data Transport is different from this API. If you have versions of the API that were distributed prior to this version, recompile your CA XCOM Data Transport applications with the current version.

The following changes have been made to the API:

- CA XCOM Data Transport now uses the following APIs:
  - XcomAPI
  - XcomQAPI
  - msgstring
  - XsetSystemPriv
- In previous releases, API users used CA XCOM Data Transport static/archive libraries (libxcom.a, libxcomtcp.a, libxcomtxpi.a, and libxcomcomp.a) to compile and link custom applications. Now, you must use a shared library (libxcomshared.so) to compile and link your custom applications. Therefore, API users can use libxcomshared.so to perform CA XCOM Data Transport transfers.

Changes to Access Permissions

Compile and link your programs, then perform the commands shown below, replacing *your_program* with the name of your program, as follows:

```
chown root your_program
chgrp xcomadm your_program
chmod 6755 your_program
```

These commands set the appropriate access permissions for using CA XCOM Data Transport with your program.

Link Libraries

When linking libraries, you must use the appropriate CA XCOM Data Transport library for non-queued transfers that use SNA/APPC or TCP/IP protocols. For queued XcomAPI transfers and XcomQAPI requests that are not combined with an XcomAPI transfer, either set of files is satisfactory.

Sample links for use with both SNA and TCP/IP transfers are provided in the makefile and qapimake files distributed with CA XCOM Data Transport. For the locations of the library files and the API source files, see the *Installation Guide* for your platform.

The libxcomshared.so library is required for performing non-queued SNA or TCP/IP transfers.

Starting States

Valid starting states are as follows:

LOCAL_SEND

Initiates the transmission of a local file to a remote system as a report, job, or file transfer.

LOCAL_RECEIVE

Initiates the transmission of a remote file to the local system.

Return Values

The value returned by XcomAPI is zero if the transfer was successful. If not, an error code as defined in the startst.h header file is generated.

Starting States in startst.h

The startst.h file contains the following:

- /* starting states to pass to XcomAPI */
- #define LOCAL_SEND 152
- #define LOCAL_RECEIVE 153

API Control Block Structure

The following is the include file xcomapi.h. This is a C language header file containing the necessary API definitions for the control block structure:

```
#ifndef __INCL_XCOMAPI_H
#define __INCL_XCOMAPI_H
#include "startst.h"
#define SEND_FILE "1"
#define SEND_REPORT "2"
#define SEND_JOB "3"
#define RECEIVE_FILE "4"
typedef struct {
char
*file_type, /* 1-file, 2-report, 3-job, 4-retreive ( the same as
starting_state=LOCAL_RECEIVE )*/
*protocol,            /* communications protocol: SNA or TCPIP */
*port,                /* TCPIP remote server port */
*remote_system,       /* remote lu name */
*xluname,             /* local lu name */
*xnodespec,           /* node specification for SNA gateway node */
*xmode,               /* mode */
*xidest,              /* intermediate destination */
*local_file,          /* local filename for send */
*remote_file,         /* local filename for receive */
*local_file_rf,       /* local filename for receive */
*remote_file_rf,      /* remote filename for receive */
*file_option,         /* CREATE / REPLACE / APPEND */
*remove_trail_blanks, /* remove trailing blanks (MVS/R) */
*volume,              /* volume */
*unit,                /* unit */
*record_format,       /* record format */
*lrecl,               /* logical record length */
*blksize,             /* block size */

/* remote printing parameters */
*local_file_sr,              /* local filename for sending reports */
*class,                      /* printjob class */
*destination,                /* name of remote printer */
*form,                       /* name of special forms to be used */
*fcb,                        /* Form Control Block */
*copies,                     /* number of copies to be printed */
*report_title,              /* report name (banner) to be used on separator sheet
*/
*hold,                       /* flag for MVS spoolers */
*spool_flag,                 /* turns on spooling on receiving side */
*disposition,                /* MVS file disposition */
*carriage_control_characters, /* A - ASA chars in column 1 M -IBM Machine chars (MVS)
*/
```

```
*local_file_sj,              /* local filename for sending jobs */
*userid,                     /* user id */
*password,                   /* password */
*queue,                     /* YES - queue transactions NO - execute immediately
*/
*start_time,                 /* time to start scheduled transaction */
*start_date,                 /* date to start scheduled transaction */
*notifyr,                    /* TSO / WTO / CICS / LU / VM */
*notify_name,                /* who to notify */
*local_notify,               /* user name for notification */
*notifyl,                    /* MAIL / WRITE */
*notify_term,                /* user's terminal to notify if notifyl=WRITE */
*maxreclen,                  /* maximum record length */
*code_flag,                  /* ASCII / BINARY / EBCDIC */
*carriage_flag,             /* specifies if it is a text file or a type of record
packing to use */
*truncation,                 /* YES - truncation is allowed for this transfer */
*compress,                   /* YES / NO */
*xtrace,                     /* trace level */
*stat_frequency,             /* How often to put record stats to Q-record */
*debug_flag,                /* YES - debug output written to /tmp/<tid>.it file */
*xlogfile,                   /* log for locally initiated transactions */
*tempdir,                    /* directory to hold temporary files */

/* parameters used by xcom62 version 2 only */
*version,
*number_of_retries,
*allocation_type,
*checkpoint_count,
*num_of_dir_blocks,
*primary_alloc,
*secondary_alloc,
*restart_supported,

/* parameters used by XCOM 3.1 only */
*trusted,
*domain,
*den,
*expdt,
*retpd,
*label,
*tape,
*unitct,
*volct,
*volsq,
*labelnum
*tapedisp,
/* parameters used by xcom r11 only */
*xcomfullssl
```

```
*configssl,
*rmtntfyl,
*lclntfyl,
*storcls,
*datacls,
*mgtclas,
*dsntype,
/* Reserved for future use */
*seclabel,
*hfs_flag,

/* parameters normally taken from shared memory but may be overwritten in API */
*eol_classes,        /* print classes having NL added at end of record */
*convert_classes,    /* classes needing ebcdic-ascii translation */
*metacode_classes,   /* classes needing variable length records */
*ppVersion;          /* Enable xcompp.bat for all transfers? */
*lear_cipher,        /* local cipher */
*lear_hash,          /*  local hash */
*ear_hash,           /* remote hash */
*ear_cipher,         /* remote cipher */
*lear_digest,        /* local digest */
*ear_digest,         /* remote digest */
*lear_key,           /* local key */
*ear_key,            /* remote key */

/* parameters used by CA XCOM Data Transport r11.6 only */
*trnencrl_cipher,   /* requested list of ciphers for password encryption */
*local_charset,
*remote_charset,
*mbcs_inputerror,
*mbcs_converror,
*local_delim,
*remote_delim,
*avgrec,
*eattr;
} XCOM_PARM;

typedef struct {
    char user_name[128];          /* user name */
    char array_message[128];      /* return text from multi-entries */
    char array_message2[128];     /* return text from multi-entries */
    char array_err_message[128];  /* return text from multi-entries */
} XCOM_QUEUE_ENTRY;

typedef XCOM_QUEUE_ENTRY XCOM_QUEUE[9999];

typedef struct {
    char funccode[2];          /* function code */
    char tid[7];               /* transfer id (6 characters + null) */
```

```
    char global_tid[128];      /* local transfer id */
    char remote_tid[128];      /* remote transfer id */
    int  queue_open_once;      /* allow queue open once */
    int  nMax_Queue_Entries;   /* Maximum number of queue_entry */
    int  nQueueEntries;        /* number of queue_entry returned */
    int  trace_level;          /* trace level */
    FILE *trace_stream;        /* file handle for trace file */
    char trace_filename[256];  /* name of the trace file */
    char start_time[128];      /* start time */
    char end_time[128];        /* end time */
    char user_name[128];       /* remote user name */
    char group_name[128];      /* group name */
    char transfer_name[128];   /* user definable id for transfer request */
    char xferdata[128];        /* user data -- transfer specific */
    char sysdata[128];         /* user data -- system dependent */
    char in_out[128];          /* in_out value */
    char pid[128];             /* process id */
    char remote_system[128];   /* remote lu name */
    char file_type[128];       /* file type */
    char transfer_type[128];   /* transfer type */
    char user_group[128];      /* owned by user from group */
    char lfile[257];           /* local file */
    char file[257];            /* temporary file */
    char rfile[257];           /* remote file */
    char file_option[128];     /* file creation option */
    char curr_cond[128];       /* current condition */
    char curr_status[128];     /* current status */
    char curr_state[128];      /* current TP state */
    char rec_read_write[128];  /* record read or written to file */
    char byte_read_write[128]; /* bytes read or written to file */
    char blocks_trans[128];    /* blocks transmitted */
    char bytes_trans[128];     /* bytes transmitted */
    char sdsmessage[160];      /* message test for SDSNOTE */
    char head_message[128];    /* title message */
    char queue_message[128];   /* return text from queue */
    char start_message[128];   /* return text from start of xcomd service*/
    char start_message2[128];  /* return text from start of xcomd service*/
    char end_message[128];     /* return text from end of process */
    char neg_cipher[128];      /* Password Encryption cipher */
        XCOM_QUEUE *pQueue;    /* An array of XCOM_QUEUE structure */

} XCOM_QUEUE_PARM;

#ifdef _NO_PROTO
int XcomAPI();
#else
int XcomAPI(int invocation_mode,XCOM_PARM xparmblock);
#endif
```

```
#ifdef _NO_PROTO
int XcomQAPI();
#else
int XcomQAPI(XCOM_QUEUE_PARM * xqueueblock);
#endif

#ifdef _NO_PROTO
char* msgstring();
#else
char* msgstring(int state, int sense);
#endif


#ifdef _NO_PROTO
int XsetSystemPriv( );
#else
int XsetSystemPriv( int type );
#endif

#endif /* __INCL_XCOMAPI_H */
```

Using XcomQAPI Option Parameters

The following XcomQAPI option parameters are for use by applications programmers to pass and retrieve information in a C structure to and from the XcomQAPI routine. These parameters need to be supplied by you and passed on to the XcomQAPI. The rest of parameters are a received buffer. They will receive data from XcomQAPI.

Option Parameters

The option parameters in the following list are used to pass and retrieve information from the queue request.

funccode

Used to define a function parameter to pass to XcomQAPI().

r

Display the release level of xcomqm and exit.

A*entryname*

Release a held transfer.

C*entryname*

Get a trace of a transfer.

D*entryname*

Display details about a queue entry.

H*entryname*

Hold a scheduled transfer.

L[ ]

List your queued transfers.

La

List all queued transfers.

R*entryname*

Remove a queue entry.

Rf*entryname*

Force the entry to be removed from the queue. Use this command when you want to remove a transfer that is still active.

R*

Remove all completed or scheduled entries from the queue. If you use this option on an active transfer, it interrupts the transfer.

T*entryname*

Terminate an active transfer.

Tf*entryname*

Force termination of the transfer. Use this command when you want to remove a transfer that is still active.

Important! When using SNA/APPC protocols, the TP will ABEND and it may be necessary to restart the underlying SNA software. When using TCP/IP, xcomtcp will ABEND.

S*entryname*

Suspend a transfer.

E*entryname*

Resume a suspended transfer.

d

Create a trace file.

g*entryname*

Display queue entry information.

p*entryname*

Used to change the value of: FILE_OPTION LOCAL_FILE REMOTE_FILE.

Note: This option can be called only when xcbp->state is in PREALLOCATION_STATE.

Note: The *entryname* variable represents the Transaction ID, which is the six-digit name of the transfer entry in the queue.

nMax_Queue_Entries

Defines maximum queue entries. The value should be set to equal MAX_QUEUE_ENTRIES in the xcom.glb file.

queue_open_once

Open queue flag. Define this only once at first call of XcomQAPI. The value should be set to 0.

tid

A queue entry TID number.

trace_level

Defines the trace level. The value can be set to 0, 1, or 9.

trace_filename

It defines a trace file name and its location; for example:

$XCOM_HOME/trace/xcomqapi.tra.

Note: $XCOM_HOME is an environment variable.

API Examples

The following API is an example of a file transfer. It uses the XcomAPI routine to pass information to CA XCOM Data Transport. This sample API program is the apitest.c file distributed with CA XCOM Data Transport. (For the location of this file, see the *CA XCOM Data Transport Installation Guide* for your platform.)

Note: The API example provided on your distribution media may have been updated for your CA XCOM Data Transport system and may be different from the example shown below.

C Language API File Transfer to Remote System File Example

The following is a C language example of an API that transfers the file named /usr/include/xcom/xcomapi.h to the file /home/user/newfile.name on the remote system.

```
/* apitest.c - This is a sample program for calling the XCOM API.*/

/*             Modify this according to your needs and make it   */
/*             with the sample 'makefile' in this directory.     */

#include <stdio.h>

/* ALL UNIX flavors except HP-UX */
#include <xcom/xcomapi.h>
/* HP-UX */
/* #include "/opt/xcom/include/xcomapi.h" */

main(int argc,char **argv)
{
    int rc;
    XCOM_PARM xcomparm;
    int starting_state;

    xcomparm.file_type = SEND_FILE;            /* "1" */

    /* remember to link with the appropriate xcom libraries for
       each protocol */
    xcomparm.protocol = "TCPIP";              /* "SNA" or "TCPIP" */

    xcomparm.remote_system = "remote_host";
    xcomparm.port = "8044";
    xcomparm.xluname = "LOCUNIX";
    xcomparm.xnodespec = NULL;

    xcomparm.xmode = "XCOMMODE";

    xcomparm.xidest = NULL;

    xcomparm.local_file = "/usr/include/xcom/xcomapi.h";
    xcomparm.remote_file = "/home/user/newfile.name";

    xcomparm.local_file_rf = "/tmp/receive.filename";
    xcomparm.remote_file_rf = "/home/user/file.there";

    xcomparm.file_option = "REPLACE";
    xcomparm.remove_trail_blanks = NULL;
    xcomparm.volume = NULL;
    xcomparm.unit = NULL;
    xcomparm.record_format = NULL;
```

```
xcomparm.lrecl = NULL;
xcomparm.blksize = NULL;

xcomparm.local_file_sr = NULL;
xcomparm.class = NULL;
xcomparm.destination = NULL;
xcomparm.form = NULL;
xcomparm.fcb = NULL;
xcomparm.copies = NULL;
xcomparm.report_title = NULL;
xcomparm.hold = NULL;
xcomparm.spool_flag = NULL;
xcomparm.disposition = NULL;
xcomparm.carriage_control_characters = NULL;

xcomparm.local_file_sj = NULL;

xcomparm.userid = "uuuuuuuu";
xcomparm.password = "xxxxxxxx";

/* if queue = NO, must link with right xcom lib */
xcomparm.queue = "NO";

xcomparm.start_time = NULL;
xcomparm.start_date = NULL;

xcomparm.notifyr = NULL;
xcomparm.notify_name = NULL;
xcomparm.local_notify = NULL;
xcomparm.notifyl = NULL;
xcomparm.notify_term = NULL;

xcomparm.maxreclen = NULL;
xcomparm.code_flag = "ASCII";
xcomparm.carriage_flag = NULL;
xcomparm.truncation = NULL;
xcomparm.compress = NULL;

xcomparm.xtrace = "10";
xcomparm.stat_frequency = NULL;
xcomparm.debug_flag = NULL;

xcomparm.xlogfile = NULL;
xcomparm.tempdir = NULL;

xcomparm.eol_classes = NULL;
xcomparm.convert_classes = NULL;
xcomparm.metacode_classes = NULL;
```

```
/* These are version two parms */
        xcomparm.version = NULL;
        xcomparm.number_of_retries = NULL;
        xcomparm.allocation_type = NULL;
        xcomparm.checkpoint_count = NULL;
        xcomparm.num_of_dir_blocks = NULL;
        xcomparm.primary_alloc = NULL;
        xcomparm.secondary_alloc = NULL;
        xcomparm.restart_supported = NULL;

/* These parms are new for V3.1 XCOM */
        xcomparm.trusted = NULL;
        xcomparm.domain = NULL;
        xcomparm.den = NULL;
        xcomparm.expdt = NULL;
        xcomparm.retpd = NULL;
        xcomparm.label = NULL;
        xcomparm.tape = NULL;
        xcomparm.unitct = NULL;
        xcomparm.volct = NULL;
        xcomparm.volsq = NULL;
        xcomparm.labelnum = NULL;
        xcomparm.tapedisp = NULL;

/* These parms are new for r11 XCOM */
        xcomparm.xcomsslfull = NULL;
        xcomparm.configssl = NULL;
        xcomparm.codetabl = NULL;
        xcomparm.rmtntfyl = NULL;
        xcomparm.lclntfyl = NULL;
        xcomparm.storcls = NULL;
        xcomparm.datacls = NULL;
        xcomparm.mgtclas = NULL;
        xcomparm.dsntype = NULL;
    /* Reserved for future use */
        xcomparm.seclabel  =NULL;
        xcomparm.hfs_flag =NULL;

        xcomparm.seclabel = NULL;
        xcomparm.hold_transfer = NULL;
        xcomparm.xtchold_count = NULL;
        xcomparm.xtcerrdecr = NULL;
        xcomparm.xtcerrincr = NULL;
        xcomparm.xtcerrpurge = NULL;
        xcomparm.xtcerrrel = NULL;
        xcomparm.xtcgooddecr = NULL;
        xcomparm.xtcgoodincr = NULL;
        xcomparm.xtcgoodpurge = NULL;
        xcomparm.xtcgoodrel = NULL;
```

```
        xcomparm.xtcnet = NULL;
        xcomparm.xtcjob = NULL;
        xcomparm.lear_cipher = NULL;
        xcomparm.ear_cipher = NULL;
        xcomparm.lear_hash = NULL;
        xcomparm.ear_hash = NULL;
        xcomparm.lear_digest = NULL;
        xcomparm.ear_digest = NULL;
        xcomparm.lear_key = NULL;
        xcomparm.ear_key = NULL;
        xcomparm.trnencrl_cipher = NULL;
        xcomparm.createdelete=NULL;
        xcomparm.compress_pds=NULL;
        xcomparm.local_charset=NULL;
        xcomparm.remote_charset=NULL;
        xcomparm.mbcs_inputerror=NULL;
        xcomparm.mbcs_converror=NULL;
        xcomparm.local_delim=NULL;
        xcomparm.remote_delim=NULL;
        xcomparm.eattr=NULL;
        xcomparm.avgrec=NULL;
        xcomparm.gatwayguid=NULL;
        xcomparm.lgatewayguid=NULL;
        xcomparm.gatewayprotocol=NULL;
        starting_state=LOCAL_SEND;
        printf("==== API started ===\n");
        rc=XcomAPI(starting_state, xcomparm);
        printf("==== API ended ===\n");
        if (!rc)
           printf("==== Transfer successful\n");
        else
           printf("==== %s\n", msgstring(rc,0));
/* Reset System Privileges for second transfer */
        if (XsetSystemPriv(2) !=0)
               {
             printf("====XsetSystemPriv() faild\n");
               }

}
```

The following XcomAPI variables are reserved for future use:

```
xcomparm.seclabel
xcomparm.hfs_flag
```

Queue Request Using XcomQAPI Routine

The API below is an example of a queue request. It uses the XcomQAPI routine to pass information to CA XCOM Data Transport. This sample API program is the qapitest.c file distributed with CA XCOM Data Transport. (For the location of this file, see the *CA XCOM Data Transport Installation Guide* for your platform.)

Note: The API example provided on your distribution media may have been updated for your CA XCOM Data Transport system and may be different from the example shown below.

Example

The following is a C example of an API calling the XcomQAPI.

```
#include <stdio.h>

/* ALL UNIX and Linux flavors except HP-UX */
#include <xcom/xcomapi.h>
/* HP-UX */
/* #include "/opt/xcom/include/xcomapi.h" */

main(int argc,char **argv)
{
    int rc, i;
    XCOM_QUEUE_PARM *xcomqparm;

    /* allocate memory for structure of xcomqparm */
    xcomqparm = (XCOM_QUEUE_PARM *)malloc(sizeof(XCOM_QUEUE_PARM));
    if (xcomqparm == NULL)
       printf("Could not allocate memory to run QAPI process !\n");
    else {
          xcomqparm->queue_open_once = 0;
          /* xcomqparm->nMax_Queue_Entries should equal to MAX_QUEUE_ENTRIES in
xcom.glb file */
          xcomqparm->nMax_Queue_Entries = 500;
          /* allocate memory for structure of pQueue */
          xcomqparm->pQueue = (XCOM_QUEUE *)malloc(xcomqparm->nMax_Queue_Entries *
sizeof(XCOM_QUEUE_ENTRY));
          if (xcomqparm->pQueue == NULL)
             printf("Could not allocate memory to run QAPI process !\n");
          else {
             printf("==== QAPI started ===\n");

             /* call option 'r' to display CA-XCOM version */
             xcomqparm->funccode[0] = 'r';
             rc=XcomQAPI(xcomqparm);
             if (strlen(xcomqparm->start_message))
                printf("%s", xcomqparm->start_message);
```

```
                if (xcomqparm->nQueueEntries == 1)
                   printf("%s", (*xcomqparm->pQueue)[0].array_message);

                /* call option 'd' to turn on trace file */
                xcomqparm->trace_level = 9;
                strcpy(xcomqparm->trace_filename,
"/usr/spool/xcom/trace/xcomqapi.tra");
                xcomqparm->funccode[0] = 'd';
                rc=XcomQAPI(xcomqparm);

                /* call option 'La' to list all queue entries */
                xcomqparm->funccode[0] = 'L';
                xcomqparm->funccode[1] = 'a';
                rc=XcomQAPI(xcomqparm);
                /* check for error message */
                if (strlen((*xcomqparm->pQueue)[0].array_err_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_err_message);
                /* print detail information for all queue entries */
                for (i = 0; i < xcomqparm->nQueueEntries; i++) {
                    printf("%s", (*xcomqparm->pQueue)[i].user_name);
                    printf("%s", (*xcomqparm->pQueue)[i].array_message);
                }
                if ((xcomqparm->nQueueEntries == 0) &&
strlen((*xcomqparm->pQueue)[0].array_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_message);

                /* call option 'Dxxxxxx' to display detail record information */
                memset(xcomqparm->lfile,0,sizeof(xcomqparm->lfile));
                memset(xcomqparm->rfile,0,sizeof(xcomqparm->rfile));
                strcpy(xcomqparm->tid, "000001");
                xcomqparm->funccode[0] = 'D';
                rc=XcomQAPI(xcomqparm);
                /* check for error message */
                if (strlen((*xcomqparm->pQueue)[0].array_err_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_err_message);
                /* display detail record information */
                if (strlen(xcomqparm->head_message))
                   printf("%s", xcomqparm->head_message);
                if (strlen(xcomqparm->global_tid))
                   printf("%s", xcomqparm->global_tid);
                if (strlen(xcomqparm->remote_tid))
                   printf("%s", xcomqparm->remote_tid);
                if (strlen(xcomqparm->group_name))
                   printf("%s", xcomqparm->group_name);
                if (strlen(xcomqparm->transfer_name))
                   printf("%s", xcomqparm->transfer_name);
                if (strlen(xcomqparm->xferdata))
                   printf("%s", xcomqparm->xferdata);
                if (strlen(xcomqparm->sysdata))
```

```
   printf("%s", xcomqparm->sysdata);
if (strlen(xcomqparm->start_time))
   printf("%s", xcomqparm->start_time);
if (strlen(xcomqparm->transfer_type))
   printf("%s", xcomqparm->transfer_type);
if (strlen(xcomqparm->lfile))
   printf("%s", xcomqparm->lfile);
if (strlen(xcomqparm->file))
   printf("%s", xcomqparm->file);
if (strlen(xcomqparm->sdsmessage))
   printf("%s", xcomqparm->sdsmessage);
if (strlen(xcomqparm->rfile))
   printf("%s", xcomqparm->rfile);
if (strlen(xcomqparm->in_out))
   printf("%s", xcomqparm->in_out);
if (strlen(xcomqparm->remote_system))
   printf("%s", xcomqparm->remote_system);
if (strlen(xcomqparm->end_time))
   printf("%s", xcomqparm->end_time);
if (strlen(xcomqparm->pid))
   printf("%s", xcomqparm->pid);
if (strlen(xcomqparm->user_group))
   printf("%s", xcomqparm->user_group);
if (strlen(xcomqparm->curr_cond))
   printf("%s", xcomqparm->curr_cond);
if (strlen(xcomqparm->curr_status))
   printf("%s", xcomqparm->curr_status);
if (strlen(xcomqparm->curr_state))
   printf("%s", xcomqparm->curr_state);
if (strlen(xcomqparm->rec_read_write))
   printf("%s", xcomqparm->rec_read_write);
if (strlen(xcomqparm->byte_read_write))
   printf("%s", xcomqparm->byte_read_write);
if (strlen(xcomqparm->blocks_trans))
   printf("%s", xcomqparm->blocks_trans);
if (strlen(xcomqparm->bytes_trans))
   printf("%s", xcomqparm->bytes_trans);
if (strlen(xcomqparm->neg_cipher))
   printf("%s", xcomqparm->neg_cipher);
if (strlen((*xcomqparm->pQueue)[0].array_message))
   printf("%s", (*xcomqparm->pQueue)[0].array_message);
if (strlen(xcomqparm->gatewayprotocol))
   printf("%s", xcomqparm->gatewayprotocol);

/* call option 'R*' to remove queue entries */
xcomqparm->funccode[0] = 'R';
xcomqparm->funccode[1] = '*';
rc=XcomQAPI(xcomqparm);
/* check for error message */
```

```
                if (strlen((*xcomqparm->pQueue)[0].array_err_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_err_message);
                /* display detail information for removed queue entries */
                for (i = 0; i < xcomqparm->nQueueEntries; i++) {
                    printf("%s", (*xcomqparm->pQueue)[i].array_message);
                    printf("%s", (*xcomqparm->pQueue)[i].array_message2);
                }
                if ((xcomqparm->nQueueEntries == 0) &&
strlen((*xcomqparm->pQueue)[0].array_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_message);
                if (!rc)
                   printf("==== Call XcomQAPI successful !\n");
                else
                   printf("==== %s\n", msgstring(rc, 0));

                /* call option 'gxxxxxx' to display queue entry information */
                memset(xcomqparm->lfile,0,sizeof(xcomqparm->lfile));
                memset(xcomqparm->rfile,0,sizeof(xcomqparm->rfile));
                memset(xcomqparm->file_option,0,sizeof(xcomqparm->file_option));

                xcomqparm->funccode[0] = 'g';
                strcpy(xcomqparm->tid, "000001");
                rc=XcomQAPI(xcomqparm);
                /* check for error message */
                if (strlen((*xcomqparm->pQueue)[0].array_err_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_err_message);
                /* display queue entry information */
                if (strlen((*xcomqparm->pQueue)[0].array_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_message);
                if (strlen(xcomqparm->group_name))
                   printf("%s", xcomqparm->group_name);
                if (strlen(xcomqparm->lfile))
                   printf("%s", xcomqparm->lfile);
                if (strlen(xcomqparm->rfile))
                   printf("%s", xcomqparm->rfile);
                if (strlen(xcomqparm->user_name))
                   printf("%s", xcomqparm->user_name);
                if (strlen(xcomqparm->transfer_type))
                   printf("%s", xcomqparm->transfer_type);
                if (strlen(xcomqparm->file_type))
                   printf("%s", xcomqparm->file_type);
                if (strlen(xcomqparm->file_option))
                   printf("%s", xcomqparm->file_option);

                /* call option 'pxxxxxx' to change value: FILE_OPTION LOCAL_FILE
REMOTE_FILE
                   note: This option only allows to call when xcbp->state is in
PREALLOCATION_STATE */
                xcomqparm->funccode[0] = 'p';
```

```
                strcpy(xcomqparm->tid, "000001");
                strcpy(xcomqparm->file_option, "CREATE");
                strcpy(xcomqparm->lfile, "/home/user/test.txt");
                strcpy(xcomqparm->rfile, "/home/user/test.del");
                rc=XcomQAPI(xcomqparm);
                /* check for error message */
                if (strlen((*xcomqparm->pQueue)[0].array_err_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_err_message);
                /* print out queue entry information */
                if (strlen((*xcomqparm->pQueue)[0].array_message))
                   printf("%s", (*xcomqparm->pQueue)[0].array_message);

                printf("====  QAPI ended  ===\n");
            }
        }
        /* free memory */
        if (xcomqparm->pQueue != NULL)
           free(xcomqparm->pQueue);
        if (xcomqparm != NULL)
           free(xcomqparm);
}
```

XcomQAPI Return Codes

The return codes for XcomQAPI are as follows:

603

The TP state is invalid.

602

There was a general error in the queue function.

601

Permission denied.

600

Transfer is not active.

469

CA XCOM scheduler service is not running or is not compatible.

-1

The named file does not exist or is not accessible in the given mode.

-9

XCOMQM detected an unknown error.

-1100

XCOMQM is unable to find the entry in the queue.

-1801

XCOMQM is unable to open the index file.

-1802

XCOMQM is unable to close the index file.

-1803

XCOMQM detected an invalid command line option.

Chapter 5: Generating SSL Certificates

This chapter describes how to generate certificates that can be used with CA XCOM Data Transport.

For more information on using OpenSSL, see *Network Security with OpenSSL* by John Vega, Matt Messier, and Pravir Chandra (O'Reilly & Associates).

This section contains the following topics:

Using SSL Mode (see page 341)
Setting Expiration (see page 342)
Creating the CA Certificate (see page 343)
Creating the Server Certificate (see page 344)
Creating the Client Certificate (see page 344)
Configuring the CA XCOM Data Transport SSL Server (see page 345)
Configuring the CA XCOM Data Transport Client (see page 346)
Default Sample Scripts (see page 346)
Cipher Suites (see page 347)
Supported Cryptographic Protocols (see page 348)
Supported Cipher Suites (see page 348)

Using SSL Mode

CA XCOM Data Transport uses SSL in client/server mode. In client/server mode, certificates are required for both the local (initiating) and remote (receiving) CA XCOM Data Transport partners. SSL considers the local CA XCOM Data Transport partner to be the client and the remote CA XCOM Data Transport partner to be the server.

When establishing the SSL connection, the server sends the server certificate to the client for verification. After the client verifies the server certificate, the client sends the client certificate to the server for verification. Both the client and the server must verify the CA certificate from the other.

Setting up SSL for CA XCOM Data Transport involves the following tasks:

1. Set the expiration for the CA Certificate.
2. Create the CA Certificate.
3. Create the server certificate.
4. Create the client certificate.
5. Configure the CA XCOM Data Transport SSL server.
6. Configure the CA XCOM Data Transport client.

Setting Expiration

When generating a CA certificate, the default_days parameter in cassl.conf that controls the expiration of server and client certificates is not used for CA certificates. The certificate is generated with a default expiration of 30 days.

To change the default expiration

1. Add 'days nnn' to the makeca script line. The following line is an example of how the makeca script is shipped:

   ```
   Openssl req -x509 -newkey rsa -out ./certs/cassl.pem -outform PEM
   ```

2. To change the expiration to one year, change the line before running the makeca script:

   ```
   Openssl req -x509 -newkey rsa -out ./certs/cassl.pem -outform PEM -days 365
   ```

Creating the CA Certificate

To create the CA certificate

1. Create a configuration file that is used as input to the openssl utility. A sample file, named cassl.conf, was installed in the ssl subdirectory of the CA XCOM Data Transport installation directory for UNIX and Windows. For z/OS, it is downloaded as part of a .TAR formatted file, and then copied to a user-specified path on the site's HFS file system. This .TAR file needs to have the SSL files extracted before it can be edited. Change to the ssl subdirectory and edit the [root_ca_distinguished_name] section, changing the values as appropriate for your system.

   Note: For UNIX, you must have 'root' authority to perform this task.

   Issue the following command to run the makeca script:

   ```
   ./makeca
   ```

   This shell script uses the cassl.conf file to generate a certificate and key file. The certificate, cassl.pem, is saved in the 'certs' subdirectory. The key file, generated as casslkey.pem, is saved in the 'private' subdirectory.

   Note: When running the makeca script the first time, the pseudo-random number generator (PRNG) file does not exist and issues a warning to this effect. The makeca utility generates the PRNG file the first time it is run and does not issue this warning on subsequent executions. This is only a warning; you can continue with the next step.

2. To list the certificate just created, issue the following command to use the listca script:

   ```
   ./listca
   ```

   This shell script displays the CA certificate and the information stored in the package.

Creating the Server Certificate

To create the server certificate

1. Create a configuration file to use as input to the openssl utility. A sample file, serverssl.conf, was installed in the ssl subdirectory. Edit the [req_distinguished_name] section, changing the values to your specifications.

   Note: For UNIX, you must have root authority to perform this task.

2. Using the script makeserver, issue the following command:

   ```
   ./makeserver
   ```

   The makeserver shell script uses the serverssl.conf file and the cassl.pem file to generate a server certificate and a key file. The server certificate, servercert.pem, is saved in the 'certs' subdirectory. The key file, generated as serverkey.pem, is saved in the 'private' subdirectory.

   To list the certificate just created, issue the following command to use the listserver script:

   ```
   ./listserver
   ```

   This shell script displays the server certificate and information stored in the package.

Creating the Client Certificate

To create the client certificate

1. Create a configuration file to use as input to the openssl utility. A sample file, clientssl.conf, was installed in the ssl subdirectory. Edit the [req_distinguished_name] section, changing the values to meet your system requirements.

   Note: For UNIX, you must have root authority to perform this task.

2. Issue the following command to use the makeclient script:

   ```
   ./makeclient
   ```

   The makeclient shell script uses the clientssl.conf file and the cassl.pem file to generate a client certificate and a key file. The certificate, clientcert.pem, is saved in the 'certs' subdirectory. The key file, generated as clientkey.pem, is saved in the 'private' subdirectory.

3. To list the certificate just created, issue the following command to use the listclient script:

   ```
   ./listclient
   ```

   The listclient shell script displays the client certificate and information stored in the package.

Configuring the CA XCOM Data Transport SSL Server

To configure CA XCOM Data Transport to use the CA and server certificates for establishing server (remote) SSL connections

1. Review and modify the CA XCOM Data Transport SSL configuration file, configssl.cnf, so that the settings meet your site standards. Server connections use the RECEIVE_SIDE values. Also, ensure that the XCOM_HOME environment variable is set correctly to the XCOM installed location since it is used within this file.
2. Set the XCOM_CONFIG_SSL parameter in your default options table/global file to point to your customized configssl.cnf file.

   Note: For z/OS, the path and file name must be an HFS file.
3. Configure CA XCOM Data Transport to receive remote SSL connections:
   a. For z/OS, specify the TCP/IP port that will accept SSL connection requests using the SSLPORT and/or SSLPORTV6 default options table parameters. In addition, the default options table parameter, SSL, must also be set to one of the following values:
      - ONLY—To allow incoming SSL transfers only
      - ALLOW—To allow both incoming SSL and incoming non-SSL transfers to this server

   b. For UNIX, during installation, the txpis and txpis6 (where applicable) services along with the default TCP/IP port values that will accept SSL connection requests are automatically added to the inetd configuration files. If different TCP/IP port values are needed from the default values then the txpis and/or txpis6 entries in the /etc/services file will have to be manually changed.

   c. For Windows, if the default TCP/IP port values that accept SSL connection requests need to be changed then they can be modified by using the SSL Port Number and/or the Ipv6 Port Number fields on the TCP/IP tab in the Global Parameters GUI. In addition, the Choose Listeners field may need to be updated from the default of IPv4 Listeners depending on what listeners the site needs to have started.
4. Verify that the port that receives incoming SSL connections is a unique port that is not in use by any other application. The port used for incoming TCP/IP connections cannot also be used for incoming SSL connections. If CA XCOM Data Transport will be receiving both incoming TCP/IP connections and incoming SSL connections, then two ports are required.
   a. For z/OS, reassemble the default options table and restart the CA XCOM Data Transport server (started task).

   b. For UNIX and Windows, restart the CA XCOM Data Transport service.

Configuring the CA XCOM Data Transport Client

To configure the CA XCOM Data Transport client to use the CA certificate and the server certificate when establishing client (local) SSL connections

1. Review and modify the settings of the CA XCOM Data Transport SSL configuration file, configssl.cnf, as appropriate for your system. Client connections use the INITIATE_SIDE values. Also, ensure that the XCOM_HOME environment variable is set correctly to the XCOM installed location since it is used within this file.
2. Point the XCOM_CONFIG_SSL parameter in your default options table/global file to your customized configssl.cnf file.

   Note: For z/OS, the path and file name must be an HFS file.

   a. For z/OS, the XCOM_CONFIG_SSL parameter can also be specified as a destination member parameter.

   b. For UNIX and Windows, the XCOM_CONFIG_SSL parameter can also be specified in your configuration (cnf or xml) file.
3. Set the SECURE_SOCKET parameter to YES to indicate an SSL connection.

   a. For z/OS, specify the SECURE_SOCKET parameter in the SYSIN01, the destination member, or the default options table.

   b. For UNIX and Windows, specify the SECURE_SOCKET parameter in the configuration (cnf) file.
4. Specify the port through which the remote CA XCOM Data Transport partner accepts SSL connections. Use one of the following parameters:

   a. PORT for UNIX and Windows

   b. IPPORT for z/OS
5. Initiate the transfer request.

Default Sample Scripts

By default, the CA XCOM Data Transport setup deploys sample scripts to create RSA certificates in your $XCOM_HOME\ssl directory. The type of certificate depends on the cipher suite used in your SSL communication. You can control this setting by using the $XCOM_HOME\config\configssl.cnf file.

Your security administrator needs to set the appropriate cipher suite for SSL communication.

Cipher Suites

When a TLS or SSL connection is established, the client and server negotiate a cipher suite, exchanging cipher suite codes in the client hello and server hello messages. The cipher suite specifies a combination of cryptographic algorithms to be used for the connection.

By default, a strong cipher suite is set in configssl.cnf as follows:

```
[SSL_METHOD]
INITIATE_SIDE = v3
RECEIVE_SIDE  = v3
# Optional
[CIPHER]
INITIATE_SIDE = ALL:!ADH:!LOW:!EXP:MD5:@STRENGTH
RECEIVE_SIDE  = ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH
```

This cipher suite uses the v3 protocol. By default, the following are excluded from the negotiation:

- Anonymous DH cipher suites
- Low encryption cipher suites (currently those using 64-bit or 56-bit encryption algorithms)
- Cipher suites using MD5

Ciphers are sorted according to their strength.

Supported Cryptographic Protocols

CA XCOM Data Transport supports the following protocols:

- TLSV1
- V3

In configssl.cnf, you can set the protocol used in SSL communications in one of the following ways:

For V3

```
# Mandatory, note that CA-XCOM uses the v3 protocol
[SSL_METHOD]
INITIATE_SIDE = v3
RECEIVE_SIDE  = v3
```

For TLSV1

```
# Mandatory, note that CA-XCOM uses the v3 protocol
[SSL_METHOD]
INITIATE_SIDE = tlsv1
RECEIVE_SIDE  = tlsv1
```

Note: The V3 method is applicable only if FIPS_MODE=OFF.

Supported Cipher Suites

This section lists the cipher suites that CA XCOM Data Transport supports when using the TLSV1 or V3 protocols and when the global parameter FIPS_MODE is set to YES or NO.

In the configssl.cnf file, you can set the Cipher Suite name as follows:

```
# Optional
[CIPHER]
INITIATE_SIDE = <Cipher-Suite-Name>
RECEIVE_SIDE = ALL
```

Cipher Suites Supported in TLSV1 when FIPS_MODE=YES

The following cipher suites are supported in TLSV1 when FIPS_MODE=YES:

Cipher Suite Name	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm	Message Digest Algorithm
DHE-RSA-AES256-SHA	DH	RSA	AES(256)	SHA1
DHE-DSS-AES256-SHA	DH	DSS	AES(256)	SHA1
AES256-SHA	RSA	RSA	AES(256)	SHA1
EDH-RSA-DES-CBC3-SHA	DH	RSA	3DES(168)	SHA1
EDH-DSS-DES-CBC3-SHA	DH	DSS	3DES(168)	SHA1
DES-CBC3-SHA	RSA	RSA	3DES(168)	SHA1
DHE-RSA-AES128-SHA	DH	RSA	AES(128)	SHA1
DHE-DSS-AES128-SHA	DH	DSS	AES(128)	SHA1
AES128-SHA	RSA	RSA	AES(128)	SHA1

Cipher Suites Supported in TLSV1 when FIPS_MODE=NO

The following cipher suites are supported in TLSV1 when FIPS_MODE=NO:

Cipher Suite Name	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm	Message Digest Algorithm
DHE-RSA-AES256-SHA	DH	RSA	AES(256)	SHA1
DHE-DSS-AES256-SHA	DH	DSS	AES(256)	SHA1
AES256-SHA	RSA	RSA	AES(256)	SHA1
EDH-RSA-DES-CBC3-SHA	DH	RSA	3DES(168)	SHA1

EDH-DSS-DES-CBC3-SHA	DH	DSS	3DES(168)	SHA1
DES-CBC3-SHA	RSA	RSA	3DES(168)	SHA1
DHE-RSA-AES128-SHA	DH	RSA	AES(128)	SHA1
DHE-DSS-AES128-SHA	DH	DSS	AES(128)	SHA1
AES128-SHA	RSA	RSA	AES(128)	SHA1
RC4-SHA	RSA	RSA	RC4(128)	SHA1
RC4-MD5	RSA	RSA	RC4(128)	MD5
EDH-RSA-DES-CBC-SHA	RSA	RSA	DES(56)	SHA1
EDH-DSS-DES-CBC-SHA	DH	DSS	DES(56)	SHA1
DES-CBC-SHA	RSA	RSA	DES(56)	SHA1
EXP-EDH-RSA-DES-CBC-SHA	DH(512)	RSA	DES(40)	SHA1
EXP-EDH-DSS-DES-CBC-SHA	DH(512)	DSS	DES(40)	SHA1
EXP-DES-CBC-SHA	RSA(512)	RSA	DES(40)	SHA1
EXP-RC2-CBC-MD5	RSA(512)	RSA	RC2(40)	MD5
EXP-RC4-MD5	RSA(512)	RSA	RC4(40)	MD5

Cipher Suites Supported in v3

SSLv3 can be used only with FIPS_MODE=NO. It uses the same cipher suites as TLSv1 with FIPS_MODE=NO.

Appendix A: Parameters

Use parameter values to control the variables that govern the behavior of CA XCOM Data Transport.

This section contains the following topics:

Using Parameters (see page 351)
Parameter Format (see page 352)
Syntax (see page 352)
Guidelines (see page 353)
List of Parameters (see page 353)

Using Parameters

Parameter values can be defined to CA XCOM Data Transport as indicated in the table below. The values take precedence depending on where they are specified. The order in which parameter values take precedence is as follows:

Order of Precedence	Parameter Value Specified in	Explanation
1	Command line	If the user specifies a value for a parameter on the command line, this overrides every other specification in the files or the program.
2	xcom.cnf or *filename.cnf filename.xml or GUI transfer record*	If the value is specified in the xcom.cnf file, or in a user-customized configuration file, *filename.cnf*, or in the transfer record created in the GUI or in an .xml configuration file saved through the GUI, it overrides the value in the xcom.glb file.
3	xcom.glb	If the value is specified in the xcom.glb file, it overrides the value in the program. The values specified in xcom.glb are generally used by the system administrator to start CA XCOM Data Transport.
4	Program defaults	If a value is not specified anywhere, the program has its own defaults.

Parameter Format

When using CA XCOM Data Transport from the command prompt or with a script, and when editing configuration files, the CA XCOM Data Transport parameters consist of assignment statements. The format for assignment statements is as follows:

- PARAMETER_NAME (always all uppercase, with underscore character (_) when indicated)
- An equal sign (=)
- A character string terminated by a new-line

Note: Parameter values are always in uppercase. However, when specifying directories, file names, user IDs and passwords, you may use uppercase and lowercase.

Syntax

The syntax for assignment statements is as follows:

```
PARAMETER_NAME=value
```

Example

In the following example, the parameter EXPIRATION_TIME is set to a value of 6000 seconds.

```
EXPIRATION_TIME=6000
```

This controls the maximum time in seconds that a transaction is held in the transfer queue after execution.

Guidelines

Note the following guidelines for using CA XCOM Data Transport parameters:

- When you type trailing spaces and tabs from the command line as part of a parameter value, they are stripped by the command line processor (the shell).
- When you type trailing spaces and tabs into a file or script with an editor, such as vi, they are treated as part of a parameter value. These trailing spaces and tabs should be removed. If they are not removed they can cause confusion and unpredictable or undesirable results.
- Empty lines and lines beginning with a pound sign (#) are discarded.

Examples:

In the following examples, the *spacespacespace* represents trailing spaces.

- If you type the following at the command line, the trailing spaces are ignored:

  `xcom62 -c1 LOCAL_FILE=`*`xyzspacespacespace`*

- If you type the following into any configuration file using an editor, the *spacespacespace* would be treated as part of the name of the LOCAL_FILE.

  `LOCAL_FILE=`*`xyzspacespacespace`*

List of Parameters

The following is a full list of parameters for CA XCOM Data Transport for UNIX and Linux. The default values listed here are based on the defaults as supplied in the sample files xcom.glb and xcom.cnf that come with CA XCOM Data Transport. Any defaults not specified by these files come from the program's own defaults.

AGE

Specifies the number of days of history records that are retained when a purge procedure is executed.

1 to 999

Indicates the number of days of history records to be retained when executing a purge.

Notes:

- After XCOMUTIL has been run, the history file comprises the current date's records plus *nnn* days of records. For example, if the current Julian date is 12300 and AGE=1 is specified, then XCOMUTIL purges any history records written on or before 12298. Only the current and previous day's history records are saved.
- Use the AGE parameter if you intend to run XCOMUTIL on a periodic basis. It allows you to set up a procedure in which a specific date does not have to be continually modified.
- This parameter is mutually exclusive with the DATE parameter.
- This parameter applies only to metatransfers that are initiated on a z/OS system.

AGE_TIME

The number of seconds before waiting queue entries are removed from queue. If the value is 0, the waiting queue entries never age and are never removed from the queue.

Range: 0 to 86313600 (999 days)

Default: 432000 (5 days)

ALLOCATION_TYPE

This parameter indicates the unit of storage allocation for a data set that was created on an IBM mainframe.

CYL

Cylinders

TRK

Tracks

BLK

Blocks

REC

Records

Default: CYL

Note: This parameter is a Version 2 parameter.

ATOE_FILENAME

The name of the file containing the ASCII-to-EBCDIC character conversion table.

This is a custom file used only for specifying custom translation tables from ASCII to EBCDIC, if needed.

Range: 0 to 256 characters

Default: $XCOM_HOME/convtab/atoe.tab

Note: $XCOM_HOME is an environment variable.

AUTH_TYPE

This parameter specifies the type of authentication to be used for transfers.

PAM

Enables Pluggable Authentication Modules authentication.

SYSTEM

Enables traditional UNIX authentication.

Default: SYSTEM

AVGREC

For a data set created on an IBM mainframe, Specifies the multiplier for Primary and Secondary allocation units when allocating based on the number of records. The record size is based on the value of the LRECL parameter.

U

Indicates that the PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records to allocate for.

K

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in thousands (so it would be the number specified multiplied by 1024).

For example, specifying 3 would be stating 3 K or 3072 records.

M

Indicates that PRIMARY_ALLOC and SECONDARY_ALLOC parameters specify the number of records in millions (so it would be the number specified multiplied by 1048576).

For example, specifying 2 would be stating 2M or 2097152 records.

Default: None

Note: This parameter applies only when the SPACE parameter specifies a value of “REC”. The parameter indicates that a file is being allocated based on a specific number of records.

BLKSIZE

Specifies the block size of a data set created on an IBM mainframe.

Note: Used when FILE_OPTION=CREATE

Range: 0 to 32767

If the format is…	Then the block size must be…
Fixed or fixed block record	A multiple of the record length
Variable record	Four bytes larger than the record length
Undefined record	Larger than the largest record length

Default: 800

CACHE_READ_SZ

This parameter specifies the size (in KB) of the cache for reading files.

nnnn (KB)

The size of the cache for reading files.

Range: 0 – 9999

Default: 0

Note: In the XCOM.GLB file, a value of 0 to 1023 can be entered for this parameter without causing an error, but it is interpreted differently, as follows:

- A value of 0 is interpreted to use the old method of reading files.
- A value of 1 to 1023 is automatically converted to the default value of 1024.

CACHE_WRITE_SZ

This parameter specifies the size (in KB) of the cache for writing files.

nnnn (KB)

The size of the cache for writing files.

Range: 0 – 9999

Default: 0

Note: In the XCOM.GLB file, a value of 0 to 1023 can be entered for this parameter without causing an error, but it is interpreted differently, as follows:

- A value of 0 is interpreted to use the old method of reading files.
- A value of 1 to 1023 is automatically converted to the default value of 1024.

CAPKIHOME

This parameter specifies the CAPKI/ETPKI library path. Set the CAPKI/ETPKI library path before using any of the following items:

- Password encryption transfers with cipher negotiation
- Encryption at rest transfers
- SSL transfers
- Transfers using CA XCOM Gateway

The CA XCOM Data Transport installer installs the CAPKI/ETPKI library. Typically it is installed to the CA common components directory. A default path is populated in XCOM.glb during installation. If CAPKI is installed at a different location, update CAPKIHOME with the correct path where libcapki.so resides.

Range: 0 to 256 characters

Default: The default depends on the platform, as follows:

- For AIX — CAPKIHOME=/opt/CA/SharedComponents/CAPKI/AIX/powerpc/32/lib
- For HP-UX PA RISC — CAPKIHOME=/opt/CA/SharedComponents/CAPKI/HP-UX/pa-risc/11.11_32_pa2.0/lib
- For HP-UX IA64 — CAPKIHOME= /opt/CA/SharedComponents/CAPKI/HP-UX/Itanium/64/lib
- For Linux — CAPKIHOME=/opt/CA/SharedComponents/CAPKI/Linux/x86/32/lib
- For Solaris — CAPKIHOME=/opt/CA/SharedComponents/CAPKI/SunOS/sparc/32/lib

CARRIAGE_CONTROL_CHARACTERS

Indicates the type of printer carriage-control codes, if any, that are included in the report file.

Note: For report transfers only.

ASA

ASA control codes in column 1.

IBM

IBM Machine Characters (valid only for IBM mainframes).

BYPASSASA

If data is already in ASA format, bypass conversion.

OTHER

No carriage-control codes are used.

Default: OTHER

CARRIAGE_FLAG

Specifies the type of file being transferred and some special characteristics of the conversion done during the transfer.

YES

Indicates that the transferred file is a text file and a newline character should be added to the end of incoming records. Also, newline characters are removed from the ends of lines before an outgoing record is sent.

NO

Indicates no special processing.

MPACK

Indicates a text file with record packing. Uses 2K pack buffer.

VLR

Indicates a binary file of variable-length records with a field of four bytes preceding each record. Applies to locally initiated transfers only.

XPACK

Indicates a text file with record packing. Uses 31K pack buffer.

Note: MPACK does not support a MAXRECLEN (actual record length) over 2K. XPACK does not support a MAXRECLEN (actual record length) over 31K.

Default: YES

CHARS

This parameter indicates the character set JES uses when the report is sent to a remote system.

xxxx

Specifies up to four alphanumeric characters representing the character set JES uses when a report is sent to a remote system.

Default: None

CHECKPOINT_COUNT

Defines how often (based on record count) the sending system requests a checkpoint to be taken. The value 0000 indicates no checkpointing.

Range: 0 to 9999

Default: 1000

Notes:

- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX
- This is a Version 2 parameter.

CKPT

Defines how often (based on record count) the sending system requests a checkpoint to be taken. The value 0 indicates no checkpointing.

Note: Setting CKPT affects performance. Each time a checkpoint is taken, the output buffers on the receiving system are written to the disk. On Token Ring, Ethernet, and other high-speed networks, CKPT should be set to 0 or to the highest allowable value. CKPT should not be set for small files.

Range: 0 to 9999

Default: 1000

Notes:

- Alias of CHECKPOINT_COUNT
- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX
- This is a Version 2 parameter.

CLASS

The print class assigned to a report transferred to a remote system.

If the remote system is an IBM mainframe, this field designates the JES SYSOUT class.

Note: For report transfers only.

Example:

Enter **B** to print the report through SYSOUT=B.

Range: 1 character

Default: None

CODE

Used to identify the type of data being transferred.

ASCII

An ASCII file is being transferred. This indicates that the incoming file is assumed to be ASCII format, and is not translated. Therefore the file on the remote system should be in ASCII format before it is transferred.

BINARY

A binary file, such as an executable file, is being transferred. This indicates to a remote system that it is not to translate the data it is exchanging with your system.

EBCDIC

An EBCDIC file is being transferred. The transferred data is translated from EBCDIC to ASCII format when the local system receives the data, and from ASCII to EBCDIC format when the local system sends the data.

UTF8

A Unicode file which is based on the UTF8 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

UTF16

A Unicode file which is based on the UTF16 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

Default: ASCII

Note:

- Alias of CODE_FLAG
- XPACK packing is selected automatically when UT8/UTF16 options are selected.
- Block level I/O is disabled automatically when UTF8/UTF16 options are selected.

CODE_FLAG

Used to identify the type of data being transferred.

ASCII

An ASCII file is being transferred. This indicates that the incoming file is assumed to be ASCII format, and is not translated. Therefore the file on the remote system should be in ASCII format before it is transferred.

BINARY

A binary file, such as an executable file, is being transferred. This indicates to a remote system that it is not to translate the data it is exchanging with your system.

EBCDIC

An EBCDIC file is being transferred. The transferred data is translated from EBCDIC to ASCII format when the local system receives the data, and from ASCII to EBCDIC format when the local system sends the data.

UTF8

A Unicode file which is based on the UTF8 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

UTF16

A Unicode file which is based on the UTF16 encoding system is being transferred. When this option is set, the LOCAL_CHARSET/ REMOTE_CHARSET parameters are used to identify the character encoding of the input file and the required encoding for the output file.

Default: ASCII

Note:

- XPACK packing is selected automatically when UT8/UTF16 options are selected.
- Block level I/O is disabled automatically when UTF8/UTF16 options are selected.

CODETABL

Specifies the prefix to the file names, atoe.tab and etoa.tab, that contain the external ASCII-to-EBCDIC and EBCDIC-to-ASCII custom character conversion tables. These custom character conversion tables determine which external translation tables will be used by the transfer.

This parameter is valid only if INTERNAL_CONVERSION_TABLES=NO.

Range: 0 to 3 alphanumeric characters

Default: None

COMPRESS

Indicates the compression type. When communicating with an IBM mainframe, if the data file contains any empty lines, COMPRESS can only be set to YES, LZSMALL, LZMEDIUM, or LZLARGE.

Notes:

- Not all compression types are supported on all platforms. For supported compression types, see the partner platform documentation.
- The LZ values enable LZ (Lempel-Ziv) compression to replace sequences of data bytes that occur more than once in a data stream with a code value.

COMPACT

RJE compaction algorithm optimized for uppercase English text.

LCOMPACT

RJE compaction algorithm optimized for lowercase English text.

LZLARGE

Activates LZ compression to search back 32K in the datastream for a matching string.

LZMEDIUM

Activates LZ compression to search back 16K in the datastream for a matching string.

LZRW3

General-purpose algorithm that runs fast and gives reasonable compression.

LZSMALL

Activates LZ compression to search back 4K in the datastream for a matching string.

NO

Indicates no compression.

RLE

Run length encoding.

YES

Indicates run length encoding of binary zeros and blanks only.

ZLIB*n*

Greater compression than LZRW3 but less than LZSMALL, LZMEDIUM, and LZLARGE. The *n* value can be 1 through 9.

HUFFMAN

Huffman compression technique will be used.

Default: YES

COMPRESS_PDS

The compress_pds parameter controls when a PDS dataset gets compressed for CA XCOM Data Transport z/OS partners that support this functionality. The value of the CMPRS_PDS_ALLOW parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMDFLT) determines if PDS compression is allowed. When this parameter is set to YES, then PDS compression can occur on the CA XCOM Data Transport z/OS partner.

NONE

Suppresses the compression of an output PDS dataset as part of a CA XCOM Data Transport transfer.

BEFORE

The output PDS dataset is compressed before the transfer of user data begins.

AFTER

The output PDS dataset is compressed after the transfer of user data has completed.

BOTH

The output PDS dataset is compressed both before and after the transfer of user data.

Default: NONE

Note: COMPRESS_PDS applies only to PDS data sets that are opened for output as the target of a CA XCOM Data Transport transfer.

CONTROL

For multiple transfers. Use the syntax for performing a single transfer and then separate parameters for different transfers in the same configuration file by using this parameter.

Range: NEWXFER or NONE

Default: NEWXFER

CONVERT_CLASSES

A character string containing print classes for EBCDIC-to-ASCII conversions to be performed. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

COPIES

The number of copies that are to be sent. If this parameter is not specified, the remote system queues one copy of the report to the system's default printer. For report transfers only.

Range: 1 to 999

Default: 1

CREATE_DIRECTORIES

Indicates whether to create the specified directory if it does not exist.

YES

Create the directory if it does not exist.

NO

Do not create the directory if it does not exist.

Default: YES

CREATEDELETE

This parameter specifies whether an existing data set can be deleted and a new data set allocated at the start of a FILEOPT=CREATE transfer. This functionality is for CA XCOM Data Transport z/OS partners with support for CREATEDELETE. The CREATEDELETE parameter in the z/OS default table (XCOMDFLT) or destination member (XCOMCNTL) specified by the z/OS CA XCOM Data Transport Administrator affects the functionality of this parameter.

When z/OS has CREATEDELETE=ALLOW:

YES

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

NO

If FILEOPT=CREATE and the data set exists, then the transfer fails with a catalog/file error.

When z/OS has CREATEDELETE=YES:

YES or NO

If FILEOPT=CREATE and the data set exists, then the data set is deleted and a new data set is allocated at the start of the transfer.

When z/OS has CREATEDELETE=NO:

YES or NO

If FILEOPT=CREATE and the data set exists, then the transfer fails with a catalog/file error.

Default: NO

Note:

- The attributes of the existing dataset are deleted and the new data set is allocated with the attributes specified in the transfer when CREATEDELETE=YES.
- CREATEDELETE applies only if the target data set is a sequential data set or an entire PDS/PDSE. CREATEDELETE is ignored for other types of data sets (such as PDS members, PDSE members, VSAM, and USS files).
- CREATEDELETE applies to relative GDGs when the dataset is specified using the fully qualified GxxxxVxx name.

DATACLAS

Specifies the name of the data class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

DEBUG_FLAG

This parameter specifies whether a trace will be output to the file /tmp/<tid>.TRA instead of to the default path $XCOM_HOME/trace/<tid>.TRA. In addition to the transfer trace information, the initialization will also be traced. The trace level is still taken from the variable XTRACE.

Note:

- The <tid> variable refers to the six-digit transfer ID number.
- $XCOM_HOME is an environment variable

YES

The trace is output to /tmp/<tid>

NO

The trace is output to $XCOM_HOME/trace/<tid>.TRA.

Default: NO

DEFAULT_CHARSET

This parameter specifies the default character set that CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, the value can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

Default: ISO-8859-1

DEFAULT_CONVERROR

This parameter identifies the appropriate action when the input file contains characters that cannot be converted. They are not included within the output character sets character repertoire.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

REPLACE

Replace each unconvertible character with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each unconvertible character with the Unicode character that the decimal value nnnnnnn identifies. (If the specified replacement character cannot be represented in the output character set, then the transfer is failed).

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of skipped characters.

FAIL

The transfer terminates with an error condition.

Default: FAIL

DEFAULT_DELIM

DEFAULT_DELIM specifies an optional encoding for which the specified DEFAULT_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list.

Additionally, specifies a colon separated list of record delimiters that are used to mark and detect the end of a record can be specified.

This parameter is used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

EBCDIC Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	

CRLF	CR
LFCR	LF
CRNL	CR
FF	

ASCII Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: ASCII:LF:CRLF

Tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which XX_CHARSET (LOCAL_CHARSET or REMOTE_CHARSET or DEFAULT CHARSET, whichever of these is in effect at the time of transfer) is based, it should not be used.

Example: If XX_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if XX_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in DEFAULT_DELIM, delimiters will be derived as explained below.

If DEFAULT_DELIM is not specified or doesn't contain at least one valid delimiter, platform default values are used as shown below,

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

DEFAULT_INPUTERROR

This parameter identifies the appropriate action when the input file contains data that is not consistent with the specified input character set.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

REPLACE

Replace each piece of erroneous data with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each piece of erroneous data with the Unicode character that the decimal value nnnnnnn identifies.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of ignored bytes.

FAIL

The transfer terminates with an error condition.

Default: FAIL

DEN

Specifies the density to be used in creating a tape on the remote system. Valid values are the same as those for the DEN parameter in JCL.

Range: 1 to 4

Default: None

DESTINATION

Identifies the printer or other device on the remote system where the report is to be sent. If this parameter is not specified, the remote system sends the report to the system's default printer. For report transfers only.

0 to 16 characters

For indirect transfers and for Version 1.

0 to 21 characters

For transfers that are not indirect and for Version 2.

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

DESTINATION_TYPE

The type of target system.

Range: HOST, MIDRANGE, or OTHER

Default: OTHER

DISPOSITION

Indicates what the remote system does with the report file after the report has been printed. For report transfers only.

Note: This field is not used when the remote system is an IBM mainframe.

DELETE

After printing the report is deleted.

KEEP

After printing the report is kept.

HOLD

After printing the report is held.

Default: DELETE

DOMAIN

The Windows domain name for use in authenticating the user ID and password when accessing a Windows-based machine that has sharable disks and drives that belong to that domain. This allows users to access these sharable drives without having to have a local user ID or password defined to the machine.

Range: 1 to 15 characters

Default: None

DROPSESS

Indicates whether CA XCOM Data Transport drops an LU-to-LU session at the conclusion of a scheduled file transfer.

YES

Indicates that CA XCOM Data Transport drops the session.

NO

Indicates that CA XCOM Data Transport does not drop the session.

QEMPTY

Indicates that CA XCOM Data Transport is to process all the transfers to a particular LU in the request queue before dropping the session.

Default: NO

DSNTYPE

This parameter specifies the data set definition.

Note: This parameter applies only to mainframe SMS data sets.

LIBRARY

Defines a PDSE.

PDS

Defines a partitioned data set.

BASIC

Defines a legacy sequential dataset.

LARGE

Defines a large format sequential dataset.

EXTREQ

Defines an extended format dataset.

EXTPREF

Specifies an extended format is preferred. If the extended format is not possible, a basic format is used.

<blank>

Defines a partitioned or sequential data set based on the data set characteristics entered.

Note: These values are IBM standards for SMS processing.

Range: One to eight characters

Default: None

EAR_CIPHER

Indicates which encryption algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
DES-CBC DES-ECB DES-CFB DES-OFB DES	YES/NO	Algorithm uses fixed key size(8 bytes)
3DES-CBC 3DES-ECB 3DES-CFB 3DES-OFB 3DES	YES/NO	Algorithm uses fixed key size(24 bytes)
AES128-CBC AES128-ECB AES128-CFB AES128-OFB AES128	YES/NO	Algorithm uses fixed key size(16 bytes)
AES192-CBC AES192-ECB AES192-CFB AES192-OFB AES192	YES/NO	Algorithm uses fixed key size(24 bytes)
AES256-CBC AES256-ECB AES256-CFB AES256-OFB AES256	YES/NO	Algorithm uses fixed key size(32 bytes)
RC2-CBC RC2-ECB RC2-CFB RC2-OFB RC2	NO	Product can choose from supported range
RC4-STREAM RC4	NO	Product can choose from supported range

Example:

```
EAR_CIPHER=3DES-CBC
```

Notes:

- This is one of a pair of parameters, as follows:
  - LEAR_CIPHER (local)
  - EAR_CIPHER (remote)
- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX

EAR_DIGEST

Indicates the digest value that is matched with the generated digest when the transfer type is a receive file. You must have already generated a digest value on the input file.

CA XCOM Data Transport generates the digest on the remote file in the case of a send file transfer. CA XCOM Gateway uses the generated digest to ensure the integrity of the file in the staging area.

Range: 0 to 128 characters

Default: None

Example:

```
ear_digest=CEA700113B151ECF63312EE6C6CFC2260AAE5FAB
```

Notes:

- The digest value specified in the example is the digest that is generated using the SHA1 digest algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual digest and the specified value must be in upper case.
- This parameter is one of a pair of parameters, as follows:
  - lear_digest (local)
  - ear_digest (remote)

EAR_HASH

Indicates which hash algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
SHA1	YES	Secure Hash Algorithm 1
MD5	NO	Message Digest-5 hash algorithm
MD4	NO	Message Digest-4 hash algorithm

Notes:

- This is one of a pair of parameters, as follows:
  - lear_hash (local)
  - ear_hash (remote)
- CA XCOM Data Transport automatically sets checkpoint_count to 0 if any of the following parameter values are set:
  - (l)ear_cipher=XXX-OFB
  - (l)ear_cipher=RC4
  - (l)ear_hash=XXX

EAR_KEY

Indicates the symmetric encryption key to be used to encrypt/decrypt.

Range: 0 to 128 characters

Default: None

Example:

```
ear_key=025BD5549BF743D99D86A2752019BCF48CC14AD5A16E0110
```

Notes:

- The key value specified in this example is a key for the 3DES encryption algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual key and the specified value must be in upper case.

EATTR

This parameter identifies if the dataset can have extended attributes when the dataset is allocated on an Extended Address Volume (EAV).

OPT

Specifies that a dataset can optionally have extended attributes.

NO

Specifies that a dataset cannot have extended attributes.

Default: None

Note: This parameter is applicable only for data set creation on an IBM mainframe.

EOL_CLASSES

A character string containing print classes for an ASCII newline that is appended to each record. For incoming report transfers only.

Range: 1 to 64 characters

Default: None

EPRTY

Indicates the execution priority for the request. The lowest priority is 1.

1 to 255

Specifies an execution priority from 1 to 255, where 1 is the lowest priority.

Notes:

- In an environment with multiple concurrent CA XCOM Data Transport transmissions, transfers with higher priorities receive preferential servicing.
- All other considerations being equal, give short file transfers a higher EPRTY than very long running transmissions.

Default: 16

ETOA_FILENAME

The name of the file containing the EBCDIC-to-ASCII character conversion table.

This is a custom file used only for creating custom translation tables from EBCDIC to ASCII, if needed.

Range: 1 to 256 characters

Default: $XCOM_HOME/convtab/etoa.tab

Note: $XCOM_HOME is an environment variable.

EXPDT

Specifies the expiration date to be placed on the tape data set being created.

yyddd

Specifies an expiration date for the tape data set as a two-digit designation for the year and a three-digit designation for the day of the year. For example, in the expiration date 13021, 13 is the year (namely, 2013) and 021 is the 21st day of that year. The tape data set will expire on January 21, 2013.

yyyy/ddd

Specifies an expiration date for the tape data set in terms of a four-digit designation for the year and a three-digit designation for the day of the year. For example, in the expiration date 2013/021, 2013 is the year and 021 is the 21st day of that year. The tape data set will expire on January 21, 2013.

Note: EXPDT and RETPD are mutually exclusive; specify one or the other.

EXPIRATION_TIME

The maximum time, in seconds, that a transaction is held in the transfer queue after completion. When the maximum time is reached, all references to the transaction are removed from the queue, including trace files and temporary files.

Note: If EXPIRATION_TIME is set to no value in xcom.glb, the program default of 6000 is used.

Range: 0 to 32767

Default: 6000

FCB

Identifies the FCB JCL parameter when sending the report file to an IBM mainframe, defining print density, lines per page, and so on. For report transfers only.

Range: Zero to four characters

Default: None

FILE_OPTION

Indicates how the transferred data is to be processed by the receiving system. For file transfers only.

For most file transfers, the parameter values are as follows:

CREATE

Create a new file on the receiving system.

APPEND

Append the transferred data to an existing file on the receiving system.

REPLACE

Replace an existing file on the receiving system.

For wildcard transfers, the parameter values are as follows:

CREATE

Create the PDS/Directory and add the transferred members. If the PDS/Directory already exists, the transfer fails with an error.

Note: The transfer will not fail if the PDS already exists if CREATEDELETE=YES is specified and the remote z/OS system has been set up to allow the PDS to be recreated through the default table (XCOMDFLT) or destination member (XCOMCNTL).

APPEND

Add transferred members/files. If the PDS/Directory does not exist or the member/file already exists, the transfer fails with an error.

REPLACE

Add or replace transferred members/files. If the PDS/Directory does not exist, the transfer fails with error XCOMN0403E Cannot open output file—No such file or directory.

Note: When creating a file on an IBM mainframe system, some additional information may be necessary. For more information, see the explanations for RECORD_FORMAT, BLKSIZE, VOLUME, and UNIT parameters.

Default: CREATE

FILE_OPTION_RF

Indicates how the transferred data is to be processed by the receiving system (that is, the local system). Used when the transfer type is Retrieve File. If a value is not specified, then the value of FILE_OPTION is used. If no default is specified in FILE_OPTION or FILE_OPTION_RF, then the value defaults to CREATE.

For most file transfers, the parameter values are as follows:

CREATE

Create a new file on the receiving system.

APPEND

Append the transferred data to an existing file on the receiving system.

REPLACE

Replace an existing file on the receiving system.

For wildcard transfers, the parameter values are as follows:

CREATE

Create the PDS/Directory and add the transferred members. If the PDS/Directory already exists, the transfer fails with an error.

Note: The transfer will not fail if the PDS already exists if CREATEDELETE=YES is specified and the remote z/OS system has been set up to allow the PDS to be recreated through the default table (XCOMDFLT) or destination member (XCOMCNTL).

APPEND

Add transferred members/files. If the PDS/Directory does not exist or the member/file already exists, the transfer fails with an error.

REPLACE

Add or replace transferred members/files. If the PDS/Directory does not exist, the transfer fails with error XCOMN0403E Cannot open output file-No such file or directory.

Default: CREATE

FILEDATA

Indicates how a remote USS file is to be allocated.

B

Binary

T

Text

If you do not specify a value for FILEDATA, then the allocation is determined based on the CODE= specification, as determined by the value of the EBCDIC/Binary/ASCII/VLR(E/B/A/V) field, as follows:

- If EBCDIC/Binary/ASCII/VLR(E/B/A/V)=B, then the file is allocated and processed as binary data.
- If EBCDIC/Binary/ASCII/VLR(E/B/A/V)=E, the type of allocation and processing depends on the value of FILEDATA, as follows:
  - If you do not specify FILEDATA=B, then the file is allocated as a text file and processed as an EBCDIC text file.
  - If you do specify FILEDATA=B, then the file is allocated as a binary file, but processed as an EBCDIC file.

Important! If you do specify FILEDATA=B, then you need to specify a value for USSLRECL, to tell CA XCOM Data Transport how many bytes there are in each logical record.

FIPS_MODE

This parameter specifies the mode to enable or disable the FIPS-compliant cipher and digest algorithms to be used for SSL transfers or for encryption at rest.

YES

Indicates that encryption/decryption will be running in FIPS mode.

NO

Indicates that encryption/decryption will not be running in FIPS mode.

Default: NO

FORM

The type of form that should be used to print the report. Because CA XCOM Data Transport places the print job in the remote system's print queue, the print control functions depend on the remote system. The user must verify beforehand that the requested form is available at the remote site. For report transfers only.

Note: When sending a report to a VAX computer, leave this parameter blank unless you are certain that this is a valid form type. VMS interprets this to mean that no special form is being requested.

Range: 0 to 10 characters

Default: None

GATEWAY_IP

This parameter indicates the CA XCOM Gateway domain (or IP address) to which this instance of the CA XCOM Data Transport server is associated.

Range: 1 to 64 characters

Default: None

GATEWAY_PORT

This parameter indicates the Gateway port to which this instance of the CA XCOM Data Transport server is allied.

For Gateway transfers only.

Default: None

GATEWAY_VERSION

Indicates the CA XCOM Gateway version to which this instance of the CA XCOM Data Transport server is allied.

For CA XCOM Gateway transfers only.

R115

Indicates that the CA XCOM Data Transport server is allied to the R11.5 Gateway

R116

Indicates that the CA XCOM Data Transport server is allied to the R11.6 Gateway

Default: R115

GATEWAYCERT

This parameter specifies the file name for the certificate that will be passed to CA XCOM Gateway to confirm that the request is from an authorized CA XCOM Data Transport server.

For CA XCOM Gateway transfers only.

Range: 1 to 256 characters

Default: None

GATEWAYCPASS

This parameter specifies the password for the CA XCOM Gateway certificates when they are created with a password. If the CA XCOM Gateway certificates were not created with a password, then this parameter is ignored.

For CA XCOM Gateway transfers only.

Range: 1 to 32 characters

Default: None

GATEWAYGUID

This parameter identifies the remote file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. This is a unique value that identifies each CA XCOM Gateway file.

Note: When the CA XCOM Gateway GUID is not known, the keyword ANY can be used to identify the remote file as a CA XCOM Gateway file.

Range: 0 to 36 characters

Default: None

GATEWAYPKEY

This parameter specifies the path and file name for the CA XCOM Gateway certificate's private key.

For CA XCOM Gateway transfers only.

Range: 1 to 256 characters

Default: None

GATEWAYPROTOCOL

Identifies the protocol that the Gateway uses to access the external server

This parameter must be specified as 'FTP' or 'SFTP' or 'FTPS' for Transfers between associated Gateway and External Server. In all other cases it will be XCOM by default.

FTP

For transfers between Gateway and registered FTP Server

SFTP

For transfers between Gateway and the registered SFTP Server

FTPS

For transfers between Gateway and the registered FTPS Server

XCOM

Default used for Transfers between Gateway and the XCOM server

Default: XCOM

Note:

- This parameter will be used only when requesting for Extract file from Gateway Environment to External Server or Insert file from External Server to Gateway Environment.
- This can be used only if the associated Gateway version is R11.6
- This parameter is not valid for SEND JOB and SEND REPORT actions.

HIST_FILE

This parameter specifies the complete path information of the file to contain the history records returned by an inquire metatransfer (-c6) or a get history record retrieval metatransfer (-c7).

Range: 0 to 256 characters

Default: None

HOLD_SESS

Specifies whether a session is to be held.

Y

The session is held.

N

The session is not held.

HOLD_TRANSFER

Specifies whether a queued transfer is to be held.

Valid only when QUEUE=YES.

Y

The queued transfer is held.

N

The queued transfer is not held.

HOLD

Prevents a TYPE=SCHEDULE transfer from starting until explicitly released.

YES

This transfer is not initiated until it is released in one of the following ways:

- Through the Menu Interface
- By an operator with a RELEASE command
- Through the XTC facility

NO

This transfer is not held.

Default: NO

HOLDCOUNT

Associates a number with a transfer request that is incremented or decremented by the successful or unsuccessful completion of other transfer requests. As long as the number is greater than 0, the transfer is not released.

0 to 255

Specifies a value that controls the holding/releasing of a transfer request. The transfer is released when the value of the parameter reaches 0.

Note: See the description of the parameters XTCERRDECR, XTCERRINCR, XTCGOODDECR, and XTCGOODINCR, which can decrement and increment the value of the HOLDCOUNT parameter.

Default: 0

HOLDFLAG

Indicates whether a transferred report file is to be placed on HOLD on the remote system or is to be printed immediately. For report transfers only.

Range: YES or NO

Default: NO

IDENT

Allows the user to enter information to identify the file transfer request.

Note: This is equivalent to the TRANSFER_ID field on Windows.

Range: 0 to 10 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

INQ_FILE

Specifies the complete path information of the file to contain the information required to inquire about the transfer.

Range: 1 to 256 characters

Default: None

INQ_WAIT

Specifies how long CA XCOM Data Transport should wait - in hours (*hh*), minutes (*mm*), and seconds (*ss*) - for transfers to complete.

hhmmss

Specifies in hours (*hh*), minutes (*mm*) and seconds (*ss*) the length of the time that CA XCOM Data Transport should wait for a transfer request to complete.

Default: 001000 (10 minutes)

Note: This parameter's value is expressed as a number of up to six digits (for example, 010000 for 1 hour).

INTERNAL_CONVERSION_TABLES

Indicates whether internal or external conversion tables should be used for ASCII-to-EBCDIC conversion and EBCDIC-to-ASCII conversion. The external conversion files may be customized to meet your needs. For the appropriate directory and file name, see ATOE_FILENAME and ETOA_FILENAME.

YES

Use internal conversion tables.

NO

Use the external tables contained in ATOE_FILENAME and ETOA_FILENAME.

Default: YES

IPNAME

For TCP/IP protocols, the name of the remote system that receives a file, job, or report.

The value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

IPPORT

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 1 to 65535

Default: 8044

LABEL

Indicates the type of label associated with a tape data set.

Processing type: AL, AUL, BLP, LTM, NL, NSL, SL, SUL

Represents the type of processing to be applied to data sets on tape.

Note: CA XCOM Data Transport for z/OS supports only standard label tapes.

Example:

LABEL=BLP

The type of processing to be applied to this data set is BLP.

Default: AL

LABELNUM

Indicates the sequence number of the data set on the tape.

Sequence number (0001 to 9999)

This value identifies the sequence number of a data set on tape.

Example:

LABELNUM=2

This specification refers to the second data set on the tape.

Default: 0001

LCLNTFYL

Specifies the local user notification level.

A or ALL

Notify on transfer completion.

W or WARN

Notify only if the transfer received a warning or error.

E or ERROR

Notify only if the transfer received an error.

Default: ALL

LDOMAIN

The domain associated with LUSERID and LPASSWORD, if the target system is UNIX or Linux.

Range: 1 to 15 characters

Default: None

LEAR_CIPHER

Indicates which encryption algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
DES-CBC DES-ECB DES-CFB DES-OFB DES	YES/NO	Algorithm uses fixed key size(8 bytes)
3DES-CBC 3DES-ECB 3DES-CFB 3DES-OFB 3DES	YES/NO	Algorithm uses fixed key size(24 bytes)
AES128-CBC AES128-ECB AES128-CFB AES128-OFB AES128	YES/NO	Algorithm uses fixed key size(16 bytes)
AES192-CBC AES192-ECB AES192-CFB AES192-OFB AES192	YES/NO	Algorithm uses fixed key size(24 bytes)
AES256-CBC AES256-ECB AES256-CFB AES256-OFB AES256	YES/NO	Algorithm uses fixed key size(32 bytes)
RC2-CBC RC2-ECB RC2-CFB RC2-OFB RC2	NO	Product can choose from supported range
RC4-STREAM RC4	NO	Product can choose from supported range

Example:

```
LEAR_CIPHER=3DES-CBC
```

Notes:

- This is one of a pair of parameters, as follows:
  - LEAR_CIPHER (local)
  - EAR_CIPHER (remote)
- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX

LEAR_DIGEST

Indicates the digest value that is matched with the generated digest when the transfer type is a send file, job, or report. You must have already generated a digest value on the input file.

CA XCOM Data Transport generates the digest on the local file in the case of a receive file transfer. CA XCOM Gateway uses the generated digest to ensure the integrity of the file in the staging area.

Range: 0 to 128 characters

Default: None

Example:

```
lear_digest=CEA700113B151ECF63312EE6C6CFC2260AAE5FAB
```

Notes:

- The digest value specified in the example is the digest that is generated using the SHA1 hash algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual digest and the specified value must be in upper case.
- This parameter is one of a pair of parameters, as follows:
  - lear_digest (local)
  - ear_digest (remote)

LEAR_HASH

Indicates which hash algorithm is to be used.

Default: None

Value	FIPS MODE	Comments
SHA1	YES	Secure Hash Algorithm 1
MD5	NO	Message Digest-5 hash algorithm
MD4	NO	Message Digest-4 hash algorithm

Notes:

- This is one of a pair of parameters, as follows:
  - LEAR_HASH (local)
  - EAR_HASH (remote)
- CA XCOM Data Transport automatically sets CHECKPOINT_COUNT to 0 if any of the following parameter values are set:
  - (L)EAR_CIPHER=XXX-OFB
  - (L)EAR_CIPHER=RC4
  - (L)EAR_HASH=XXX

LEAR_KEY

Indicates the symmetric encryption key to be used to encrypt/decrypt.

Range: 0 to 128 characters

Default: None

Example:

```
lear_key=025BD5549BF743D99D86A2752019BCF48CC14AD5A16E0110
```

Notes:

- The key value specified in this example is a key for the 3DES encryption algorithm.
- CA XCOM Data Transport accepts the hexadecimal value of the actual key and the specified value must be in upper case.

LFILE

The name of the file on the local system that is being transferred. At the command prompt or in a script, if this variable is null or unset, standard input is read. In this manner, CA XCOM Data Transport commands can be used in a pipeline or with redirection. All Windows and UNC file naming conventions apply. If more than one file is included in the transfer, L_FILE must include wildcard characters (*).

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

LGATEWAYGUID

Identifies the local file as a CA XCOM Gateway file and specifies the CA XCOM Gateway GUID. The CA XCOM Gateway GUID is a unique value that identifies each CA XCOM Gateway file. The keyword ANY can be used to identify the local file as a CA XCOM Gateway file when the CA XCOM Gateway GUID is not known.

Range: 0 to 36 characters

Default: None

LOCAL_CHARSET

LOCAL_CHARSET specifies the local character set CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

Default: The DEFAULT_CHARSET parameter in the CA XCOM Data Transport Global Parameters specifies the default.

LOCAL_DELIM

LOCAL_DELIM specifies an optional encoding for which the specified LOCAL_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list. If LOCAL_CHARSET is UTF based charset, UTF rules will be applied disregarding encoding specified in LOCAL_DELIM.

Additionally, a colon separated list of record delimiters can be specified that are used to mark and detect the end of a record.

This parameter is used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

EBCDIC Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
FF	
NA	

ASCII Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: The DEFAULT_DELIM parameter in the CA XCOM Data Transport Global Parameters specifies the default.

The tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which LOCAL_CHARSET is based, it should not be used.

Example: If LOCAL_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if LOCAL_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in LOCAL_DELIM, delimiters will be derived as explained below.

If LOCAL_DELIM is not specified or doesn't contain at least one valid delimiter, DEFAULT_DELIM parameter in the CA XCOM Data Transport Global Parameters is considered. If DEFAULT_DELIM is also not specified or doesn't contain at least one valid delimiter, platform default values are used as shown below,

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

Note: During Unicode conversion some of the delimiters may fail to convert to some code pages.

For example, In case of Receive File transfer, when LOCAL_CHARSET set to ASCII based charset and LOCAL_DELIM set to NL/CRNL, then transfer may fail. The Unicode conversion fails as ASCII based code pages may not have mapping character for NL/CRNL.

LOCAL_FILE

The name of the file on the local system that is being transferred. At the command prompt or in a script, if this variable is null or unset, standard input is read. In this manner, CA XCOM Data Transport commands can be used in a pipeline or with redirection. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk (*) as a file name to indicate that all files within the specified directory are to be transferred. For example, the statement LOCAL_FILE=/NAMES/* indicates that all files under the NAMES directory are to be transferred.

When a prefix is followed by an asterisk, all members beginning with a specific prefix are to be transferred. For example, LOCAL_FILE=/NAMES/AL* requests that files AL, ALEX, and ALICE are all to be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated. However, when sending files to a mainframe PDS, any file name over 8 characters in length is truncated. These systems do not recognize file extensions. For example, a file called *longfilename.txt* will be truncated to *longfile* and a file called *file.txt* will be truncated to *file*.

Note: If QUEUE=YES, the full path name must be specified.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LOCAL_FILE_RF

The file name that is created, appended, or replaced on the local system when it receives a file. At the command prompt or in a script, if this value is null or unset, then CA XCOM Data Transport writes to the stdout. All UNIX or Linux file naming conventions apply.

For wildcard transfers, use an asterisk (*) as a file name to indicate that multiple files will be received.

Example

```
LOCAL_FILE_RF=/PAYROLL/*.
```

If multiple files are received and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

For platforms that support it, you can specify a common file extension to be appended to each file name.

Example

```
LOCAL_FILE_RF=/PAYROLL/*.TXT.
```

Note: If QUEUE=YES, the full path name must be specified.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LOCAL_FILE_SJ

Indicates the name of the file on the local system to be sent as a job. All the UNIX or Linux file naming conventions apply. If this value is null or unset, then CA XCOM Data Transport reads the standard input file.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LOCAL_FILE_SR

Local file name to be sent as a report to the remote system. If this value is null or unset, then CA XCOM Data Transport reads the standard input file. For report transfers only.

Note: If QUEUE=YES, user must specify full path name.

Range: 0 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LOCAL_NOTIFY

Specifies the user on the local system who is to be notified that CA XCOM Data Transport has completed a transfer. CA XCOM Data Transport uses the NOTIFYL parameter to determine the type of notification to use.

Range: 0 to 64 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

LOG_CONNECT_MSG

This parameter specifies whether the CA XCOM Data Transport log file should include the following connection messages:

- XCOMN0785I
- XCOMN0786I
- XCOMN0793I
- XCOMN0805I
- XCOMN0811I
- XCOMN0812I
- XCOMN0813I
- XCOMN0814I
- XCOMN0818I

Range: YES or NO

Default: YES

LPASS

The password of the local user to be validated on the CA XCOM Data Transport server handling a meta-transfer.

LPASSWORD

The password of the local user to be validated on the CA XCOM Data Transport server handling a metatransfer.

Range: 1 to 31 characters

Default: None

LRECL

Specifies the actual or maximum length, in bytes, of a logical record. Corresponds to the JCL LRECL subparameter.

Range: 0 to 32767

If the format is…	Then the maximum length of a logical record must be equal to the…
Variable blocked record	Maximum record length plus four
Fixed or fixed block record	Constant record length

Default: 160

LUSER

The user ID to use on the CA XCOM Data Transport system receiving a meta-transfer.

Range: 1 to 12 characters

Default: None

Note: LUSER is an alias of LUSERID.

LUSERID

The user ID to use on the CA XCOM Data Transport system receiving a metatransfer.

Range: 1 to 12 characters

Default: None

MAIL_TYPE

This parameter specifies the type of MAIL server used for sending mail notifications.

MAPI

Windows MAPI server

SMTP

SMTP server

Default: None

MAX_QUEUE_ENTRIES

The maximum number of entries allowed in the transfer queue. Once the maximum number of queue entries is reached, subsequent transfer attempts are rejected by CA XCOM Data Transport.

Note: This value depends on the memory available when CA XCOM Data Transport is started.

Range: 0 to 32767

The value cannot be set larger than the maximum allowed shared memory segment divided by 512.

Default: 50

MAX_REMOTE_TCP

The maximum number of simultaneous remote TCP/IP transfers accepted by CA XCOM Data Transport. Transfers received after this limit has been reached are rejected.

Note: Setting this parameter to 0 prohibits any remote transfers from taking place.

Range: 0 to 999

Default: 32

MAX_SESSIONS_ENTRIES

The maximum number of partners that can be described in the XCOM.SES file.

Range: 1 to 999

Default: 15

MAXRECLEN

For Windows, UNIX, and Linux systems, the locally initiating CA XCOM Data Transport system determines the values for MAXRECLEN, TRUNCATION, and LRECL, for send and receive operations. When the local CA XCOM Data Transport system initiates a transfer of a text file, this parameter designates the length, in bytes, of the largest record that can be transferred. If a record length is longer than this value, CA XCOM Data Transport uses the value in the TRUNCATION parameter on the initiating side to determine whether to terminate the transfer or to truncate the record and continue the transfer. When CA XCOM Data Transport transfers binary files, this value indicates the length of the records that are transferred. On a receive operation, MAXRECLEN is set to whatever the LRECL value is on the initiating side.

Range: 1 to 32767

Default: 1024

MBCS_CONVERROR

MBCS_CONVERROR identifies the appropriate action when the input file contains characters that cannot be converted. They are not included within the output character sets character repertoire.

REPLACE

Replace each unconvertible character with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each unconvertible character with the Unicode character that the decimal value nnnnnnn identifies. (If the specified replacement character cannot be represented in the output character set, then the transfer is failed). This option is not supported for z/OS systems and is treated as REPLACE.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of skipped characters. This option is not supported for z/OS systems and is treated as REPLACE.

FAIL

The transfer terminates with an error condition.

Default: The DEFAULT_CONVERROR parameter in the CA XCOM Data Transport Global Parameters specifies the default.

MBCS_INPUTERROR

MBCS_INPUTERROR identifies the appropriate action when the input file contains data that is not consistent with the specified input character set.

REPLACE

Replace each piece of erroneous data with the default substitution characters defined for the Unicode character set.

REPLACE#nnnnnnn

Replace each piece of erroneous data with the Unicode character that the decimal value nnnnnnn identifies. This option is not supported for z/OS systems and is treated as REPLACE.

SKIP

The erroneous data is disregarded, but a warning message is issued at the end of the transfer. The message identifies that this condition occurred and provides a total count of the number of ignored bytes. This option is not supported for z/OS systems and is treated as REPLACE.

FAIL

The transfer terminates with an error condition.

Default: The DEFAULT_INPUTERROR parameter in the CA XCOM Data Transport Global Parameters specifies the default.

METACODE_CLASSES

Classes of print jobs saved in metacode format, a variable length record format. For incoming report transfers only.

Range: 0 to 64 characters

Default: None

MGMTCLAS

Specifies the name of the management class to use when allocating a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

NOTIFY_NAME

The user on the remote system who is to be notified when CA XCOM Data Transport completes a transfer.

Note: If the remote system is an IBM mainframe, CA XCOM Data Transport uses the value of NOTIFYR to determine the type of notification to deliver.

If the remote system is a UNIX or Linux system, the user receives a mail message.

Range: 0 to 12 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

NOTIFY_TERM

Specifies which terminals to write to if NOTIFYL=WRITE. If NOTIFY_TERM is not set, all users specified in LOCAL_NOTIFY are notified at the first terminal where they are logged in, as found in the system table.

Range: 0 to 256 characters

Default: None

NOTIFYL

The local user notification flag.

WRITE

A message is displayed on the workstation where the user is logged in.

MAIL

A mail message will be sent to the user.

NONE

No notification is sent.

ALL

A message is displayed on all workstations attached to the server.

Default: None

Note: The L in NOTIFYL indicates that the local system governs the processing of the resulting notification on that system.

NOTIFYR

Specifies the remote user notification type when sending data to a remote system.

WRITE

A message is displayed on the screen.

MAIL

A mail message is sent to the user.

TSO

The specified TSO user is notified.

WTO

CA XCOM Data Transport writes to the log only (WTO).

CICS

The specified CICS user is notified.

LU

The specified Logical Unit is notified.

ROSCOE

Notify CA Roscoe user.

NONE

No notification is sent.

ALL

Write to all users.

Default: None

Note: The R in NOTIFYR indicates that the remote system governs the processing of the resulting notification on that system.

NUM_OF_DIR_BLOCKS

This parameter specifies the number of directory blocks to allocate for a data set that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

NUMBER_OF_RETRIES

Maximum number of retries before a transfer is logged as failed and taken out of the transfer queue. If the value is 0, no retries are attempted.

Range: 0 to 255

Default: 1

Note: This is a Version 2 parameter.

OEDATE

Limits the history request to only those file transfers that were scheduled or completed on or before the end date and time.

YYYYMMDD

The end date used to limit the history request to only those file transfers that were scheduled or completed on or before the end date and time specified.

YYYY

The four-digit year

MM

The two-digit number of the month, as follows:

01 = January	02 = February	03 = March
04 = April	05 = May	06 = June
07 = July	08 = August	09 = September
10 = October	11 = November	12 = December

DD

The two-digit day of the month (01 to 31)

Default: None

Notes:

- OEDATE and OETIME form an end date and time used to limit the history request to only those file transfers that were scheduled or completed on or before the date and time specified.
- See OSDATE and OSTIME for the start date and time.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OETIME

Limits the history request to only those file transfers that were scheduled or completed on or before the end date and time.

HHMMSS

The end time used to limit the history request to only those file transfers that were scheduled or completed on or before the end date and time specified.

HH

The two-digit hour (00 through 23)

MM

The two-digit minute (00 through 59)

SS

The two-digit second (00 through 59)

Default: 235959

Notes:

- OEDATE and OETIME form an end date and time used to limit the history request to only those file transfers that were scheduled or completed on or before the date and time specified.
- See OSDATE and OSTIME for the start date and time.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OFILE

Specifies the file name, local, or remote, to match for a history request.

You can use the following wildcard characters when you specify the file name:

*** or %**

Represents a string of zero or more characters.

_

Represents any single character.

Range: 1 to 256

Default: None

Example:

An OFILE value of %MASTER.FIL_.G* tells CA XCOM Data Transport to locate a file with following attributes:

- Starting with anything
- Ending with anything
- With the characters MASTER.FIL found in the name, followed by any single character and .G.

Notes:

- Supported in –c7 CNF meta transfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

OFILECASE

Specifies whether the specified file name (OFILE parameter) search is case-sensitive.

YES

The value that is specified for the OFILE parameter is case-sensitive.

NO

The value that is specified for the OFILE parameter is not case-sensitive.

Default: NO

Notes:

- For a case-sensitive search, case-sensitive collation is set to the history database table. Such collation can also be set to 'file' and 'lfile' columns in the history database table.
- Supported in –c7 CNF meta transfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.

OFILETYPE

Limits the history request to those transfers with the specified FILETYPE.

JOB

Restrict the search for file transfers to only FILETYPE JOB transfers.

REPORT

Restrict the search for file transfers to only FILETYPE REPORT transfers.

FILE

Restrict the search for file transfers to only FILETYPE FILE transfers.

Default: None

OFLMAX

Limits the history request to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

***NNNNNNNNN*(*N*|*X*)**

A 1- to 10-digit number, where the last digit can be either another numeric digit or a 1-character qualifier. This parameter is used to restrict the search for file transfers to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

X

One of the following qualifiers (default B):

- B = Bytes
- K = Kilobytes
- M = Megabytes
- G = Gigabytes

Default: None

Notes:

- Use OFLMIN and OFLMAX to specify a range that can be used to limit the history request by number of bytes transferred.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OFLMIN

Limits the history request to only those file transfers where the number of bytes transferred is equal to or greater than the value specified.

***NNNNNNNNN*(*N*|*X*)**

A 1- to 10-digit number, where the last digit can be either another numeric digit or a 1-character qualifier. This parameter is used to restrict the search for file transfers to only those file transfers where the number of bytes transferred is equal to or less than the value specified.

X

One of the following qualifiers (default B):

- B = Bytes
- K = Kilobytes
- M = Megabytes
- G = Gigabytes

Default: None

Notes:

- Use OFLMIN and OFLMAX to specify a range that can be used to limit the history request by number of bytes transferred.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OID

Limits the history request to only those file transfers with a specific transfer ID. The transfer ID is a user-defined identifier for file transfer requests.

xxxxxxxxxx

A 1- to 10-character transfer ID used to limit the history request to only those file transfers that contain the specified transfer ID.

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **TRANSFER01** returns the same results as specifying **transfer01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OINIT

Limits the history request to only locally initiated transfers or only remotely initiated transfers.

LOCAL or L

Restrict the search for file transfers to only locally initiated transfers.

REMOTE or R

Restrict the search for file transfers to only remotely initiated transfers.

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OJOB

Specifies the invoking job name to match for a history request.

Range: 1 to 8 characters

Default: None

Notes:

- Supported in –c7 CNF meta transfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.
- You can use the * wildcard character when you specify the job name.

OLMSG

Limits the history request by the transfer's last message. The format to use for CA XCOM Data Transport messages is as follows:

XCOM*XNNNNS*

A 1- to 10-character name used to restrict the search for file transfers to those where the last message matches the value specified.

XCOM

Indicates that the message is from CA XCOM Data Transport.

X

Identifies the system.

NNNN

Is the message number.

S

Is the message severity:

- I = Informational
- W = Warning
- E = Error

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is case sensitive.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OLOCATN

Limits the history request to transfers in one of the following locations:

- Transfers that have been completed and have been stored in a database
- Transfers still in the queue

DATABASE

Restrict the search for file transfers to those stored in the database. This applies only if a history database has been set up at your site.

QUEUE

Restrict the search for file transfers to those still in the queue.

Default: Transfers located in both the QUEUE and DATABASE.

Notes:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OLU

Limits the history request to only those file transfers with a specific remote LU name.

xxxxxxxx

A one- to eight-character LU name used to limit the history request to only those file transfers for the specified remote LU.

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **LU01** returns the same results as specifying **lu01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OREQ

Limits the history request to only those file transfers that contain this specific request number.

NNNNNN

A one- to six-character request number used to limit the history request to a specific request number.

Default: All request numbers

Note: You can use symbolic variables with this parameter in batch SYSIN01. For more information, see Symbolic Parameters in Configuration Files.

OSDATE

Limits the history request to only those file transfers that were scheduled or completed on or after the start date and time.

YYYYMMDD

The start date used to limit the history request to only those file transfers that were scheduled or completed on or after the start date and time specified.

YYYY

The four-digit year

MM

The two-digit number of the month, as follows:

01 = January	02 = February	03 = March
04 = April	05 = May	06 = June
07 = July	08 = August	09 = September
10 = October	11 = November	12 = December

DD

The two-digit day of the month (01 to 31)

Default: None

Notes:

- OSDATE and OSTIME form a start date and time used to limit the history request to only those file transfers that were scheduled or completed on or after the date and time specified.
- See OEDATE and OETIME for the end date and time.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OSTIME

Limits the history request to only those file transfers that were scheduled or completed on or after the start date and time.

HHMMSS

The start time used to limit the history request to only those file transfers that were scheduled or completed on or after the start date and time specified.

HH

The two-digit hour (00 through 23)

MM

The two-digit minute (00 through 59)

SS

The two-digit second (00 through 59)

Default: 235959

Notes:

- OSDATE and OSTIME form a start date and time used to limit the history request to only those file transfers that were scheduled or completed on or after the date and time specified.
- See OEDATE and OETIME for the end date and time.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

OSYSID

Limits the history request to only those file transfers executed on the specified system ID.

Note: OSYSNAME and OSYSID together uniquely identify a CA XCOM Data Transport r11.5 and higher server.

Range: 1 to 4 characters

Default: None

Note:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OSYSNAME

Limits the history request to only those file transfers executed on the specified system name.

Note: OSYSNAME and OSYSID together uniquely identify a CA XCOM Data Transport r11.5 and higher server.

Range: 1 to 8 characters

Default: None

Note:

- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OTNAME

Limits the history request to only those file transfers with a specific remote TCP/IP name or TCP/IP address.

xxxxxxxx...x

A 1- to 64-character TCP/IP name or address used to limit the history request to only those file transfers for the specified TCP/IP name or address.

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **TCPIP01** returns the same results as specifying **tcpip01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OTYPE

Specifies if the history request should include inactive transfer requests, active transfer requests, or completed transfers.

I

Selecting this option retrieves history records for file transfers whose execution is still pending.

A

Selecting this option retrieves history records for file transfers that are currently in progress.

C

Selecting this option retrieves history records for file transfers that have been successfully or unsuccessfully completed.

ALL|AIC|*

Selecting this option retrieves history records for all file transfers, independent of their status.

Default: AIC

Note: You can also specify values in combination; for example, specify AI to request history records for file transfers whose execution status is inactive and active.

OTYPEREQ

Limits the history request to only send transfers or only receive transfers.

SEND or S

Restrict the search for file transfers to only send transfers.

RECEIVE or R

Restrict the search for file transfers to only receive transfers.

Default: None

Note: This parameter is case sensitive.

OUSER

Limits the history request to only those file transfers submitted by a specific user.

xxxxxxxxxxx

A 1- to 12-character user name used to limit the history request to only those file transfers submitted by the specified user.

Default: None

Notes:

- The wildcard character, *, can be used for this parameter only when specified as the last character.
- This parameter is not case-sensitive. So filtering history records based on this parameter returns case-insensitive results. For example, specifying **USER01** returns the same results as specifying **user01**.
- Supported in –c7 CNF metatransfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer will be authorized to access.

OVOL

Specifies the volser (local or remote) to match for a history request.

Range: 1 to 6 characters

Default: None

Notes:

- Supported in –c7 CNF meta transfers only.
- The users defined in the groups XCOMADM and/or XCOMSADM determine what records the user running the get history record retrieval transfer is authorized to access.
- You can use the * wildcard character when you specify the volser.

PASSWORD

The password associated with the user ID on a remote system.

Range: 0 to 31 characters

Default: None

PORT

The number of the TCP/IP port on the remote CA XCOM Data Transport server. Used for TCP/IP transfers only.

Range: 1 to 65535

Default: 8044

PRIMARY_ALLOC

This parameter identifies primary storage allocation for a data set that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 1

Note: This parameter is a Version 2 parameter.

PRIORITY

Indicates the priority that CA XCOM Data Transport uses for scheduling a transfer. If two transfers are scheduled for the same time, the one with the high priority is processed before one with a normal or low priority.

HIGH

Set high priority.

NORMAL

Set medium priority.

LOW

Set low priority.

Default: NORMAL

PROGLIB

Specifies whether the transfer is a z/OS transfer of a PDSE program library or not.

YES

Indicates a z/OS transfer of a PDSE program library.

NO

Indicates that the transfer is not of a PDSE program library.

Default: None

PROTOCOL

The type of communication protocols to use.

SNA

For transfers using SNA/APPC communication protocols

TCPIP

For transfers using TCP/IP communication protocols

Default: SNA

Note: If only the TCP/IP component is installed with the base components of CA XCOM Data Transport, the default value is TCPIP.

QUEUE

Indicates whether to execute the transmission request immediately or to allow the request to be queued. If the user does not specify a .cnf file, and has not changed a .cnf file, the default value is YES.

Note: If NO is specified and the remote system is unavailable, the request aborts. If YES is specified, START_TIME and START_DATE are read.

YES

The transfer request goes into a queue and executes depending on the traffic in the queue and START_DATE and START_TIME.

NO

The transfer starts immediately.

Default: YES

Important! If you load a .cnf file into the CA XCOM Data Transport GUI, the value (QUEUE=YES or QUEUE=NO) specified in the .cnf file is shown correctly in the generated xml file. However, when you submit the transfer from the CA XCOM Data Transport GUI, it is always treated as though QUEUE=YES.

QUEUE_PATH

Directory containing the transfer queue data files.

Range: 0 to 256 characters

Default: $XCOM_HOME/Q

Note: $XCOM_HOME is an environment variable.

RECORD_FORMAT

Specifies the record format of a data set created on an IBM mainframe. This corresponds to the JCL RECFM subparameter.

Range: The range of values is listed in the following table:

Value	Description	Record Length	Comment
F	Fixed unblocked	The same length as the data set	

Value	Description	Record Length	Comment
FA	Fixed unblocked ANSI	The same length as the data set	Contains ISO/ANSI/FIPS control characters
FB	Fixed blocked	Fixed	Fixed record length with multiple records per block
FBA	Fixed blocked ANSI	Fixed	Multiple records per block where these records contain ISO/ANSI/FIPS control characters
FBS	Fixed blocked spanned	Fixed	Multiple records per block written as standard blocks
FM	Fixed unblocked machine	The same length as the data set	Contains machine code control characters
FS	Fixed unblocked spanned	The same length as the data set	Written as standard blocks where these records do not contain any truncated blocks or unfilled tracks
U	Undefined	Undefined	
V	Variable unblocked	Variable	
VA	Variable unblocked ANSI	Variable	Contains ISO/ANSI/FIPS control characters
VB	Variable blocked	Variable	Multiple records per block
VBA	Variable blocked ANSI	Variable	Multiple records per block where these records contain ISO/ANSI/FIPS control characters
VBM	Variable blocked machine	Variable	Multiple records per block where these records contain machine code control characters
VBS	Variable blocked spanned	Variable	May have multiple records per block where these records can span more than one block
VM	Variable unblocked machine	Variable	Contains machine code control characters
VS	Variable unblocked spanned	Variable	A record can span more than one block

Default: VB

RELEASE

Specifies whether the remote z/OS partner is to release unused DASD space when creating a new file.

YES

The remote partner is to release unused DASD space.

The unused DASD space that is specified for the transfer is released when the file is closed at the end of the transfer.

NO

The remote partner is not to release unused DASD space.

Default: No

REMOTE_CHARSET

REMOTE_CHARSET specifies the remote character set CA XCOM Data Transport uses for Unicode transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

When the remote system is CA XCOM Data Transport for z/OS:

CCSID#nnnnn/tttttttt

nnnnnn - specifies the CCSID number that corresponds to the character set of the local file. Valid values are 1 - 65535.

tttttttt (optional) – specifies the technique search order IBM Unicode Services uses when performing conversion. From 1 to 8 characters are specified. Valid values to use are:

R - Roundtrip conversion

E - Enforced Subset conversion

C - Customized conversion

L - Language Environment Behavior conversion

M - Modified for special use conversion

B - Bidi transformation (Bi-directional) conversion

0-9 - User defined conversions

Note: If the technique search order is not specified, Unicode Services defaults to 'RECLM'.

When the remote system is CA XCOM Data Transport for Windows or CA XCOM Data Transport for Linux/Unix:

CCSID#nnnnn

nnnnnn - specifies the CCSID number that corresponds to the character set. Valid values are 1 - 65535.

Alternatively, can be specified as an IANA character set name, or (ICU) acceptable alias name.

Range: 0 to 60 characters

Default: The DEFAULT_CHARSET parameter specifies the default in the CA XCOM Data Transport Default Options Table/Global Parameters on the remote system.

REMOTE_DELIM

REMOTE_DELIM specifies an optional encoding for which the specified REMOTE_CHARSET is based. The encoding can be either ASCII or EBCDIC. If specified, the encoding must be the first option in the list. If REMOTE_CHARSET is UTF based charset, UTF rules will be applied disregarding encoding specified in REMOTE_DELIM.

Used only for UNICODE transfers (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

When the remote system is CA XCOM Data Transport for z/OS:

Additionally it specifies the delimiter to use for USS-based output files when FILEDATA=TEXT.

Valid options:

EBCDIC – The specified character-set is EBCDIC encoded.

NA – Not applicable, the system default delimiter is used.

NL – New line

CR – Carriage return

LF – Line feed

CRLFL – Carriage return/Line feed

LFCR – Line feed/Carriage return

CRNL – Carriage return/New line

Note: The only valid encoding for CA XCOM Data Transport for z/OS is EBCDIC.

When the remote system is CA XCOM Data Transport for Windows or CA XCOM Data Transport for Linux/Unix:

Additionally, specifies a colon separated list of record delimiters that are used to mark and detect the end of a record.

EBCDIC Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR

FF	
NA	

ASCII Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	

UTF Rules

Record Delimiter Option	Mutually exclusive with
CR	CRLF, CRNL
LF	LFCR
NL	
CRLF	CR
LFCR	LF
CRNL	CR
VT	
FF	
LS	
PS	

Range: 0 to 60 characters

Default: The DEFAULT_DELIM parameter specifies the default in the CA XCOM Data Transport Default Options Table/Global Parameters on the remote system.

The tables indicate valid record delimiters for various encodings. If a record delimiter is not applicable for encoding on which REMOTE_CHARSET is based, it should not be used.

Example: If REMOTE_CHARSET is not UTF based, PS and LS delimiters should not be used. Similarly if REMOTE_CHARSET is EBCDIC based, VT, PS, and LS delimiters should not be used.

In these cases, these delimiters will be treated as invalid and disregarded if used. If no valid delimiter is found in REMOTE_DELIM, delimiters will be derived as explained below.

If REMOTE_DELIM is not specified or doesn't contain at least one valid delimiter, DEFAULT_DELIM parameter in the CA XCOM Data Transport Default Options Table/Global Parameters on the remote system is considered. If DEFAULT_DELIM is also not specified or doesn't contain at least one valid delimiter, platform default values are used as shown below,

- Default for ASCII: 'LF,CRLF'
- Default for EBCDIC: 'NL'

Note: During Unicode conversion some of the delimiters may fail to convert to some code pages.

For example, In case of Send File transfer, when REMOTE_CHARSET set to ASCII based charset and REMOTE_DELIM set to NL/CRNL, then transfer may fail. The Unicode conversion fails as ASCII based code pages may not have mapping character for NL/CRNL.

REMOTE_FILE

Indicates the file on the remote computer to which the transferred data is being written. If you are creating the file (FILE_OPTION=CREATE), the file name must be consistent with the file naming conventions of the remote system. The local CA XCOM Data Transport system does not validate this name. The remote I/O system determines whether the file name is valid.

For wildcard transfers, use an asterisk (*) as a file name to indicate and to inform the receiving partner that multiple files will be sent. For example, REMOTE_FILE=/PAYROLL/*.

If multiple files are sent and the user specifies a file name, all files received by the partner are written to that specified file as one single file.

An asterisk used to send files to an IBM mainframe system shows that all files are to be transferred to a partitioned data set (PDS).

For platforms that support it, you can specify a common file extension to append to each file name. For example, REMOTE_FILE=/PAYROLL/*.TXT.

Note: For send file transfers only.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_FILE_RF

Indicates the name of the file to be retrieved from the remote system.

For wildcard transfers, use an asterisk (*) as a file name to indicate that all files within the specified PDS/Directory should be transferred. For example, the statement REMOTE_FILE_RF=/NAMES/* indicates that all files under the NAMES directory should be transferred.

When a prefix is followed by an asterisk, all members beginning with a specific prefix are transferred. For example, REMOTE_FILE_RF=/NAMES/AL* requests that files AL, ALEX, and ALICE should be transferred. The same rules apply if an asterisk is followed by a suffix.

The actual file name range (not including its path) for wildcard transfers can be between 0 and 71 characters. This also includes the file extension where applicable. File names over 71 characters are truncated.

Note: For retrieve file transfers only.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM

The name of the remote system that receives a file, job, report, or ping request.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, the name is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote systems IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM_RF

The name of the remote system that sends a file on a receive file operation. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM_SJ

The name of the remote system to which a job is sent. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOTE_SYSTEM_SR

The name of the remote system to which a report is sent. If no value is specified here, the value in REMOTE_SYSTEM is used.

For SNA/APPC protocols, the name is as specified in the SNA/APPC configuration of the local system. For RS/6000, this is the LU6.2 Side Information record.

For TCP/IP protocols, the value can be the remote system's IP address, host name, or domain name.

Range: 1 to 256 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

REMOVE_TRAIL_BLANKS

Indicates whether or not to remove the blanks at the end of each record when receiving a text file from a remote partner.

Range: Yes or No

Default: No

REPORT_TITLE

This field contains the report title and job number that will be printed on the report. The field has the following format:

8 characters	2 characters	8 characters
Job Name	Blanks	Job Number

The job is optional and can be skipped. The job name can also be skipped, but if you skip the job name and want to use the job number, you must pad the number with 10 blanks.

Note: For report transfers only.

Examples

```
REPORT_TITLE="Salary94   Job12345"
REPORT_TITLE="           Job23456"
```

Non-example

```
REPORT_TITLE="      Job34567"
```

This is not a valid REPORT_TITLE because the job number spans both subfields.

This parameter is used by CA XCOM Data Transport on remote systems in the following ways:

System	Uses the REPORT_TITLE...
z/OS	To interpret a non-blank value in this field as specifying the generation of a separator (banner) page for this value.
VAX/VMS	To print with the report.
UNIX/Linux	To allow CA XCOM Data Transport to pass this field to the LP spooler as a title field.
Other systems	As a descriptive comment only and does not print it as part of the report.

Range: 0 to 21 alphanumeric or blank characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

RESTART_SUPPORTED

Specifies whether automatic restart is to be supported on a transfer.

Range: YES or NO

Default: YES

Note: This is a Version 2 parameter.

RETPD

Specifies the number of days (1 to 9999) that the tape data set being created is to be retained.

Range: 1 to 9999

Default: None

Note: RETPD and EXPDT are mutually exclusive; specify one or the other.

RETRY_TIME

The number of seconds between retries of unsuccessful transfers.

Range: 0 to 99999

Default: 1

Note: This is a Version 2 parameter.

RMTNTFYL

Specifies the remote user notification level when sending data to a remote system.

A (ALL)

Notify on transfer completion.

W (WARN)

Notify only if the transfer received a warning or error.

E (ERROR)

Notify only if the transfer received an error.

Default: ALL

RNOTIFY

Specifies the remote user notification type when sending data to a remote system.

MAIL

Mail is sent to the user

WRITE

A message is displayed on the local Windows system console.

ALL

A message is displayed on the local Windows system console.

TSO

Notify specified TSO user.

WTO

Write to log only (WTO).

ROSCOE

Notify the specified CA Roscoe user.

CICS

Notify CICS user (not used in Release 1).

LU

Notify Logical Unit (not used in Release 1).

NONE

No notification is sent.

Default: NONE

Notes:

- The R in NOTIFYR indicates that the remote system governs the processing of the resulting notification on that system.
- A MAIL message is sent to the specified user only if a MAPI-compliant provider is configured as a non-domain account and/or allows local accounts (such as the xcomd service and the local user ID) access to the Windows system (for example, Microsoft Mail boxes and address books on the local system).
- Some Microsoft Outlook 2003 security updates may prevent mail notification from working.

RNOTIFYNAME

Specifies the remote user notification type when sending data to a remote system.

MAIL

Mail is sent to the user

WRITE

A message is displayed on the local Windows system console.

ALL

A message is displayed on the local Windows system console.

TSO

Notify specified TSO user.

WTO

Write to log only (WTO).

ROSCOE

Notify the specified CA Roscoe user.

CICS

Notify CICS user (not used in Release 1).

LU

Notify Logical Unit (not used in Release 1).

NONE

No notification is sent.

Default: NONE

Notes:

- The R in NOTIFYR indicates that the remote system governs the processing of the resulting notification on that system.
- A MAIL message is sent to the specified user only if a MAPI-compliant provider is configured as a non-domain account and/or allows local accounts (such as the xcomd service and the local user ID) access to the Windows system (for example, Microsoft Mail boxes and address books on the local system).
- Some Microsoft Outlook 2003 security updates may prevent mail notification from working.

SAVE_PASSWORD_IN_CNF

Determines if the xcomtool GUI saves the password in the .CNF file.

Range: The range of values is listed in the following table:

YES

The xcomtool GUI saves the password in the .CNF file in encrypted format.

NO

The xcomtool GUI does not save the password in the .CNF file and blanks the password field out after each operation.

Default: NO

SECONDARY_ALLOC

This parameter identifies secondary storage allocation for a data set that was created on an IBM mainframe.

Range: 0 to 16,777,215

Default: 0

Note: This parameter is a Version 2 parameter.

SECURE_SCHEDULE

Specifies whether the meta-transfer uses SSL.

Y

The meta-transfer uses SSL.

N

The meta-transfer does not use SSL.

SECURE_SOCKET

Specifies whether to use an OpenSSL socket or non-OpenSSL socket for transfers.

YES

Performs a secure transfer. The transfer uses an OpenSSL socket and must to connect to a SSL listener on the remote partner.

NO

Performs a non-secure transfer. The transfer uses a non-OpenSSL socket.

Default: NO

SESSIONS_FILE

The path name of the xcom.ses file, which tells CA XCOM Data Transport how many sessions can be used by a single LU.

Range: 0 to 256 characters

Default: $XCOM_HOME/xcom.ses

Note: $XCOM_HOME is an environment variable.

SHELL_CMD

Name of the command that runs jobs, reports, notification scripts, and post processing scripts on the local system.

Range: 1 to 256 characters

Default: /bin/sh

SMTP_SERVER

This parameter specifies the name of the SMTP server.

Note: This is required only when MAIL_TYPE=SMTP.

Range: 1 to 64 characters

Default: None

SOCK_DELAY

TCP/IP socket option TCP_NODELAY. Refers to the Nagle algorithm for send coalescing. By default, small sends may be delayed. Should have no impact for normal CA XCOM Data Transport record sizes. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: The range of values is listed in the following table:

YES

Small sends may be delayed. (Does not disable the Nagle algorithm.)

NO

All sends are immediate. (Disables the Nagle algorithm.)

Default: YES

SOCK_RCV_BUF_SIZE

TCP/IP Socket option SO_RCVBUF. The buffer size for receives. Use zero for the default size provided by the socket implementation. The value for SOCK_RCV_BUF_SIZE can be smaller than the value for TXPI_BUF_SIZE. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: 0 to 65536

Default: 0

SOCK_SEND_BUF_SIZE

TCP/IP Socket option SO_SNDBUF. The buffer size for sends. Use zero for the default size provided by the socket implementation. The value for SOCK_SEND_BUF_SIZE can be smaller than the value for TXPI_BUF_SIZE. Used for TCP/IP transfers only.

Note: Socket options affect the way CA XCOM Data Transport uses the TCP/IP stack implementation.

Range: 0 to 65536

Default: 0

SPACE

The unit of storage allocation for the remote file.

CYL

Cylinders

TRK

Tracks

BLK

Blocks

REC

Records

Specify by:

- Primary allocation space for the remote file
- Secondary allocation for the remote file
- Directory blocks for the remote file

Default: CYL

SPOOL_FLAG

Indicates whether the report is to be spooled to disk or printed immediately. For report transfers only.

Note: If the remote system is an IBM mainframe, this parameter has no effect on the transfer.

Range: YES or NO

Default: YES

SPRTY

Indicates the scheduling priority of this meta-transfer request. When multiple file transfer requests are eligible for initiation (that is, they are past their start date/time) to the same LU or GROUP, those with higher SPRTY values are scheduled first. One is the lowest priority.

1 to 255

Specifies a scheduling priority for this file transfer request.

Note: This feature has no effect once the transfer begins executing. Do not confuse it with EPRTY, the execution priority.

Default: 16

START_DATE

Indicates the date on which the transfer becomes eligible for execution. The format is mm/dd/yy. If this field is blank, the current date is used.

Example:

A value of 02/28/13 indicates February 28, 2013 as the start date.

Format: *mm/dd/yy*

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

START_TIME

Indicates the time when the transfer becomes eligible for execution. The military format of *hh*:*mm*:*ss* is used. If this field is blank, then the current time is used.

Example:

A value of 14:00:00 indicates 2 p.m. as the start time.

Format: *hh*:*mm*:*ss*

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

STARTDATE

Specifies the date when the transfer is to begin.

The transfer start date can be provided in one of three formats:

- *yyyymmdd*

  yyyy

  Is a four-digit designation for a year (for example, 2013).

  mm

  Is a two-digit designation for one of the twelve months of the year as shown in the following chart:

```
01 = January      02 = February     03 = March
04 = April        05 = May          06 = June
07 = July         08 = August       09 = September
10 = October      11 = November     12 = December
```

  dd

  Is a two-digit number in the range 01 to 31 designating a day of the month.

  Example:

```
STARTDATE=20130201 schedules a transfer to begin on February 1, 2013.
```

- *yyddd* (Julian date)

  yy

  Is a two-digit designation for a year (for example, 13).

  ddd

  Is a three-digit number in the range 001 to 366 designating a day of the year.

  Example:

```
STARTDATE=13032 schedules a transfer to start on the 32nd day of 2013, which is
the same as February 1, 2013.
```

- +*nnn* (*nnn* days from today)

  nnn

  Is a number in the range 1 to 999.

  Example:

  Specifying STARTDATE=+31 on January 1, 2013 schedules a transfer to begin on February 1, 2013.

Default: Current date

STARTTIME

Specifies the time (*hhmm*) this transfer becomes eligible for execution.

0000 to 2400

Specifies the time (*hhmm*) this transfer becomes eligible for execution.

Notes:

- You can also set the time for the future in terms of its separation in hours and minutes (+*hhmm*) from the current time.
- For example, if a transmission is to start no earlier than 2 p.m.:

  STARTTIME=1400
- Or if the start time is two hours from now:

  STARTTIME=+0200.

Default: Current time

STAT_FREQUENCY

Indicates the frequency with which transfer statistics are made available to xcomqm. Intended for tuning high-speed links. Longer values help performance, but byte/record counts in xcomqm -D may be slightly behind the actual counts.

Range: 1 to 9999 records

Default: 10

STCIP

Specifies the IP address of the CA XCOM Data Transport server to handle the metatransfer.

Range: 1 to 64 characters

Default: None

STCPORT

Specifies the TCP/IP port number of the CA XCOM Data Transport server to handle the metatransfer request.

Range: 1 to 65536

Default: 8044

STCTRNENCRL_CIPHER

STCTRNENCRL_CIPHER specifies the requested list of ciphers which are used to encrypt the password fields for locally initiated -c5 meta-transfer request.

The cipher list consists of one or more ciphers from the table that colons separate. The characters (!) or (-) can precede each cipher in the cipher list.

If (!), is used then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-), is used then the ciphers are deleted from the list, but some or all of the ciphers are added again using later options.

Default: COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining
DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook

3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining
RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.

COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To request all ciphers except for any of the DES ciphers, use the following command:

```
STCTRNENCRL_CIPHER=ALL:!DES
```

To request only a 3DES or AES cipher, use the following command:

```
STCTRNENCRL_CIPHER=3DES:AES
```

To disable the cipher negotiation or remain backward compatible with the earlier releases of CA XCOM Data Transport, use the following command:

```
STCTRNENCRL_CIPHER=COMPAT
```

STORCLAS

Specifies the name of the storage class for a new SMS-managed data set.

Note: This parameter applies only to mainframe SMS data sets.

Range: One to eight characters

Default: None

SYSID

This parameter specifies the system ID (one to four characters).

This value is set during the installation process and is used for Trusted Transfers and for getting history records.

SYSID and SYSNAME together provide a unique system identifier.

Range: 1 to 4 characters

Default: None

SYSNAME

This parameter specifies the system name.

This value is set during the installation process and is used for Trusted Transfers and for getting history records.

SYSID and SYSNAME together provide a unique system identifier.

Range: 1 to 8 characters

Default: None

SYSTEM_USER_DATA

A user-defined text data field that can be used by an application as a customized system identifier. When specified in the configuration file of a locally initiated transfer, this identifying information is sent to the remote system and can be used to identify the system to certain applications. When received, the identifying information is displayed in the xcomqm detail display if the -D option is used, and is labeled System user data.

Range: 0 to 10 characters

Default: None

SYSUDATA

An open field where a user can specify any text associated with the transfer.

Note: This is equivalent to the USER_DATA field on Windows.

Range: 0 to 10 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

TAPE

Indicates to the remote system whether the transfer is to a tape volume or a disk file.

YES

The transfer is to a tape volume. Mounts are allowed when performing dynamic allocation.

NO

The transfer is to a disk file.

Default: NO

TCP_CMD

Path and name of the CA XCOM Data Transport program started by the XCOMD service that is used for queued locally initiated transfers, and for all remotely initiated transfers for TCP/IP protocols.

Range: 0 to 256

Default: $XCOM_HOME/bin/xcomtcp

Note: $XCOM_HOME is an environment variable.

TDUDATA

An open field where a user can specify any text associated with the transfer.

Note: This is equivalent to the XFERDATA field on UNIX and Linux.

Range: 0 to 10 characters

Default: None

TEMPDIR

This parameter indicates the directory in which temporary files for jobs and reports can be created.

Range: 1 to 256 characters

Default: $XCOM_HOME/tmp

Note: $XCOM_HOME is an environment variable.

TP_CMD

Command to start a transaction program using SNA protocols.

Range: 0 to 256 characters

Default: $XCOM_HOME/bin/xcomtp

TRACE_PATH

The directory containing the trace data files.

Range: 0 to 256 characters

Default: $XCOM_HOME/trace

Note: $XCOM_HOME is an environment variable.

TRANSFER_NAME

Allows a user to assign a name to a transfer. This name is also available to the remote CA XCOM Data Transport system. In addition, if the remote CA XCOM Data Transport system has defined a transfer identifier, that name is available to the local CA XCOM Data Transport for UNIX or Linux system. When received, the transfer identifier is displayed in the xcomqm display if the -D option is used, and is labeled Transfer name. The equivalent to this parameter on z/OS is the transfer identifier (XFERID).

Range: 0 to 10 characters

Default: None

TRANSFER_TYPE

Generated by the graphical user interface to specify the type of transfer to initiate.

Can also be used in a configuration file that has multiple transfers in it, to specify a different transfer type from the default value.

Range: 1 to 4

1

Send job

2

Send report

3

Send file

4

Receive file

Default: 3

Note: The functionality of this parameter is similar to using the **-c***number* command with xcom62 or xcomtcp at the command line. However, the values used to indicate the type of transfer are different.

TRANSFER_USER_DATA

A user-defined text data field that can be used by an application as a customized transfer identifier. When specified in the configuration file of a locally initiated transfer, this identifying information is sent to the remote system and can be used to identify the transfer to certain applications. When received, the identifying information is displayed in the xcomqm detail display if the -D option is used, and is labeled Transfer user data.

Range: 0 to 10 characters

Default: None

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

TRNENCRL_CIPHER

TRNENCRL_CIPHER specifies the requested list of ciphers which are used to encrypt the password fields for locally initiated transfers.

The cipher list consists of one or more ciphers from the table that colons separate. The characters (!) or (-) can precede each cipher in the cipher list.

If (!), is used then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-), is used then the ciphers are deleted from the list, but some or all of the ciphers are added again using later options.

Default: COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining
DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB

AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining
RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.
COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To request all ciphers except for any of the DES ciphers, use the following command:

```
TRNENCRL_CIPHER=ALL:!DES
```

To request only a 3DES or AES cipher, use the following command:

```
TRNENCRL_CIPHER=3DES:AES
```

To disable the cipher negotiation or remain backward compatible with the earlier releases of CA XCOM Data Transport, use the following command:

```
TRNENCRL_CIPHER=COMPAT
```

TRNENCRR_CIPHER

This parameter specifies the permitted list of ciphers that are used to encrypt the password fields for remotely initiated transfers. The permitted list of ciphers is matched against the requested list of ciphers provided by the local system by the TRNENCRL_CIPHER parameter. The common cipher with the highest ranking is selected to encrypt the password fields.

The cipher list consists of one or more ciphers that colons separate, with the highest ranked cipher listed first and the lowest ranked cipher listed last. The characters (!), (-) or (+) precedes each cipher in the cipher list.

If (!) is used, then the ciphers are permanently deleted from the list. The ciphers that are deleted can never reappear in the list even if they are explicitly stated.

If (-) is used, then the ciphers are deleted from the list. Some or all of the ciphers are added again using later options.

If (+) is used, then the ciphers are moved to the end of the list. This option does not add any new ciphers. The option just looks for existing ones to move to the end of the list.

Default: XCOM:ALL:COMPAT

Value	Comments
ALL	ALL ciphers AES:3DES:RC4:RC2:DES:XCOM **Note:** ALL does NOT include the COMPAT value.
DES	All DES ciphers DES-CBC:DES-ECB:DES-CFB:DES-OFB
DES-CBC	DES cipher with cipher-block chaining

DES-ECB	DES cipher with electronic codebook
DES-CFB	DES cipher with cipher feedback
DEC-OFB	DES cipher with output feedback
3DES	All 3DES ciphers 3DES-CBC:3DES-ECB:3DES-CFB:3DES-OFB
3DES-CBC	3DES cipher with cipher-block chaining
3DES-ECB	3DES cipher with electronic codebook
3DES-CFB	3DES cipher with cipher feedback
3DES-OFB	3DES cipher with output feedback
AES	All AES ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB:AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB:AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128	All AES 128-bit ciphers AES128-CBC:AES128-ECB:AES128-CFB:AES128-OFB
AES128-CBC	AES 128-bit cipher with cipher-block chaining
AES128-ECB	AES 128-bit cipher with electronic codebook
AES128-CFB	AES 128-bit cipher with cipher feedback
AES128-OFB	AES 128-bit cipher with output feedback
AES192	All AES 192-bit ciphers AES192-CBC:AES192-ECB:AES192-CFB:AES192-OFB
AES192-CBC	AES 192-bit cipher with cipher-block chaining
AES192-ECB	AES 192-bit cipher with electronic codebook
AES192-CFB	AES 192-bit cipher with cipher feedback
AES192-OFB	AES 192-bit cipher with output feedback
AES256	All AES 256-bit ciphers AES256-CBC:AES256-ECB:AES256-CFB:AES256-OFB
AES256-CBC	AES 256-bit cipher with cipher-block chaining
AES256-ECB	AES 256-bit cipher with electronic codebook
AES256-CFB	AES 256-bit cipher with cipher feedback
AES256-OFB	AES 256-bit cipher with output feedback
RC2	All RC2 ciphers
RC2-CBC	RC2 cipher with cipher-block chaining

RC2-ECB	RC2 cipher with electronic codebook
RC2-CFB	RC2 cipher with cipher feedback
RC2-OFB	RC2 cipher with output feedback
RC4	RC4 cipher
XCOM	CA XCOM Data Transport proprietary cipher.
COMPAT	Permit the CA XCOM Data Transport proprietary cipher without cipher negotiation. This is required for backward password compatibility with CA XCOM Data Transport versions before 11.6. **Note:** This cipher is also required for transfers sent to CA XCOM Data Transport for z/OS r11.6 and earlier versions.

Examples:

To permit all ciphers except for any of the DES ciphers and remain compatible with CA XCOM Data Transport versions before r11.6, use the following example:

```
TRNENCRR_CIPHER=ALL:!DES:COMPAT
```

To permit only a 3DES or AES cipher, use the following example:

```
TRNENCRR_CIPHER=3DES:AES
```

Note: To permit all ciphers for transfers from 11.6 but remain backward compatible with earlier versions of CA XCOM Data Transport, use the following example:

```
TRNENCRR_CIPHER=ALL:!XCOM:COMPAT
```

TRNENCRR_DHBITS

Specify the size (in bits) of the prime number that is used during DH (Diffie-Hellman) exchange for remotely initiated transfers or meta-transfers. The exchanged value generates the key that is used to encrypt the password fields transmitted in the CA XCOM Data Transport header.

Default: 1024

Value	Comments
256	DH exchange using a 256-bit prime number
512	DH exchange using a 512-bit prime number
1024	DH exchange using a 1024-bit prime number
2048	DH exchange using a 2048-bit prime number

4096	DH exchange using a 4096-bit prime number

Notes:

- The more bits used for the prime number the more secure the key exchange.
- The more bits used for the prime number the more CPU overhead is required to negotiate a secret value that is based on the prime number.
- Each CA XCOM Data Transport connection generates a unique secret value.
- No DH exchange is performed when the cipher the TRNENCRL_CIPHER/TRNENCRR_CIPHER negotiates is XCOM.

TRUNCATION

Indicates whether CA XCOM Data Transport truncates excess characters in the source file if the record exceeds the maximum record length as indicated by the MAXRECLEN parameter. If NO is selected, and the maximum record length is exceeded, CA XCOM Data Transport aborts the transfer. This parameter is ignored if CARRIAGE_FLAG=NO.

Note: Truncation is not valid for BINARY data or for non-text data received on the UNIX or Linux platform.

Range: YES or NO

Default: NO

TRUSTED

Allows the user to request a trusted transfer and the partner's CA XCOM Data Transport TRUSTED database to be searched to verify the user's credentials. This eliminates the need for the user to specify a USERID and PASSWORD. If XCOM_TRUSTED_OVR is set to NO or no USERID is specified, the USERID of the process that initiated the transfer is used.

Note: TRUSTED=YES cannot be specified with indirect transfers, because this is not supported.

Range: YES, NO, Y, N

Default: NO

TRUST_DATABASE_NAME

This parameter specifies the name of the database where the trusted tables were created.

Range: 1 to 256 characters

Default: None

TRUST_DATABASE_TYPE

This parameter specifies the database type that resides on the database server.

Range: DB2 or MySQL

Default: DB2

TRUST_ODBC

This parameter specifies the ODBC Data Source Name for the Trusted Tables.

Range: 1 to 16 characters

Default: xcomtrust

TRUST_OWNER

This parameter identifies the owner of the database tables. May be omitted if using MySQL or if it is the same as TRUST_USER when using DB2.

Range: 1 to 32 characters

Default: xcomtrust

TRUST_PASS

This parameter specifies the password of the user (TRUSTED_USER) for the Trusted Tables.

Note: When any changes are made via the Global Parameters tab, the updates encrypt this password and it is saved as TRUST_PASS.ENCRYPTED.

Range: 1 to 128 characters

Default: pass

TRUST_PORT

This parameter specifies the database server port where the trusted database resides.

Range: 1 to 65535

Default: 50000

TRUST_SERVER

This parameter specifies the IP address or name of the server where the trusted database resides.

Range: 1 to 256 characters

Default: None

TRUST_TABLE_PFX

This parameter specifies the prefix to use for the names of the Trusted Tables.

Range: 1 to 16 characters

Default: XCOM_TRUSTED

TRUST_USER

This is a generic user ID that has been defined to the RDMS for the Trusted Tables.

Range: 1 to 64 characters

Default: root

TXPI_BUF_SIZE

For TCP/IP transfers, the internal buffer size for sends and receives. The default size allows multiple CA XCOM Data Transport records to be received in a single socket call. With this default, if your CA XCOM Data Transport record size is less than 32K, CA XCOM Data Transport attempts to receive multiple records in a single socket call. Used for TCP/IP transfers only.

Range: 0 to 65536

Default: 32768

TXPI_RECEIVE_TIMEOUT

Maximum wait time, in seconds, that this CA XCOM Data Transport waits to receive from the partner system. If a value of 0 is specified, it waits indefinitely. Use for TCP/IP transfers only.

Range: 0 to 999 seconds

Default: 0

TXPI_SEND_CHECK_FREQ

Indicates the frequency with which CA XCOM Data Transport checks to see if incoming error information is available when sending data. For example, if the value is 5, a check is made every fifth time that data is sent, to determine if data is available for receiving. Larger values give better performance. Smaller values minimize the sending of data after the partner reports an error. Used for TCP/IP transfers only.

Range: 0 to 9999

Default: 10

TXPI_TERM_TIMEOUT

Maximum wait time, in seconds, for partner to terminate TCP/IP communications. If a transfer terminates normally, both sides of the conversation coordinate the termination, and there should be no need to wait. This timeout occurs only during an error in the termination of the connection. Used for TCP/IP transfers only.

Range: 0 to 999 seconds

Default: 20 seconds

UMASK

Used to set the permissions assigned to a file when the file is being created and received on the system for the first time. The value is expressed as an octal number (base 8). The octal number has the same meaning as in the standard umask command.

Range: 000 to 777

Default: 022

Note:

- For directories—CA XCOM Data Transport sets permissions for a created directory to 7xx, no matter what owner UMASK value was specified. Group and other permissions, represented by xx, represent the permissions with the specified UMASK removed.
- For files – While the file is being transferred, CA XCOM Data Transport sets permissions for a created file to 6xx, where xx represents the permissions with the specified UMASK removed. After the transfer has been completed, CA XCOM Data Transport sets the owner permission with the specified UMASK removed.

UNIT

Specifies the unit on which a data set is to be created on an IBM mainframe.

Note: Used when FILE_OPTION=CREATE.

Range: Zero to six characters

Default: None

UNIT_RF

Specifies the unit from which a data set is to be retrieved from an IBM mainframe.

Range: Zero to six characters

Default: None

UNITCT

Specifies the number of units to be allocated on the remote system.

This is a tape parameter and is used when the partner is an IBM mainframe.

Range: 1 to 20

Default: None

USE_TP_SECURITY

Only for NCR systems using SNA. When set to YES, CA XCOM Data Transport instructs the APPC to send out the user ID and password in the APPC attach request.

Note: This parameter applies to outgoing security only.

Range: YES or NO

Default: NO

USERID

The user ID that the security system on the remote system checks before granting access for the file transfer.

Range: 0 to 12 characters

Default: None

USEROVR

This parameter specifies whether the local user ID parameter LUSERID can be used when scheduling a transfer to a remote machine, using the xcomtcp -c5 option.

YES

Specifies that LUSERID can be used.

NO

Indicates that the local user ID (LUSERID) cannot be changed.

Default: YES

VERSION

Indicates the version of the CA XCOM Data Transport protocol to be used for this transfer. For TCP/IP, only a value of 2 is valid.

Range: 1 or 2

Default: 2

Note: This is a Version 2 parameter.

VOLCT

Specifies the maximum number of volumes to be used in processing a multi-volume output tape data set on the remote system.

Range: 1 to 255

Default: None

VOLSQ

Specifies the sequence number of the first volume of a multi-volume remote data set to be used.

Range: 1 to 255

Default: None

VOLUME

Specifies the volume on which a data set is to be created on an IBM mainframe.

Range: Zero to six characters

Default: None

VOLUME_RF

Specifies the volume from which a data set is to be retrieved from an IBM mainframe.

Range: Zero to six characters

Default: None

WRITER

Specifies the name of the external writer that is to process the report on the remote system.

xxxxxxxx

Specifies up to eight alphanumeric characters identifying the external writer that is to process this report on the remote system.

Note: This parameter cannot be specified with Version 1 transfers.

XBUFFSIZE

The size of CA XCOM Data Transport's internal buffer.

Important! Do not change this parameter unless you are instructed to do so by Technical Support.

Range: 1 to 32767

Default: 32767

XCOM_CONFIG_SSL

This parameter specifies the configssl.cnf file path and file name.

Range: 1 to 256 characters

Default: $XCOM_HOME/config/configssl.cnf

Note:

- $XCOM_HOME is an environment variable
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XCOM_ICUPATH

This parameter specifies the path to ICU shared libraries icudata and icuuc.

Range: 0 to 256 characters.

Default: $XCOM_HOME/bin

XCOM_JVM

This parameter specifies the full path and file name for the Java JVM shared library, libjvm.so.

Range: 1 to 256 characters

Default: /opt/CA/SharedComponents/JRE/1.6.0_32_ALL/lib/i386/client/libjvm.so.

XCOM_MYSQLODBC

This parameter specifies the MySQL ODBC driver. It is used for HP only.

Default: /usr/lib/libmyodbc5.sl

XCOM_ODBC

This parameter specifies the full path to the ODBC library (libodbc.so) on Unix.

Range: 0 to 256 characters

Default: None

XCOM_SHLIBPATH

This parameter specifies the path to ODBC driver to be added to SHLIB_PATH. It is used for HP only.

Default: None

XCOM_PASSWORD

This parameter specifies the default password for remotely initiated transfers.

Range: 0 to 31 characters

Default: None

XCOM_SHOW_CIPHER

Specifies whether to display encryption algorithms in the CA XCOM Data Transport queue detailed information, which is used for transfers.

NO

Do not display encryption algorithms in the queue detail information.

YES

Display encryption algorithms in the queue detail information.

Default: NO

XCOM_TRUSTED_OVR

Specifies if the user is permitted to override the user ID by using the USERID parameter for locally initiated trusted transfers. If XCOM_TRUSTED_OVR is set to YES, the user is permitted to override the user ID by using the USERID parameter. If it is set to NO, the user ID of the process that initiated the transfer is used.

Range: YES, NO, Y, N

Default: NO

XCOM_USERID

This is the default user ID for remotely initiated transfers.

Range: 0 to 12 characters

Default: xcom

XCOMPRE_LOCAL

Indicates whether the xcompre script file for a locally initiated transfer should be run.

Range: YES, NO

Default: NO

XENDCMD

The name of the post processing command optionally invoked by the CA XCOM Data Transport transfer program after any type of transfer is finished, whether successful or not. Invoked after partner communications have ended.

Range: 0 to 256 characters

Default: $XCOM_HOME/cmd/xcomend

Note: $XCOM_HOME is an environment variable.

XIDEST

Specifies the name of the remote system on the intermediate destination that is designated for store-and-forward transfers. If this variable is null or unset, then a direct connection to a remote system is attempted.

Note: For store-and-forward transfers only.

Range: 0 to 14 characters

Default: None

XLOGFILE

The name of the file where CA XCOM Data Transport logs activity. If you do not specify this parameter, the system-wide log file $XCOM_HOME/xcom.log is used. If you specify this parameter with a different file name, the logging information is only sent to the specified file.

Note:

- If QUEUE=YES, specify the full path n*a*me.
- $XCOM_HOME is an environment variable.

Range: 0 to 256 characters

Default: $XCOM_HOME/xcom.log

XLPCMD

The name of the post-processing command file used to send print jobs to the spooler. For incoming reports only.

Range: 0 to 256 characters

Default: $XCOM_HOME/cmd/xcomlp

Note: $XCOM_HOME is an environment variable.

XLUNAME

Indicates the name of the local LU. This name must match the remote LU name on the remote system.

Range: 1 to 17 characters

Default: None

Note: Not used in CA XCOM Data Transport for RS/6000 SNA Services or RS/6000 SNA Server.

XMODE

Indicates the mode name associated with the SNA/APPC configuration for the XLUNAME-Remote LU name pair. This name must match the mode name defined on the remote system.

Range: One to eight characters

Default: XCOMMODE

Note: Not used in CA XCOM Data Transport for RS/6000 SNA Services or RS/6000 SNA Server.

XNODESPEC

Required for Brixton and SunLink APPCs. Indicates the name of the node that specifies the gateway to use for the transfer. Does not apply to other APPCs.

Range: 1 to 64 characters

Default: None

XNOTIFYCMD

The name of the command file that CA XCOM Data Transport will use to notify users on the local system of the completion of a transfer. This is normally a shell script that composes a message and invokes mail or write as appropriate.

Range: 0 to 256 characters

Default: $XCOM_HOME/cmd/xcomntfy

XPPCMD

The name of the command file used for user-defined post processing, for file transfers only. Only used when the local system is receiving the file.

Range: 0 to 64 characters

Default: $XCOM_HOME/cmd/xcompp

Note: $XCOM_HOME is an environment variable.

XPRECMD

The name of the command file used for user-defined pre-allocation processing for locally and remotely initiated transfers. Specify this parameter in xcom.glb to invoke the pre-allocation exit xcompre. A sample command file is provided in $XCOM_HOME/cmd/xcompre.bat.

Range: 0 to 64 characters

Default: $XCOM_HOME/cmd/xcompre.bat

Note: $XCOM_HOME is an environment variable.

XTCERRDECR

Specifies the transfer requests for which the HOLDCOUNT parameter value is decremented when the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if a transfer completes unsuccessfully.

Note:

- VTAM errors are not considered failed file transfers, because they are retried later.
- If the current file transfer fails, the value of the HOLDCOUNT parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the HOLDCOUNT value has already reached zero.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRINCR

Specifies the transfer requests for which the HOLDCOUNT parameter value is incremented if the current file transfer fails.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if the current file transfer fails.

Notes:

- A VTAM communications error does not count as a failure.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRPURGE

Specifies the transfer requests to be purged if the transfer concludes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be purged if the current file transfer concludes unsuccessfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCERRREL

Specifies the transfers to be released if the current transfer completes unsuccessfully.

Up to eight transfer request names

Specifies up to eight transfers to be released if the current transfer completes unsuccessfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODDECR

Indicates an XTCNET job whose hold count decrements if the file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be decremented if the current transfer completes successfully.

Notes:

- If the current file transfer completes successfully, the value of the HOLDCOUNT parameter of the transfers assigned to this parameter is reduced by one, unless the next transfer has already started or the HOLDCOUNT value has already reached zero.
- You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODINCR

Indicates the transfer requests whose HOLDCOUNT parameter is incremented when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight transfers whose hold counts are to be incremented if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODPURGE

Specifies the transfer requests to be purged when the current file transfer completes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be purged if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCGOODREL

Specifies the transfer requests to be released if the current file transfer concludes successfully.

Up to eight transfer request names

Specifies up to eight jobs to be released if a transfer completes successfully.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCJOB

Defines the name of a transfer request belonging to the group of interdependent transfer requests named through the XTCNET parameter.

xxxxxxxx

Specifies the name of a transfer request in a group of interrelated transfer requests.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTCNET

Defines the name of a group of interdependent transfer requests.

xxxxxxxx

Specifies up to eight alphanumeric characters representing the name of the XTC network running this transfer.

Note: You can use symbolic variables with this parameter. For more information, see Symbolic Parameters in Configuration Files.

XTRACE

Indicates the level of desired execution tracing.

0

No tracing.

1 to 8

Levels of tracing from 1 (lowest) to 8 (highest).

9

Show the contents of the data buffers.

10

Includes levels 1 to 9 with additional detailed technical information.

Default: 0

Note: Turning on the trace can seriously degrade performance.

Appendix B: Messages

The messages contained in this section are written to a specified log file on the local system (see the XLOGFILE parameter). Some messages are also displayed on the user's terminal and sent to the remote system. The messages are listed in ascending order.

This section contains the following topics:

Error Messages (see page 483)
List of Messages (see page 485)

Error Messages

Messages with numbers ranging from 0 to 255 are informational or prompt messages. Messages numbering between 100 and 255 represent CA XCOM Data Transport states and are used in traces, and so on. Message numbers of 256 and above are error messages. The following is an example of a CA XCOM Data Transport message:

```
XCOMU0010I XCOM62 STARTING TRANSFER
```

Message ID Content

The parts of a CA XCOM Data Transport message ID are as follows:

XCOM (positions 1 to 4; alphabetic)

Displays the first four characters of a CA XCOM Data Transport message.

system identifier **(position 5; alphabetic)**

Specifies the CA XCOM Data Transport system that generated the message. Valid values include the following:

- C—Microsoft PC-DOS and MS-DOS, and IBM OS/2
- D—Digital Equipment Corp. VAX/VMS, OpenVMS Alpha, and OpenVMS VAX
- E—IBM VSE
- G—Data General AOS/VS
- K—IBM CICS
- M—IBM z/OS
- N—Windows NT
- R—Netware
- S—IBM AS/400
- T—Tandem
- U—UNIX
- V—IBM VM
- W—Windows
- 8—Stratus Computer and IBM System/88

Note: Some of the system identifiers in this list represent systems that are supported for existing customers but are no longer generally available. For a list of currently available CA XCOM Data Transport platforms, see the *CA XCOM Data Transport for UNIX and Linux Overview Guide*.

message no **(positions 6 to 9; numeric)**

Specifies the message number.

message type **(position 10; alphabetic)**

Indicates the message type, as follows:

- I—An informational message. No action is required on the part of the user.
- E—An error message. Usually some action is necessary to correct the problem or to determine the cause.
- P—A prompt requesting a response by the user.
- W—A warning message that some type of failure is occurring which may or may not be fatal to what you are doing.

List of Messages

The following pages list messages that CA XCOM Data Transport can return in response to job requests, with explanations and recommended user actions, where appropriate. In the list, the four-character constant (XCOM) and the component identifier (in our case, U for UNIX or Linux), have been deleted from each message ID. The messages are listed in numerical order.

0007I

Major code: 0x00000000 Descriptive message

Reason:

This is a return code from the underlying system software. It is usually followed by a message describing the meaning of the code.

Action:

Refer to the documentation for the SNA LU 6.2 or other system software.

0008I

Minor code: 0x00000000 Descriptive message

Reason:

This is a return code from the underlying SNA APPC.

Action:

Refer to the documentation for the SNA LU 6.2 software.

0010I

Starting XCOM Transfer on Thu Aug 8 12:19:18 2002

Reason:

This start of transfer message is issued when a locally initiated transfer is begun. It is simultaneously placed in the transfer log.

Action:

None required.

0011I

Transfer ended; 0 blocks (0 bytes) transmitted in 0 seconds (0 bytes/second)

Reason:

This message marks the completion of a locally initiated transfer. It placed in the log and written to standard error at the end of a transfer.

Action:

None required.

0016E

Transfer started by a remote system.

Reason:

This message is issued when a transfer is started by a remote system by allocating the CA XCOM Data Transport transaction program.

Action:

None required.

0017I

Transfer expired and was removed from queue.

Reason:

The transfer has expired and has been removed from the queue.

Action:

None required.

0018I

Transfer ended; File successfully imported from (or) exported to external server

Reason:

This message marks the completion of Locally Initiated Gateway Transfer which will Import File from (or) Export File to External Server Registered with Gateway.

Action:

None

0022I

Transfer suspended by user

Reason:

The transfer has been suspended by the user. This is an informational message that is sent to the remote side of the conversation to inform the remote system and user that the transfer has been suspended.

Action:

None required.

0023I

Transfer resumed by RESUME command.

Reason:

The transfer has been resumed by a RESUME command issued from the screen interface or by the use of the XCOMQM command.

Action:

None required.

0024I

Transfer scheduled for future execution.

Reason:

The transfer is placed in the CA XCOM Data Transport scheduler's queue.

Action:

None required.

0025I

Remotely initiated retry.

Reason:

The retry was initiated by the remote system.

Action:

None required.

0026I

Remotely initiated first try.

Reason:

The transfer was initiated by the remote system.

Action:

None required.

0027I

Remotely initiated restart.

Reason:

The restart was initiated by the remote system.

Action:

None required.

0028I

Starting locally initiated transfer.

Reason:

The locally initiated transfer is starting.

Action:

None required.

0029I

Locally initiated transfer started.

Reason:

The locally initiated transfer has started.

Action:

None required.

0030E

Cannot process a non-queued TCPIP transfer from xcom62.

Reason:

A transfer using TCP/IP was attempted using the xcom62 command.

Action:

Use the xcomtcp command for transfers using TCP/IP protocols.

0031E

Cannot process a non-queued SNA/APPC transfer from xcomtcp.

Reason:

A transfer using SNA/APPC was attempted using the xcomtcp command.

Action:

Make sure xcomd is running. Use the xcom62 command for transfers using SNA/APPC protocols.

0032E

Version 1 transfers not permitted with TCPIP.

Reason:

PROTOCOL=TCPIP was specified with VERSION=1.

Action:

Use VERSION=2 with PROTOCOL=TCPIP or VERSION=1 with PROTOCOL=SNA.

0034I

Starting Gateway FTP export

Reason:

Locally Initiated Gateway Transfer started to Export File from Gateway Environment to External FTP server which is registered with Gateway.

Action:

None

0035I

Starting Gateway FTP import

Reason:

Locally Initiated Gateway Transfer started to Import File from External FTP server which is registered with Gateway to Gateway Environment.

Action:

None

0036I

Starting Gateway SFTP export

Reason:

Locally Initiated Gateway Transfer started to Export File from Gateway Environment to External SFTP server which is registered with Gateway.

Action:

None

0037I

Starting Gateway SFTP import

Reason:

Locally Initiated Gateway Transfer started to Import File from External FTP server which is registered with Gateway to Gateway Environment.

Action:

None

0038I

Starting Gateway FTPS export

Reason:

Locally Initiated Gateway Transfer started to Export File from Gateway Environment to External FTPS server which is registered with Gateway.

Action:

None

0039I

Starting Gateway FTPS import

Reason:

Locally Initiated Gateway Transfer started to Import File from External FTPS server which is registered with Gateway to Gateway Environment.

Action:

None

0046I

CONFIRMED_CHECKPOINT

Reason:

CA XCOM Data Transport is confirming a checkpoint.

Action:

None provided.

0047I

Local transfer restarted.

Reason:

The local transfer has restarted.

Action:

None required. This message is used to report the state of the CA XCOM Data Transport transaction program.

0052I

Transfer held.

Reason:

The transfer was held from starting at the time for which it was scheduled.

Action:

None required. This message is used to report the state of the CA XCOM Data Transport transaction program.

0053I

Transfer released.

Reason:

The transfer was released to start at the time for which it was scheduled.

Action:

None required.

0075P

Warning: xcomd will abort. Data may be lost. Go ahead?

Reason:

This is a warning message allowing you one last chance before forcefully bringing down the CA XCOM Data Transport scheduler.

Action:

Respond with y if you want to stop the daemon. Any other reply will not stop the process.

0077I

Aborting xcomd; PID=0

Reason:

CA XCOM Data Transport scheduler is being forced down.

Action:

None required. The daemon will stop.

0078P

Please answer y(yes) or n(no).

Reason:

Please answer y (yes) or n (no).

Action:

Reply to the prompt with a *y* or an *n*.

0079I

xcomd: stop requested.

Reason:

A user has requested that the CA XCOM Data Transport scheduler to stop gracefully.

Action:

None required.

0080I

xcomd: stop request received.

Reason:

A user has requested that the CA XCOM Data Transport scheduler, xcomd, stop, and that request has been received by the running daemon process.

Action:

None required.

0081I

xcomd: stop request rejected.

Reason:

The CA XCOM Data Transport scheduler has rejected a user's request to stop. CA XCOM Data Transport scheduler cannot be stopped while there are started or active transfers in the queue.

Action:

Suspend all active transfers and retry the operation.

0082I

xcomd: stop request accepted.

Reason:

The CA XCOM Data Transport scheduler has accepted a user's request to stop.

Action:

None required.

0085P

Configuration cannot be read correctly. Defaults will be taken.

Reason:

Some parameters cannot be taken from the configuration file. This may be caused by the following reasons:

- The configuration file contains errors or invalid parameters.
- The read() system call returned an error.

Action:

Reply to the prompt. If you do not want defaults to be taken for parameters that cannot be parsed, reply 'cancel' and check your configuration file. Otherwise, reply 'ok'.

0086P

Configuration cannot be read correctly. No action will be taken.

Reason:

Some parameters cannot be taken from the configuration file. This may be caused by the following reasons:

- The configuration file contains errors.
- The read() system call returned an error.

Action:

Reply 'ok'. If you do not want defaults to be taken for parameters that cannot be parsed, check your configuration file and try to open it again.

0087P

Do you want to save changes?

Reason:

This warning message appears when you are about to lose some changes you made to CA XCOM Data Transport parameters to give you the last chance to save the changes in file.

Action:

Reply to the prompt.

0088I

xcomd started.

Reason:

The CA XCOM Data Transport scheduler has started.

Action:

None required.

0089I

xcomd ended.

Reason:

The CA XCOM Data Transport scheduler has ended.

Action:

None required.

0090I

Problems starting xwait, Retrying...

Reason:

The CA XCOM Data Transport wait process is unable to start.

Action:

None required.

0091I

xcomxped started.

Reason:

The CA XCOM Data Transport XPE interface has started.

Action:

None required

0092I

xwait started.

Reason:

The CA XCOM Data Transport wait process has started.

Action:

None required.

0093I

Transfer held by XTC.

Reason:

This message is issued if HOLD_TRANSFER=YES or HOLDCOUNT > 0 is set with XTCNET.

Action:

None

0094I

XTC Command processing.

Reason:

This message is issued if XTC processing starts for a transfer.

Action:

None

0100I

XEXIT

Reason:

The CA XCOM Data Transport transaction program is about to exit.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0102I

CLEANUP

Reason:

The state machine is about to enter CLEANUP state. In this state, the LU 6.2 conversation is deallocated.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0103I

TP_VALID

Reason:

The state machine is about to enter TP_VALID state. In this state, CA XCOM Data Transport accepts incoming remote allocates if the GET_ALLOCATE verb has been issued.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0105I

ALLOCATE_CONVERSATION

Reason:

The state machine is about to enter ALLOCATE_CONVERSATION state. It is about to issue the allocate verb for the remote transaction program.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0106I

GET_ALLOCATE

Reason:

The state machine is about to enter GET_ALLOCATE state. In this state, incoming allocates are sought by the local transaction program.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0107I

DEALLOCATE_CONVERSATION

Reason:

The state machine is about to enter DEALLOCATE_CONVERSATION state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0108I

SEND_HEADER

Reason:

The state machine is about to enter SEND_HEADER state. In this state, a buffer is allocated, the CA XCOM Data Transport header record is created from the transfer parameters, and the header is sent.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0109I

REQUEST_HEADER_CONFIRMATION

Reason:

The state machine is about to enter REQUEST_HEADER_CONFIRMATION state where the incoming header is confirmed, and, if necessary, the connection is turned around.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0111I

SEND_MAXLRECL

Reason:

The state machine is about to enter SEND_MAXLRECL state. Here, the state machine will send the maximum logical record length as specified in the configuration file.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0112I

SENDING_DATA

Reason:

The state machine is about to enter SENDING_DATA state. CA XCOM Data Transport sends one data record each time it enters this state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0113I

DATA_CONFIRM

Reason:

The state machine is about to enter DATA_CONFIRM state. In this state, the transaction program issues the confirm that is sent at the end of a data file.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0114I

SEND_TRAILER

Reason:

The state machine is about to enter SEND_TRAILER state. The trailer record containing the number of records is sent.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0115I

TRAILER_CONFIRM

Reason:

The state machine is about to enter TRAILER_CONFIRM state. The CA XCOM Data Transport transaction program is about to issue a deallocate confirm verb for the conversation.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0116I

RECEIVE_HEADER

Reason:

The state machine is about to enter RECEIVE_HEADER state. In this state, CA XCOM Data Transport receives the incoming header.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0117I

CONFIRM_HEADER

Reason:

The state machine is about to enter CONFIRM_HEADER state. The CA XCOM Data Transport transaction program is about to issue the confirmed LU 6.2 verb.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0118I

RECEIVE_MAXLRECL

Reason:

The state machine is about to enter RECEIVE_MAXLRECL state. The transaction program receives the maximum record length record and uses it to initialize system parameters.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0119I

RECEIVE_DATA

Reason:

The state machine is about to enter RECEIVE_DATA state. In this state, the data records are received, decompressed, unpacked, and written to the target file.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0120I

DATA_CONFIRMED

Reason:

The state machine is about to enter DATA_CONFIRMED state. It issues the confirmed verb and closes the received file in preparation for receiving the trailer record.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0121I

RECEIVE_TRAILER

Reason:

The state machine is about to enter RECEIVE_TRAILER state. A receive_and_wait verb is issued for the incoming CA XCOM Data Transport trailer record.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0122I

TRAILER_CONFIRMED

Reason:

The state machine is about to enter TRAILER_CONFIRMED state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0123I

PROCESS_DATA

Reason:

The state machine is about to enter PROCESS_DATA state. In this state, the transaction program compares the record count actually received to the count sent in the trailer record. If they don't match, the state machine enters an invalid trailer state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0125I

TERMINATE_INITIATE_CHECK

Reason:

The state machine is about to enter TERMINATE_INITIATE_CHECK state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0127I

TP_DONE

Reason:

The state machine is about to enter TP_DONE state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0128I

RECEIVE_FEATURE_RECORD

Reason:

The state machine is about to enter RECEIVE_FEATURE_RECORD state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0129I

SEND_FILE_DESCRIPTOR

Reason:

The state machine is about to enter SEND_FILE_DESCRIPTOR state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0130I

REQUEST_FEATURE_RECORD_CONFIRM

Reason:

The state machine is about to enter REQUEST_FEATURE_RECORD_CONFIRM state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0131I

CONFIRMED_FEATURE_RECORD

Reason:

The state machine is about to enter CONFIRMED_FEATURE_RECORD state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0152I

LOCAL_SEND

Reason:

The state machine is about to enter LOCAL_SEND state. In this state, the transaction program initializes several internal variables, logs the startup message, and sets the userid for this transfer.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0153I

LOCAL_RECEIVE

Reason:

The state machine is about to enter LOCAL_RECEIVE state. In this state, the transaction program initializes several internal variables, logs the startup message, and sets the userid for this transfer.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0154I

REMOTE_SEND

Reason:

The state machine is about to enter REMOTE_SEND state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0155I

REMOTE_RECEIVE

Reason:

The state machine is about to enter REMOTE_RECEIVE state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0156I

OPEN_REMOTE_INPUT_FILE

Reason:

The state machine is about to enter OPEN_REMOTE_INPUT_FILE state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0157I

OPEN_LOCAL_INPUT_FILE

Reason:

The state machine is about to enter OPEN_LOCAL_INPUT_FILE state. The local input file is about to be opened.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0158I

OPEN_OUTPUT_FILE

Reason:

The state machine is about to enter OPEN_OUTPUT_FILE state. The output file is about to be opened.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0159I

LOCAL_ATTACH

Reason:

The state machine is about to enter LOCAL_ATTACH state. In this state, the transaction program tries to establish a connection to the local LU 6.2 SNA server.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0161I

REMOTE_ATTACH

Reason:

The state machine is about to enter REMOTE_ATTACH state. In this state, the invoked transaction program tries to establish a connection to the local LU 6.2 SNA server.

Action:

This message is used to report the state of the CA XCOM Data Transport transaction program.

0162I

SET_UP_OVERLAY

Reason:

The state machine is about to enter SET_UP_OVERLAY state. In this state, the indirect transfer header record is created.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0163I

SEND_OVERLAY

Reason:

The state machine is about to enter SEND_OVERLAY state. In this state, it sends the overlay record that is part of the indirect transfer protocol.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0164I

RECEIVE_OVERLAY

Reason:

The state machine is about to enter RECEIVE_OVERLAY state. An indirect transfer has been requested by the remote side and the overlay record is about to be received.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0165I

DO_SYSTEM

Reason:

The state machine is about to enter DO_SYSTEM state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0167I

SET_REMOTE_USER_ID

Reason:

The state machine is about to enter SET_REMOTE_USER_ID state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0168I

DO_COMMAND

Reason:

The state machine is about to enter DO_COMMAND state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0169I

SEND_FEATURE_RECORD

Reason:

The state machine is about to enter SEND_FEATURE_RECORD state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0170I

RECEIVE_FILE_DESCRIPTOR

Reason:

The state machine is about to enter RECEIVE_FILE_DESCRIPTOR state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0171I

RESTART_TRANSFER

Reason:

The state machine is about to enter RESTART_TRANSFER state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0172I

CHECKPOINT_CONFIRM_REQUEST

Reason:

The state machine is about to enter CHECKPOINT_CONFIRM_REQUEST state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0173E

rc=173 sense=xxxx

Reason:

The state machine is about to enter OPEN_COMPRESS state. rc=173 indicates OPEN_COMPRESS and is not an error.

Action:

If sense=0, no action is required. If sense=>0, take a CA XCOM Data Transport trace and contact CA Technical Support.

0180I

SENDING_ERROR

Reason:

The state machine is about to enter SENDING_ERROR state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0182I

RECEIVE_ERROR

Reason:

An error message is being received from the remote partner. The CA XCOM Data Transport transaction program issues a receive and wait verb to retrieve the text.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0185I

SEND_ERROR_MESSAGE

Reason:

The state machine is about to enter SEND_ERROR_MESSAGE state. A send error verb has been issued and the text of the error encountered is sent.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0186I

WAIT_FOR_REMOTE_DEALLOCATE

Reason:

The state machine is about to enter WAIT_FOR_REMOTE_DEALLOCATE state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0187E

TRANSFER_FAILED

Reason:

The CA XCOM Data Transport transfer has failed.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0188I

CHECK_RETURN_CODES

Reason:

CA XCOM Data Transport is checking return codes at the end of a transfer.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0200I

LOCAL_NOTE

Reason:

The state machine is about to enter LOCAL_NOTE state. In this state, the transaction program initializes several internal variables, logs the startup message, and sets the userid for this transfer.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0201I

REMOTE_NOTE

Reason:

The state machine is about to enter REMOTE_NOTE state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0202I

LOCAL_NOTE_ATTACH

Reason:

The state machine is about to enter LOCAL_NOTE_ATTACH state. In this state, the transaction program tries to establish a connection to the local LU 6.2 SNA server.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0203I

ALLOCATE_NOTE_CONVERSATION

Reason:

The state machine is about to enter ALLOCATE_NOTE_CONVERSATION state. It is about to issue the allocate verb for the remote transaction program.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0204I

SENDING_NOTE_DATA

Reason:

The state machine is about to enter SENDING_NOTE_DATA state. CA XCOM Data Transport sends one sdsnote when it enters this state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0205I

DATA_NOTE_CONFIRM

Reason:

The state machine is about to enter DATA_NOTE_CONFIRM state. In this state, the transaction program issues the confirm that is sent at the end of a sdsnote.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0207I

LOCAL_PING

Reason:

The state machine is about to enter LOCAL_PING state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0211I

queue entry (TID=000000) removed.

Reason:

Queue entry was removed from the queue.

Action:

None. This message is used to report the status of the transfer.

0212I

transfer (TID=000000) suspended.

Reason:

Transfer was suspended.

Action:

None. This message is used to report the status of the transfer.

0213I

transfer (TID=000000) terminated.

Reason:

Transfer was terminated.

Action:

None. This message is used to report the status of the transfer.

0214I

transfer (TID=000000) resumed.

Reason:

Transfer was resumed.

Action:

None. This message is used to report the status of the transfer.

0215I

transfer (TID=000000) held.

Reason:

Transfer was held.

Action:

None. This message is used to report the status of the transfer.

0216I

transfer (TID=000000) released.

Reason:

Transfer was released.

Action:

None. This message is used to report the status of the transfer.

0217I

CHECK_XCOMLIB

Reason:

The state machine is about to enter CHECK_XCOMLIB state. In this state, the transfer file name is checked for wild cards. If a wild card is found, the transfer is marked as a DIRTREE type transfer.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0218I

GATEWAY_SEND

Reason:

The state machine is about to enter GATEWAY_SEND state. In this state, the state machine determines if the file being sent is a CA XCOM Gateway file. This state also initiates the connection to the CA XCOM Gateway to extract a file.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0219I

GATEWAY_RECEIVE

Reason:

The state machine is about to enter GATEWAY_RECEIVE state. In this state, the state machine determines if the file being received is a CA XCOM Gateway file. This state also initiates the connection to the CA XCOM Gateway to insert a file.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0224I

REMOTE_DISCOVER_REQUEST

Reason:

The state machine is about to enter REMOTE_DISCOVER_REQUEST state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0225I

ALLOCATE_DISCOVERY_CONVERSATION

Reason:

The state machine is about to enter ALLOCATE_DISCOVERY_CONVERSATION state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0226I

LOCAL_PING_ATTACH

Reason:

The state machine is about to enter LOCAL_PING_ATTACH state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0227I

UNICODE_INIT

Reason:

The state machine is about to enter the UNICODE_INIT state.

Action:

None. This message is used to report the state of the CA XCOM Data Transport transaction program.

0283E

Error allocating send buffer

Reason:

An error was encountered when trying to increase the send buffer size using realloc call.

Action:

Retry the transfer. This error may be caused by memory fragmentation.

0284E

Error forking: Error 0

Reason:

Cannot create an additional process. Either the user or system process limit has been reached.

Action:

Check your system and user process limits. Your system administrator may be able to supply you with this information and/or raise these limits.

0285E

Error creating pipe

Reason:

An error occurred in creating a pipe. File descriptors are not available to open a UNIX or Linux pipe. You may have exceeded the limit on the number of open files you are allowed.

Action:

Contact your system administrator.

0286E

Error setting local user id

Reason:

An error occurred in setting the local user ID. The call to setuid failed.

Action:

Be sure that the local CA XCOM Data Transport transaction program is either owned by the invoking user or is running setuid root. Use fully qualified file names.

0287E

Error setting remote user id

Reason:

An error was encountered while trying to log on to the partner.

Action:

Check that the remotely requested user ID is valid on the partner system and that the initiator has entered it and the password for it correctly. Check the case for the password.

Error setting remote user id: PAM library load failure.

Reason:

The XCOM application is unable to load the library from the location specified in the PAM Path field.

Action:

Verify the correctness of PAM Path entered. Check if the location has the PAM load libraries.

0288E

System function failed (Wait Timed Out).

Reason:

A call to system( ) has failed. This function is used to initiate command functions from within CA XCOM Data Transport.

If "wait timed out" is specified after the error message text, this indicates that the transferred job did not complete before the DAEMON_TIMEOUT period expired.

Action:

Retry the transfer. This message may indicate the failure of a system call in a sendjob or in a pre or post processing script. If the problem persists, take a trace and contact CA Technical Support.

0288E

System function failed

Reason:

A call to system() has failed. This function is used to initiate command functions from within CA XCOM Data Transport.

Action:

The transfer was successful but the send job or a script failed. Retry the transfer. If the problem persists, take a CA XCOM Data Transport trace and contact CA Technical Support.

0289E

Command failed, RC=n

Reason:

A command issued using the system() call has returned a non-zero completion code. A send job failed when the job was submitted on the remote partner. The transfer to send the job was successful, however.

Action:

Retry the transfer. Put debugging statements in the job. See the sample post-processing scripts for debugging statements. If the problem persists, contact CA Technical Support.

0290E

Error sending overlay

Reason:

Send of indirect transfer record failed.

Action:

Retry the transfer. This may be due to a network problem. If the error persists, contact your network administrator or CA Technical Support.

0291E

Send of indirect transfer record failed: Descriptive message

Reason:

An error has occurred while attempting to send the overlay record for an indirect file transfer.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0292E

Error sending error message to remote system: Descriptive message

Reason:

An error was encountered while sending an error message to the remote system.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0296E

Error deallocating conversation: Descriptive message

Reason:

An error was detected while deallocating a send conversation. This deallocate occurred after sending a file.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0297E

Error requesting header confirmation: Descriptive message

Reason:

The remote CA XCOM Data Transport transaction program did not confirm the transfer request header that it received.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0298E

Unable to allocate remote transaction program: Descriptive message

Reason:

The APPC interface returned an error after an attempt was made to allocate a CA XCOM Data Transport transaction program on a remote LU. For transfers using TCP/IP protocols, refer to the common return codes and explanations listed in the system errno.h file.

Action:

Check that the remote transaction program can be allocated. On z/OS and OS/390, this means that the APPL is active. On other systems, there are different criteria for accessibility.

Check that the listener is running on the remote system for TCP/IP transfers and is listening on the specified port.

0299E

Cannot attach to LU 6.2 facility on local initiate: Descriptive message

Reason:

An error was detected when a locally initiated transfer request attempted to attach to the APPC manager process.

Action:

This is frequently an SNA configuration problem. Check the SNA status as well as your connection status. Check the REMOTE_SYSTEM parameter to make sure that it contains the name of the connection profile that you want to use for this transfer.

0300E

Cannot attach to LU 6.2 facility on remote allocate: Descriptive message

Reason:

An error was detected when a remotely initiated transfer request transaction program attempted to attach to the APPC manager process.

Action:

Check the SNA configuration and retry the transfer. If the problem persists, contact CA Technical Support.

0301E

Unable to start APPC Communications

Reason:

Error starting SNA Services.

Action:

Check for problems starting your SNA APPC.

0302E

Unable to open local input file: Descriptive message

Reason:

A problem was encountered while trying to open a CA XCOM Data Transport input file; the data following the explanation is the reason.

Action:

Check the existence and permissions on the file you have specified.

0303E

Send of XCOM header failed: Descriptive message

Reason:

An error has occurred while attempting to send a transfer header record.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0304E

Send of Maximum Record Length failed: Descriptive message

Reason:

An error has occurred while attempting to send the maximum logical record length as part of the file transfer setup procedure.

Action:

Check the MAXRECLEN parameter and retry. If the problem persists, contact CA Technical Support.

0305E

Receive of incoming header failed: Descriptive message

Reason:

An error was detected while attempting to receive a transfer request header from a remote system.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0309E

Error reading local input file: Descriptive Message

Reason:

An error occurred while reading the input file. The system call read returned -1. The error number displayed is the value of the system variable error.

Action:

Check the descriptive message, fix the problem, and retry the transfer. If the problem persists, contact CA Technical Support.

0310I

Received error from remote system.

Reason:

An error has been received from the remote system and has been placed in the log.

Action:

Check the error received from the remote system and fix the problem detailed there. Error messages from partner CA XCOM Data Transport systems can be found in the manuals for those platforms.

0311E

Send of user data record failed: Descriptive message

Reason:

An error was returned from the APPC interface while attempting to send data to a remote system.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0312E

Confirm to last data record failed: Descriptive message

Reason:

The APPC interface has returned an error when attempting to issue a confirm following the last data record in a transfer.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0313E

Negative response to data confirm request: Descriptive message

Reason:

Negative response to data confirm request.

Action:

Check that the target file is not open or locked. Contact CA Technical Support.

0314E

Send of trailer record failed: Descriptive message

Reason:

The send of trailer record failed. An error was returned by the SNA subsystem after the CA XCOM Data Transport trailer was sent.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0315E

Trailer record not confirmed

Reason:

The trailer record was not confirmed. The remote side has detected an error in the CA XCOM Data Transport trailer, either in its format or in the number of records transferred.

Action:

This error may be the result of a send text file which contains zero-length records. If this is the case, try to send the file with the compression turned on.

0318E

Receive_and_wait failed to turn line around

Reason:

This may be a communication or internal error.

Action:

Retry the transfer. If the error persists, contact your network administrator or CA Technical Support.

0319E

Error confirming checkpoint: Descriptive message

Reason:

There is an error in confirming the checkpoint.

Action:

Possible network problems. Try turning off checkpointing. Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0320E

Error issuing confirmed for checkpoint: Descriptive message

Reason:

There is an error issuing confirmed for the checkpoint.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0321E

Error requesting trailer confirmation: Descriptive message

Reason:

An error was encountered when requesting trailer confirmation.

Action:

Check the SNA connectivity and retry. If the problem persists, contact CA Technical Support.

0322E

Preallocation User Exit failed, RC=*n*

Reason:

The user customized script specified by the XPRECMD parameter failed.

Action:

Check for errors in the user customized script specified by the XPRECMD parameter. The sample script provided is xcompre.

0403E

Cannot open output file: Descriptive message

Reason:

Cannot open the output file using the requested action.

Action:

Check the problem reported by the descriptive message, fix the problem, and retry the transfer.

0404E

Error confirming header: Descriptive message

Reason:

An error was encountered when attempting to issue a confirmed response to an incoming header.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0405E

Error receiving maxreclen: Descriptive message

Reason:

An error was encountered when trying to read an incoming maximum record length record or when trying to realloc a read buffer of the size specified in the record received.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0406E

Error while trying to read Feature Negotiation Record: Descriptive message

Reason:

An error was encountered while trying to read a feature negotiation record.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0407E

Error compress flag in feature negotiation protocol

Reason:

Compression type provided by the remote system is not supported.

Action:

Check CA XCOM Data Transport configuration file on the remote system and retry.

0408E

Error encountered in feature negotiation protocol

Reason:

An error was encountered in feature negotiation protocol.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0409E

Error encountered trying to confirm feature negotiation record: Descriptive message

Reason:

An error was encountered in trying to confirm a feature negotiation record.

Action:

Possible SNA configuration problems. Check RUSIZES. Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0410E

Error sending feature negotiation record: Descriptive message

Reason:

An error occurred in sending a feature negotiation record.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0411E

Error while confirming feature negotiation record: Descriptive message

Reason:

An error occurred while confirming a feature negotiation record.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0412E

Error receiving data: Descriptive message

Reason:

An error was detected by the SNA APPC receive and wait verb or TCP/IP while reading incoming data.

Action:

Check the TCP/IP or SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0413E

Trailer invalid

Reason:

The count in the received trailer record did not match the actual number of records received.

Action:

Make sure your version of CA XCOM Data Transport is up to date. Retry the transfer. If the problem persists, take a CA XCOM Data Transport trace and contact CA Technical Support.

0415E

Error receiving trailer: Descriptive message

Reason:

The receive and wait issued for an incoming trailer record has failed.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0416E

Error writing output file: Descriptive message

Reason:

The write() system call has returned a -1. This probably means that the file system is out of space or that the user's file size limit has been exceeded.

Action:

Check the free space in the file system. If free space is not the problem, check the size of the user's ulimit (file size limit) and compare it to the size of the file that caused the transfer to fail. Repair the problem and retry the transfer.

If you are transferring files larger than 2 GB, make sure your version of CA XCOM Data Transport supports large file sizes.

0417E

Error [opening | creating] temporary [file name | directory]

Reason:

The temporary file used during transfer processing could not be opened. This may mean that the file system is out of space or the user does not have the proper authority to create the file or directory.

Action:

Check the free space in the file system. Verify that the user has the authority to create the file or directory. Repair the problem and retry the transfer.

0418E

Transmission interrupted: signal received

Reason:

An interrupt signal has been received and transmission has been interrupted.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0420E

While setting the user id an error occurred, errno=n

Reason:

An error was encountered while setting the transfer user id.

Action:

Check that the user ID has valid user id number on the system.

0421E

While setting the group id an error occurred, errno=n

Reason:

An error was encountered while setting the transfer group id(s).

Action:

Check that the user ID has valid group id number on the system.

0422E

Login incorrect.

Reason:

User ID and/or password do not match security database.

Action:

Provide correct user ID and/or password.

0423I

Error receiving file descriptor record.

Reason:

An error occurred while trying to receive a file descriptor record.

Action:

Check the SNA connectivity and retry the transfer. If the problem persists, contact CA Technical Support.

0424E

Attempting to create an existing file

Reason:

You cannot create an existing file.

Action:

Decide whether you want to replace the target file or create one using a different name. Set the parameters appropriately and retry the transfer.

0425E

Cannot seek to position on restart

Reason:

When trying to reposition a file to restart a transfer, the fseek call returned a -1. This usually indicates a file that has been updated since its transfer was interrupted.

Action:

Determine why the file was updated and/or retry the transfer.

0426E

Restarting from beginning of file

Reason:

Checkpointing information was not found.

Action:

None required. The transfer will restart from the beginning of the file.

0427E

Error starting TP: no retry possible

Reason:

A call to execl subroutine returned a non-zero return code; the data following the explanation is the sense.

Action:

Check if the file specified in TP_CMD parameter in your xcom.glb file exists and has execute permission.

0428E

Error setting ulimit size, errorno=n

Reason:

An error was encountered while trying to set the file size limit (ulimit).

Action:

Check that the user ID has a valid file size limit (ulimit) on the system.

0435I

Waiting for an available session.

Reason:

CA XCOM Data Transport is waiting for available sessions to start a transfer.

Action:

Wait until a session becomes available or increase the session limits in the xcom.ses file, restart xcomd, and retry the transfer.

0436E

TP ended abnormally.

Reason:

The CA XCOM Data Transport daemon has detected a transaction program failure. Or, possible security problems.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0437E

Record larger than input buffer.

Reason:

The length of a file record is greater than the length of the input buffer.

Action:

Check for correct file type. May also need to increase MAXRECLEN. Retry.

0438E

Record length greater than maxreclen, but truncation not allowed.

Reason:

The length of the received record is greater than maxreclen, but truncation is not allowed.

Action:

Make sure you are specifying the correct parameters for the file and retry.

Note: On a receive operation, MAXRECLEN is set to whatever the LRECL value is on the initiating side.

0439E

Record exceeds maxreclen following compression

Reason:

The data in this record is already compressed.

Action:

Turn off compression for this transfer.

0440E

Internal buffer < maxreclen received.

Reason:

The internal buffer is smaller than the maxreclen received.

Action:

Retry the transfer. If the problem persists, run a trace and contact CA Technical Support.

0441I

Restart request IDs and group name do not match.

Reason:

A restart request was received from the remote side, but the request IDs do not match.

Action:

No action necessary; this is an informational message that indicates that the transaction program has encountered a problem during an attempted restart and must restart the transfer from the beginning.

0442I

Filename does not match on restart request.

Reason:

A restart request was received from the remote side, but the local file names do not match.

Action:

Retry the transfer. If the problem persists, run a trace and contact CA Technical Support.

0443I

Group name does not match on restart request

Reason:

A restart request was received from the remote side, but the remote group names do not match.

Action:

No action necessary; this is an informational message that indicates that the transaction program has encountered a problem during an attempted restart and must restart the transfer from the beginning.

0448E

Type of record packing not supported.

Reason:

An unsupported type of record packing was specified in the header.

Action:

Check that the remote system is generating a packing type that corresponds to a CA XCOM Data Transport supported packing type: YES, NO, MPACK or XPACK. For AT&T or NCR systems, the value of the mode definition code set field must be set to A for ASCII. If PACK=LENGTH is specified, RECSEP=NO must also be specified.

0450E

xcomd not running.

Reason:

The XCOM API program may have been compiled with a different version of CA XCOM Data Transport than what is running.

Action:

Make sure the XCOM API program was compiled at the level of CA XCOM Data Transport that you are running.

0451E

Invocation mode undefined.

Reason:

This is an internal error.

Action:

Try the transfer again and if the error persists, contact CA Technical Support.

0454E

Invalid starting date format.

Reason:

The transfer was not scheduled because the starting date cannot be parsed.

Action:

Repair the date format and retry the transfer.

0455E

Invalid starting time format.

Reason:

The transfer was not scheduled because the starting time cannot be parsed.

Action:

Repair the time format and retry the transfer.

0456E

Cannot open EBCDIC-ASCII translation file.

Reason:

Global parameter INTERNAL_CONVERSION_TABLES is set to NO and the specified EBCDIC-ASCII translation file cannot be opened.

Action:

Check that the EBCDIC to ASCII conversion table file specified in the global variable ETOA_FILENAME exists and is readable.

0457E

Cannot open ASCII-EBCDIC translation file.

Reason:

Global parameter INTERNAL_CONVERSION_TABLES is set to NO and the specified ASCII-EBCDIC translation file cannot be opened.

Action:

Check that the ASCII to EBCDIC conversion table file specified in the global variable ATOE_FILENAME exists and is readable.

0458E

Error parsing EBCDIC-ASCII file.

Reason:

An error was encountered while trying to parse the EBCDIC-ASCII translation file.

Action:

Check the EBCDIC to ASCII file. It must be 256 lines long and contain a single integer per line specified according to C conventions: a leading 0 indicates an octal number, a leading 0x indicates a hexadecimal; anything else is decimal.

0459E

Error parsing ASCII-EBCDIC file.

Reason:

An error was encountered while trying to parse the ASCII-EBCDIC translation file.

Action:

Check the ASCII to EBCDIC file. It must be 256 lines long and contain a single integer per line specified according to C conventions: a leading 0 indicates an octal number, a leading 0x indicates a hexadecimal; anything else is decimal.

0460E

Unable to Queue: *Reason*.

Reason:

CA XCOM Data Transport is unable to perform an enqueue function. The reason follows.

Action:

This message may appear for a number of reasons. If the reason is 'queue is full', remove some unnecessary entries from the queue, wait for them to be removed automatically, or increase MAX_QUEUE_ENTRIES and/or decrease EXPIRATION_TIME and restart xcomd. If the reason is 'cannot close data file,' then the file system is full. Issue a df -v command to list file systems and check to see which file system is full, and then clear disk space as needed. If the problem persists, contact CA Technical Support.

0461E

Cannot get shared memory.

Reason:

The CA XCOM Data Transport queue manager cannot get shared memory.

Action:

Increase shared memory. Retry the transfer. If the problem persists, contact CA Technical Support.

0463E

Cannot allocate shared memory.

Reason:

The CA XCOM Data Transport queue manager cannot allocate shared memory.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0464E

Cannot create shared memory.

Reason:

The CA XCOM Data Transport queue manager cannot create shared memory.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0465E

Check of xcom.glb pathnames failed.

Reason:

You must specify full paths to XLPCMD, TP_CMD, TRACE_PATH, and QUEUE_PATH in your xcom.glb file.

Action:

Check the xcom.glb file to be sure that pathnames have been specified for each of the variables listed in the Explanation. Restart the daemon after you fix any problems.

0466E

Unable to create an index file for queue.

Reason:

Unable to create an index file for the queue.

Action:

Check the permissions on the directory specified in the QUEUE_PATH variable in xcom.glb. Restart the daemon after you have repaired the problem.

0467E

Queue entry passed to xcomtp is NULL.

Reason:

This is an internal error. The queue entry passed to xcomtp is NULL.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0468E

Cannot allocate memory.

Reason:

Cannot allocate memory.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0469E

xcomd is not running or not compatible.

Reason:

The CA XCOM Data Transport scheduler, xcomd, is not running, or the version of its running copy is not compatible to the version of the CA XCOM Data Transport package you are using.

Action:

Try to start the CA XCOM Data Transport daemon. If it does not start, use xcomd -cy to clean up. If it still does not start, make sure that you have installed the right version of xcomd. Reinstall the CA XCOM Data Transport package, if necessary.

0470E

Invalid number of parameters passed to xcomtp.

Reason:

An invalid number of parameters has been passed to xcomtp.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0471E

Invocation mode undefined.

Reason:

This is an internal error. The invocation mode is undefined.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0472E

Current mode undefined.

Reason:

This is an internal error. The current mode is undefined.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0473E

Error(s) parsing xcom.glb; xcomd cannot start.

Reason:

Error(s) found in parsing /usr/spool/xcom.glb; xcomd cannot start. Possible problem with the /usr/spool/xcom/config/xcom.tid file.

Action:

Check the xcom.glb file and fix any syntax errors you find there. Make sure the xcom.tid contains a non-zero six-digit number, the next transfer ID number. Edit and correct if necessary. Restart the CA XCOM Data Transport daemon.

0474E

xcomd is already running. Multiple instances not supported.

Reason:

xcomd is already running. Multiple instances are not supported.

Action:

None. Required as xcomd is already running. In order to stop the xcomd use xcomd –s command. If it does not stop, use xcomd -cy to clean up. xcomd can now be restarted.

0475E

xcomd index file 000025.IDX is bigger than allowed for maximum number of queue entries (xx bytes).

Reason:

The xcomd index file 000025.IDX is bigger than allowed for maximum number of queue entries (xx bytes). The MAX_QUEUE_ENTRIES parameter in xcom.glb may have been reduced.

Action:

Increase the MAX_QUEUE_ENTRIES parameter in xcom.glb and restart xcomd. If the error persists remove all files in the Q directory and restart xcomd.

0476E

Error reading xcomd index file 000025.IDX, records read xx.

Reason:

When reading records from the 000025.IDX file an error occurred at record number xx.

Action:

Try restarting xcomd. If the error persists, remove all files in the Q directory and restart xcomd.

0477E

xcomd index file 000025.IDX size (xx bytes) not a multiple of record size.

Reason:

The xcomd index file 000025.IDX size must be a multiple of the record size, otherwise a corrupt file is assumed.

Action:

Remove all files in the Q directory and restart xcomd.

0478E

xcomd index file 000025.IDX contains a corrupt TID at index record xx.

Reason:

TID must be six digits. The index record contains something other than digits. The 000025.IDX file got corrupted.

Action:

Remove all files in the Q directory and restart xcomd.

0478W

An error occurred attempting to connect to the relational database: Descriptive message

Reason:

XCOM did not successfully connect to the relational database identified by xcom.glb parameters:

- XCOMHIST=name of ODBC connection
- XCOMHIST_USER=name of user authorized to use the xcomhist database.
- XCOMHIST_PASSWORD=password (may be encrypted) of the authorized user

Action:

XCOM continues without history functionality.

0479W

Error(s) parsing xcom.glb. Error processing Global Parm on Line nnnnn in the xcom.glb file is invalid. XCOMD will not start.

Reason:

Messages will be written for ALL invalid global parameters.

Action:

After correcting the above errors, restart XCOMD.

0483E

Error: unable to fork.

Reason:

Either the system or the user process limit has been reached and no additional processes can be created.

Action:

Stop some processes, or raise process limits, and retry the operation.

0486E

Error writing configuration.

Reason:

An error was encountered while writing the CA XCOM Data Transport configuration file.

Action:

Check the free space in the file system. If free space is not the problem, check the permissions on the configuration file that is being written. Repair the problem and retry the operation.

0488E

Input file not specified, but stdin cannot be used.

Reason:

When the local input file is not specified, CA XCOM Data Transport normally uses console input stream (stdin). This message indicates that stdin cannot be used for this transfer.

Action:

Specify an input file and retry.

0489E

Output file not specified: stdout cannot be used.

Reason:

When the local output file is not specified, CA XCOM Data Transport normally uses console output stream (stdout). This message indicates that stdout cannot be used for this transfer.

Action:

Specify an output file and retry. If doing a receive, LOCAL_FILE_RF must be specified.

0491E

Versions of different CA XCOM Data Transport components are not compatible.

Reason:

The versions of the CA XCOM Data Transport modules are not compatible.

Action:

Contact CA Technical Support.

0492E

MAXRECLEN exceeds 2048 bytes limit for record packing.

Reason:

Since the maximum block size for packing is 2048 bytes, MAXRECLEN should not exceed this limit if record packing is turned on.

Action:

Change MAXRECLEN so that it is not greater than 2048 bytes or turn off record packing.

0493E

Cannot generate internal transaction id

Reason:

This is an internal error.

Action:

Reduce the EXPIRATION_TIME and increase MAX_QUEUE_ENTRIES, then stop and restart the CA XCOM Data Transport scheduler, xcomd.

0494E

Duplicate transfer request number

Reason:

This is an internal error.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0495E

Compaction not supported

Reason:

SNA compression with dynamic compaction is not supported.

Action:

Turn off dynamic compaction.

0497E

Can't open display

Reason:

Cannot open the display when bringing up XCOMTOOL.

Action:

Check with your administrator or CA Technical Support.

0500E

Attempting to write to a directory as a regular file

Reason:

CA XCOM Data Transport is attempting to write a file, but directory exists with the same name.

Action:

Change the name of the file or delete directory.

0501E

Problems starting xwait. Retrying...

Reason:

For SUNAPPC and BRIXTON. Either XNODESPEC or XLUNAME is incorrect in xcom.glb or the APPC is down. xcomd will come up and will schedule requests in the queue, but no transfers will go. CA XCOM Data Transport tries to start xwait every three seconds.

Action:

Check that the SNA APPC is up. Verify that XNODESPEC and XLUNAME are set correctly in xcom.glb.

0503E

Invalid compressed data

Reason:

Unable to compress or decompress the data sent or received.

Action:

Retry the transfer. If the error persists, contact CA Technical Support.

0505E

Received a signal from (communications)

Reason:

An interrupt signal has been received from APPC (for SNA) or TCP/IP (for TCP/IP) and transmission has been interrupted.

Action:

Retry the transfer. If the problem persists, contact CA Technical Support.

0506E

Invalid pack header data

Reason:

While unpacking data an error occurred. This can be due to incorrect data coming from the remote system.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0507E

Unable to rename temporary file

Reason:

CA XCOM Data Transport was unable to rename the temporary file to the permanent file name.

Action:

Retry the transfer and if the error persists, contact CA Technical Support.

0508E

VLR/VLR2 are for binary data only

Reason:

CARRIAGE_FLAG=VLR or VLR2 is supported for binary files only.

Action:

Specify CODE_FLAG=BINARY or change CARRIAGE_FLAG to other than VLR or VLR2.

0509E

Big packing not supported by partner

Reason:

The remote system does not support the big packing type.

Action:

Retry the transfer using record packing type Y, M, or N.

0510E

Big packing not supported in version 1 transfers

Reason:

Version 1 transfers are supported for compatibility with old releases of CA XCOM Data Transport. Big packing was not supported in these old releases.

Action:

Specify version 2 to use big packing. If the remote partner does not support version 2, retry the transfer using record packing type Y, M, or N.

0511E

rc=xxx sense=yyyy

Reason:

Applies only to AIX RS/6000 systems. This error can occur when a TP is not available or because of a resource failure.

Action:

The transfer is retried automatically for queued transfers, depending on your configuration. If the problem persists, it may be an RS/6000 SNA configuration problem involving hardware or software.

0512E

Maximum remote tcp sessions reached

Reason:

The maximum number of remote TCP/IP transfer requests as specified in the parameter MAX_REMOTE_TCP in xcom.glb has been reached and the transfer is rejected.

Action:

Either increase the value of MAX_REMOTE_TCP and recycle xcomd or run less concurrent remote TCP/IP transfers.

0514E*

DESTINATION cannot exceed 16 characters for an indirect transfer

Reason:

An indirect transfer was submitted and the DESTINATION parameter is too long. The DESTINATION cannot exceed 16 characters for an indirect transfer.

Action:

Specify a DESTINATION of 16 characters or less.

0515E

Error occurred attempting to assume super-user privileges

Reason:

CA XCOM Data Transport needs system privileges (root) to be able to set the next owner, group, and file ownership and permissions correctly.

Action:

Try the transfer again. If the error persists, contact CA Technical Support.

0520E

Number of jobs to be processed by XTC cannot exceed 8.

Reason:

If number of jobs to be processed through XTC is more than eight for a transfer.

Action:

Provide at most eight jobs to be processed through XTC by each transfer in CNF or XML.

0521E

XTCJOB name already exists in XTCNET group.

Reason:

If a job name already exists in the XTCNET group.

Action:

User should try with some other job name. Can look into the trace file to know which JOB name is being duplicated for a particular XTCNET group.

0522E

XCOM is licensed

Reason:

License check for CA XCOM Data Transport is successful

Action:

None

0523E

CA LICENSE CHECK FAILED: text
where text is one of the following:
XCOM not licensed
Workgroup license count exceeded
License expired
License will expire
License dll expired
Can't open license file
Corrupt license file
Can't read license file
MAC address or serial number incorrect
Machine type incorrect
Enterprise or Workgroup grace period expired
Enterprise/TNG license count exceeded
License needs to be upgraded
License cannot be upgraded
Register later
Using the safety key
Internal Error in Licensing
Could not load CA Licensing Shared Library

Reason:

1. License check for CA XCOM Data Transport was unsuccessful.
2. CA XCOM Data Transport is not licensed.
3. The Workgroup license count has exceeded the allowed limit.
4. The license for CA XCOM Data Transport has expired.
5. The license for CA XCOM Data Transport will expire.
6. The license dll for CA XCOM Data Transport has expired.
7. Cannot open the license file.
8. The license file is corrupt.
9. Cannot read the license file.
10. The MAC address of the machine does not match the one in the license file.
11. The machine type does not match the one in the license file.
12. The Enterprise or Workgroup grace period has expired.
13. The Enterprise/TNG license count has exceeded the allowed limit.
14. The license for CA XCOM Data Transport needs to be upgraded.
15. The license for CA XCOM Data Transport cannot be upgraded.
16. Register later.

17. Using the safety key
18. Internal Error in Licensing (apart from those listed above)
19. CA Licensing did not install successfully and/or the $CALIB and $CASHCOMP environment variables are not set correctly.

Action:

1. Error message will be displayed on standard error as well as logged in to xcom.log, but CA XCOM Data Transport will continue to work.
2. Use the appropriate license file or contact the CA Licensing Group.
3. Use the appropriate license file or contact the CA Licensing Group.
4. Use the appropriate license file or contact the CA Licensing Group.
5. Contact the CA Licensing Group for a license upgrade.
6. Contact the CA Licensing Group.
7. Use the appropriate license file or contact the CA Licensing Group.
8. Use the appropriate license file or contact the CA Licensing Group.
9. Use the appropriate license file or contact the CA Licensing Group.
10. Use the appropriate license file or contact the CA Licensing Group.
11. Use the appropriate license file or contact the CA Licensing Group.
12. Contact the CA Licensing Group
13. Use the appropriate license file or contact the CA Licensing Group.
14. Contact the CA Licensing Group.
15. Contact the CA Licensing Group.
16. Contact the CA Licensing Group.
17. Contact the CA Licensing Group.
18. Contact the CA Licensing Group.
19. Ensure that CA Licensing has been installed successfully, by checking the settings of the environment variables $CALIB and $CASHCOMP. Also check that the path defined in the $CALIB environment variable is included in the system library path environment variable. If any changes are made, then the CA XCOM daemon, xcomd, will need to be recycled. If all these checks are good, then try recycling the CA XCOM daemon, xcomd, to see if this resolves the error.

0525E

CA XCOM Gateway file insertion failed

Reason:

Failed to Insert File in to associated gateway policy

Possible Reason text can be one of the following:

- Filename does not meet gateway criteria
- Not aligned to a CA XCOM Gateway
- CA XCOM Gateway returned an error
- Error Received from Gateway Control Server

Action:

- File Name should meet Gateway file name criteria. Refer to “Using CA XCOM Data Transport with CA XCOM Gateway” section in Chapter 1.
- Check in Global parameters whether GATEWAY_IP and GATEWAY_PORT are holding the correct IP address and Port number of Gateway and GATEWAY_VERSION is set to the correct version of Gateway.
- Check the Error message and contact Gateway Administrator for any configuration problems with gateway.
- Error Message will have additional information and error code returned from Gateway in case of Error Received from gateway Control Server

0526E

CA XCOM Gateway file extraction failed

Reason:

Failed to Insert File in to associated gateway policy

Possible Reason text can be one of the following:

- Filename does not meet gateway criteria
- Not aligned to a CA XCOM Gateway
- CA XCOM Gateway returned an error
- Error Received from Gateway Control Server

Action:

- File Name should meet Gateway file name criteria. Refer to “Using CA XCOM Data Transport with CA XCOM Gateway” section in Chapter 1.
- Check in Global parameters whether GATEWAY_IP and GATEWAY_PORT are holding the correct IP address and Port number of Gateway and GATEWAY_VERSION is set to the correct version of Gateway.
- Check the Error message and contact Gateway Administrator for any configuration problems with gateway.
- Error Message will have additional information and error code returned from Gateway in case of Error Received from gateway Control Server

0527E

Unknown ENC Algorithm value from Gateway

Reason:

Received Unknown Encryption Algorithm from Gateway while preparing to insert a file to Gateway Environment.

Action:

Contact Gateway Administrator.

0528E

JVM creation failed.

Reason:

Failed to Create JVM while preparing to start Gateway Transfer.

Action:

Check the JAVA is installed to correctly with the XCOM Installation. Typically it will be installed to %ProgramFiles%\ CA\SharedComponents\JRE location.

Ensure that User has full access permissions on jvm.dll.

0529E

CA XCOM Gateway Certificate Error:reason.

Reason:

Failed to read Gateway Certificate file provided in GATEWAY_CERT Global Parameter.

Possible reason could be one of the following:

- A bad parameter was passed to the open certificate routine
- File not found
- File exceeds maximum size of num bytes
- File could not be opened for read
- Not enough memory to read certificate
- I/O error reading file
- Could not get digest object from certificate
- Could not get private key from certificate
- Unknown Error Number n

Action:

Unable to open or read the Gateway Certificate file provided in GATEWAYCERT global parameter. Verify the certificate file or re generate the new certificate file.

0530E

CA XCOM Gateway failed to export file to external Server.

Reason:

CA XCOM Gateway failed to Export File to External Server. Error Message will contain detailed message with the error code received from Gateway.

Action:

Contact Gateway Administrator for any configuration changes or to know more about the Error code received from Gateway.

0531E

CA XCOM Gateway failed to import file from external Server.

Reason:

CA XCOM Gateway failed to Import File from External Server. Error Message will contain detailed message with the error code received from Gateway.

Action:

Contact Gateway Administrator for any configuration changes or to know more about the Error code received from Gateway.

0532E

SEND JOB and SEND REPORT transfers are not supported with GATEWAYPROTOCOL other than XCOM.

Reason:

GATEWAYPROTOCOL is not allowed to use any value other than XCOM in SEND JOB and SEND REPORT Transfers. If GATEWAYPROTOCOL is other than XCOM, it will be treated as Transfer between Gateway and External Server.

Action:

Avoid using GATEWAYPROTOCOL parameter in case of Send Job and Send Report or set it to Default value i.e., XCOM.

0533E

Unsupported External Server operation.

Reason:

External Server Operations are not supported with older gateway versions. Possible reason can be one of the following:

- Insert File from External Server is not supported with current allied Gateway version
- Extract File to External Server is not supported with current allied Gateway version

Action:

Check the associated Gateway version from XCOM Global Parameters. Older Gateway versions are not supported to do external server related transfers. Inert from External Server to Gateway or Extract File to External Server from Gateway are supported with Gateway R116 or above.

0542E

HOLDCOUNT value should be between 0 and 255.

Reason:

The HOLDCOUNT value for a job has done one of the following:

- Gone over 255 during GOOD/ERR increment.
- Gone below 0 during GOOD/ERR decrement.

Action:

Check the log file to find which job's HOLDCOUNT is going out of the desired range.

0543E

Error occurred while Gateway File Insertion.

Reason:

Generic error message when Gateway file insertion fails.

Action:

Check the log to find the exact error returned by Gateway and if needed contact the Gateway Administrator.

0544E

Error occurred while Gateway File Extraction.

Reason:

Generic error message when Gateway file extraction fails.

Action:

Check the log to find the exact error returned by Gateway and if needed contact the Gateway Administrator.

0545E

Unknown ENC Algorithm value from Gateway.

Reason:

The encryption algorithm returned by Gateway cannot be identified by the Gateway Bridge.

Action:

Check the log to find the exact encryption algorithm returned by Gateway and if needed contact the Gateway Administrator.

0546E

JVM creation failed.

Reason:

JVM was not created successfully.

Action:

The program exits. Connect to the correct Java version, and then restart the transfer.

0549E

XTCXXXXX parameter exceeds maximum job length

Reason:

The XTC parameter *xxxxxxxxxxxx* was entered but, when parsing it, CA XCOM Data Transport determined that the maximum length of a job had been exceeded. The maximum job length is eight bytes.

Action:

Correct the XTCXXXXX parameter and resubmit the transfer. To construct a valid XTC transfer, see the following sections in the *CA XCOM Data Transport for Windows Server/Professional User Guide*:

- Transfer Control (XTC) Parameters
- Coding Interdependent Transfers

0550E

HOLD_TRANSFER Parameter not valid for QUEUE=NO

Reason:

The user has attempted an immediate file transfer using the CA XCOM Data Transport parameter QUEUE=NO, but the user specified HOLD_TRANSFER=YES. This parameter is invalid with HOLD_TRANSFER=YES.

Action:

Change the transfer to either QUEUE=YES or HOLD_TRANSFER=NO.

0600E

Transfer is not active

Reason:

The operation which caused this message is applicable to active transfers only.

Action:

Wait until the transfer becomes active or try another operation.

0601E

Permission denied

Reason:

You do not have security rights necessary to perform the operation which caused this message.

Action:

None

0602E

General error in queue function; rc=*n*

Reason:

CA XCOM Data Transport is unable to perform an enqueue function. The reason follows.

Action:

If the reason is "cannot write/open", check to see if your file system is full. If the reason is "queue is full", remove some unnecessary entries from the queue, wait for them to be removed automatically, or increase QUEUE-SIZE and restart xcomd. If the reason is "cannot generate TID", contact CA Technical Support. Other reasons may also be displayed. Try the operation again. If you cannot resolve the error, contact CA Technical Support.

0603E

The TP state is invalid

Reason:

The operation that caused this message requires that TP be in some particular state. The current TP state is invalid for this operation.

Action:

Wait for TP to enter the proper state or try different operation.

0604E

The maximum number of clients for this server is exceeded

Reason:

No client was available.

Action:

Log off some of the clients or wait for clients to log off.

0605E

The client is already connected to this server

Reason:

A client with the same XPE name is already logged onto the server.

Action:

Try again. If the problem persists, contact CA Technical Support.

0606E

The xxxxx request is unknown to this server

Reason:

A client submitted a request that is not supported by the server. The server will ignore this request.

Action:

A server upgrade may be required. Try the transfer again. If the error persists, contact CA Technical Support.

0609E

Timeout waiting for SEND_COMPLETE event; removing client

Reason:

Server removed client from the list of logged clients because of timeout waiting for SEND_COMPLETE event.

Action:

Check the client for possible problems. Check with your administrator.

0611E

Timeout waiting for POLL_COMPLETE event; removing client

Reason:

The client did not respond.

Action:

Logout from the server and connect to it again.

0612E

Transport error - please reconnect

Reason:

Communication error.

Action:

Reconnect to the server.

0613E

The file record in not in a VLR format.

Reason:

VLR transfers can only be done on VLR format files.

Action:

Be sure that the input file is in the right format-that it was received specifying the CARRIAGE_FLAG=VLR/VLR2 parameter.

0785I

Starting TCP/IP Connection

Reason:

A TCP/IP connection is being started to a remote partner.

Action:

None

0786I

TCP/IP Connection Established

Reason:

A TCP/IP connection was successfully made to a remote partner.

Action:

None

0793I

Remote TCP/IP Connection Established

Reason:

A TCP/IP remote connection was successfully established from a remote partner.

Action:

None

0805I

TCP/IP Connection Ended

Reason:

A TCP/IP connection completed.

Action:

None

0811I

Starting Secure TCP/IP Connection

Reason:

A secure TCP/IP connection is being started to a remote partner.

Action:

None

0812I

Remote Secure TCP/IP Connection Requested

Reason:

A request for a secure TCP/IP remote connection was received from a remote partner.

Action:

None

0813I

Secure TCP/IP Handshake Complete

Reason:

Secure Socket (SSL) negotiation for the connection successfully completed. The connection is now a secure TCP/IP connection.

Action:

None

0814I

Secure TCP/IP Connection Requested

Reason:

A request for a secure TCP/IP connection was sent to the remote partner.

Action:

None

0818I

Secure TCP/IP Connection Ended

Reason:

A secure TCP/IP connection completed.

Action:

None

0820I

Compression type requires BIG packing.

Reason:

The LZ and Huffman compression routines require a buffer with at least 500 bytes for the routines to be efficient.

Action:

Specify the packing type XPACK.

0821E

Could not find Method

Reason:

This message is issued if a method is not found in Java classes, that is, if the xml tag name and the getter method name or method signature do not match.

Action:

The program exits. Put the correct jar in the <XCOM_HOME>\lib folder, and then restart the transfer.

0822E

Could not find Class

Reason:

This message is issued if a particular class is not found in Java classes.

Action:

The program exits. Put the correct jar in the <XCOM_HOME>\lib folder, and then restart the transfer.

0823E

Unknown Data type.

Reason:

This message is issued if an unsupported data type is given in a class that is not handled in the code.

Action:

The program exits. Put the correct jar in the <XCOM_HOME>\lib folder, and then restart the transfer.

0824E

Validation of Transfer Container Failed.

Reason:

This message is issued if xml is not being validated by XSD.

Action:

The program exits. Put the correct XSD in the <XCOM_HOME>\config folder, and then restart the transfer.

0825E

NULL STRING.

Reason:

This message is issued if a null value is given in xml.

Action:

The program exits. Put the correct value in xml, and then restart the transfer.

0886E

Ping action cannot be performed with TRNENCRL_CIPHER=COMPAT.

Reason:

This message is issued if the XCOM Ping action is performed with TRNENCRL_CIPHER value set to COMPAT.

Action:

Retry XCOM Ping action with TRNENCRL_CIPHER value set to ciphers other than COMPAT. See the list of ciphers allowed for TRNENCRL_CIPHER.

0898I

Unicode conversion detected malformed characters.

Reason:

Malformed characters were detected during Unicode conversion.

Action:

None. The detected malformed characters can be found in the transfer trace.

0899I

Unicode conversion performed character substitution.

Reason:

Malformed characters detected during Unicode conversion were substituted.

Action:

None. The detected malformed characters can be found in the transfer trace.

0900I

Unicode conversion skipped malformed characters.

Reason:

Malformed characters detected during Unicode conversion were skipped.

Action:

None. The detected malformed characters can be found in the transfer trace.

0901I

CARRIAGE_FLAG=XPACK has been set for the Unicode transfer.

Reason:

CARRIAGE_FLAG=XPACK is automatically set for Unicode transfer.

Action:

None

0902I

Unicode conversion summary: Descriptive message.

Reason:

Summary of Unicode conversion is written in the trace.

Action:

None

0903I

Block I/O has been disabled for Unicode transfer.

Reason:

Block I/O is not supported for Unicode transfer. Hence Block I/O has been disabled internally.

Action:

None

0904I

Unicode converter opened successfully for charset=Descriptive message

Reason:

Unicode converter is opened successfully for Unicode transfer.

Action:

None. The name of the charset, canonical name and CCSID (if available) of the charset opened for Unicode transfer can be found in the trace.

0905E

Remote XCOM server does not support Unicode. Transfer terminated.

Reason:

The remote partner returned an error indicating that the Unicode transfer (CODE_FLAG=UTF8 or CODE_FLAG=UTF16) is not supported by that version of CA XCOM Data Transport.

Action:

Upgrade the remote CA XCOM Data Transport.

0906E

Could not load ICU Library.

Reason:

Failed to load ICU (International Components for Unicode) Library required for Unicode transfer.

Action:

Check if your CA XCOM Data Transport installation is valid. The ICU library will be installed by CA XCOM Data Transport installer. Retry the operation. If the problem persists contact CA Technical Support.

0907E

LOCAL_DELIM contains invalid value.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the local_delim contains an invalid value.

Action:

Correct the value of the local_delim XCOM API member passed to XcomAPI routine. See the list of supported delimiters for local_delim in Application Programming Interface chapter in this guide.

0908E

REMOTE_DELIM contains invalid value.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the remote_delim parameter contains an invalid value.

Action:

Correct the value of the remote_delim XCOM API member passed to the XcomAPI routine. See the list of supported delimiters for remote_delim in the Application Programming Interface chapter in this guide.

0909E

MBCS_INPUTERROR contains REPLACE# succeeded by invalid value.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_inputerror parameter contains an invalid value.

Action:

Correct the value of the mbcs_inputerror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_inputerror in the Application Programming Interface chapter in this guide.

0910E

MBCS_INPUTERROR contains REPLACE# succeeded by value out of range(0-1114111).

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_inputerror parameter contains an invalid value.

Action:

Correct the value of the mbcs_inputerror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_inputerror in the Application Programming Interface chapter in this guide.

0911E

MBCS_INPUTERROR value is invalid.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_inputerror parameter contains an invalid value.

Action:

Correct the value of the mbcs_inputerror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_inputerror in the Application Programming Interface chapter in this guide.

0912E

MBCS_CONVERROR contains REPLACE# succeeded by invalid value.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_converror parameter contains an invalid value.

Action:

Correct the value of the mbcs_converror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_converror in the Application Programming Interface chapter in this guide.

0913E

MBCS_CONVERROR contains REPLACE# succeeded by value out of range(0-1114111).

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_converror parameter contains an invalid value.

Action:

Correct the value of the mbcs_converror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_converror in the Application Programming Interface chapter in this guide.

0914E

MBCS_CONVERROR value is invalid.

Reason:

CA XCOM Data Transport API (XcomAPI) returns this error code when the mbcs_converror parameter contains an invalid value.

Action:

Correct the value of the mbcs_converror XCOM API member passed to the XcomAPI routine. See the list of correct formats for mbcs_converror in the Application Programming Interface chapter in this guide.

0915E

Failed to open Unicode converter.

Reason:

Failed to open Unicode converter for the Unicode transfer (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

Action:

Check the LOCAL_CHARSET, REMOTE_CHARSET or DEFAULT_CHARSET values. See the list of charsets supported in the Appendix chapter in this guide.

0916E

Failed to open Unicode converter for [Descriptive Message].

Reason:

Failed to open Unicode converter for the Unicode transfer (CODE_FLAG=UTF8 or CODE_FLAG=UTF16).

Action:

Check the LOCAL_CHARSET, REMOTE_CHARSET or DEFAULT_CHARSET values. See the list of charsets supported in the Appendix chapter in this guide.

0917E

Failed to open internal Unicode converter for [Descriptive Message].

Reason:

Failed to open the Unicode converter for translating the Unicode data to the negotiated Unicode encoding internally.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0920E

Failed to allocate memory for Unicode conversion.

Reason:

Failed to allocate the memory required for Unicode conversion.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0921E

Invalid MBCS_CONVERROR replacement Unicode character.

Reason:

The replacement character specified with the MBCS_CONVERROR=REPLACE#nnnnnn option is not a valid Unicode character.

Action:

Specify a valid Unicode character in the decimal value for the MBCS_CONVERROR=REPLACE#nnnnnn option.

0922E

Invalid MBCS_INPUTERROR replacement Unicode character.

Reason:

The replacement character specified with the MBCS_INPUTERROR=REPLACE#nnnnnn option is not a valid Unicode character.

Action:

Specify a valid Unicode character in the decimal value for the MBCS_INPUTERROR=REPLACE#nnnnnn option.

0923E

MBCS_CONVERROR replacement character cannot be converted to [Descriptive Message].

Reason:

The replacement character specified with the MBCS_CONVERROR=REPLACE#nnnnnn option cannot be converted to the specified charset.

Action:

Specify a valid Unicode character in the decimal value for the MBCS_CONVERROR=REPLACE#nnnnnn option which can be converted to the charset.

0924E

MBCS_INPUTERROR replacement character cannot be converted to [Descriptive Message].

Reason:

The replacement character specified with the MBCS_INPUTERROR=REPLACE#nnnnnn option cannot be converted to the specified charset.

Action:

Specify a valid Unicode character in the decimal value for the MBCS_INPUTERROR=REPLACE#nnnnnn option which can be converted to the charset.

0925E

Unicode conversion failed.

Reason:

An error occurred during Unicode conversion.

Action:

Check the actual error message in the transfer trace.

0926E

Failed to convert to Unicode. rc=[Descriptive message]

Reason:

An error occurred during converting from the specified charset to Unicode encoding.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0927E

Failed to convert from Unicode. rc=[Descriptive message]

Reason:

An error occurred during converting from the Unicode encoding to specified charset.

Action:

Retry the operation. If the problem persists, contact CA Technical Support.

0929E

Unicode conversion detected malformed characters. Transfer terminated.

Reason:

An error occurred during Unicode conversion.

Action:

None. The detected malformed characters can be found in the transfer trace.

0930I

Checkpoint restart feature has been disabled for Non BlockIO transfers with Encryption At Rest feature.

Reason:

Checkpoint restart is not supported for Encryption At Rest when BlockIO is disabled (CACHE_READ_SZ and CACHE_WRITE_SZ set to 0). Hence Checkpoint restart is internally disabled.

Action:

None. The transfer will restart from the beginning of the file.

0931I

Checkpoint restart feature has been disabled for digest.

Reason:

Checkpoint restart is not supported with (L)EAR_HASH. Hence Checkpoint restart is internally disabled.

Action:

None. The transfer will restart from the beginning of the file.

0932E

TRUNCATION IS NOT SUPPORTED FOR UNICODE TRANSFERS

Reason:

Using truncation for Unicode data can result in data loss or corruption if truncation occurs in the middle of an MBCS character.

Action:

Initiate the transfer without using truncation. Insure that a large enough logical record length (LRECL) is specified that can hold Unicode data which may have a larger LRECL than the source data.

0981E

SUBSCRIPT VALUE IS NOT NUMERIC

Reason:

This message is issued if the subscript provided to a symbolic variable is not numeric, for example &LUNAME(a,3).

Action:

The program exits. Provide proper (numeric) subscripts.

0982E

SUBSCRIPT LENGTH IS NOT VALID

Reason:

This message is issued if the subscripts length exceeds eight numeric digits or if the addition of the offset and length parts exceeds the total length of the symbolic value.

Action:

The program exits. Provide the proper value for the subscript.

0983E

SUBSCRIPT CONTAINS INVALID CHARACTERS

Reason:

This message is issued if the subscripts provided to a symbolic variable are not numeric, for example &LUNAME(#,3).

Action:

The program exits. Provide proper (numeric) subscripts.

0984E

ZERO SUBSCRIPT VALUE ENTERED

Reason:

This message is issued if a subscript provided for a symbolic variable is zero, for example &LUNAME(0,3).

Action:

The program exits. Provide proper (non-zero) subscripts.

0985E

SUBSCRIPT OUT OF RANGE FOR SPECIFIED SYMBOLIC

Reason:

This message is issued if a subscript is not in the prescribed format for the symbolic, for example &DATE(*DDMMYY*).

Action:

The program exits. Provide subscripts in the proper format for the symbolic.

0986E

Incorrect destination address

Reason:

The "to" address is incorrect for SMTP notification.

Action:

Notification is not sent.

To specify the "to" address (destination address), set one of the following parameters:

- For *local* notification:
  - LOCAL_NOTIFY (in the configuration file)
- For *remote* notification:
  - NOTIFY_NAME (in the configuration file)

0987E

Incorrect source address

Reason:

The "from" address is incorrect for SMTP notification.

Action:

Notification is not sent.

Correct the "from" address (source address) by setting one of the following parameters:

- For *local* notification:
  - MAIL_USERNAME (in the global file or the configuration file)
  - FILE_ACCESS_USERNAME (in the global file or the configuration file)
- For *remote* notification:
  - MAIL_USERNAME (in the global file)
  - USERID (in the configuration file)
  - XCOM_USERID (in the global file)

0988E

Error sending notification, Check SMTP Server name, source or destination address

Reason:

This message is issued if the "to" or "from" address is incorrect or the SMTP server name is incorrect in the SMTP notification.

Action:

Notification is not sent.

To specify the SMTP server name, set the following parameter:

- SMTP_SERVER (in the global file)

To specify the "to" address (destination address), set one of the following parameters:

- For *local* notification:
  - LOCAL_NOTIFY (in the configuration file)
- For *remote* notification:
  - NOTIFY_NAME (in the configuration file)

To specify the "from" address (source address), set one of the following parameters:

- For *local* notification:
  - MAIL_USERNAME (in the global file or the configuration file)
  - FILE_ACCESS_USERNAME (in the global file or the configuration file)
- For *remote* notification:
  - MAIL_USERNAME (in the global file)
  - USERID (in the configuration file)
    - XCOM_USERID (in the global file)

1300I

REQUEST #xxxxxx PROCESSED SUCCESSFULLY

Reason:

CA XCOM Data Transport has successfully scheduled a file transfer request. The CA XCOM Data Transport server on which the request is scheduled logs this message and sends it to the user requesting the schedule operation. *xxxxxx* is the request number that is assigned to the file transfer request.

Action:

None

1301E

SCHEDULE REQUEST FAILED : reason

Reason:

Possible reason text includes the following:

- Xrcvwt FAILED
- Xrcvwt/Xcnfrmed FAILED
- Xsnddta FAILED
- Xprprcv FAILED
- CANNOT GENERATE TID
- XCOMD NOT RUNNING
- ERROR_CREATE_EVENT
- ERROR_WAIT_EVENT
- QUEUE FULL
- UNABLE TO QUEUE TRANSFER
- ERROR SETTING LOCAL USERID
- A PASSWORD IS REQUIRED FOR THIS TRANSFER
- INTERNAL ERROR
- NO PID WAS PASSED IN SCHDSECT
- USERNAME CANNOT BE OVERRIDDEN
- INVALID INVOCATION MODE
- ONLY TCP/SNA Supported
- ERROR IN INPUT SCHEDULE DATE/TIME

Action:

Depends on reason.

1303E

Invalid request, improper state for function

Reason:

These messages are returned when an operator request fails for one of the following reasons:

- Invalid request
- Improper state for function

For example, a request to suspend a transfer that is not active is an improper state

Action:

Depends on reason.

Error_Q_Function returned from QAPI

Reason:

CA XCOM Data Transport called the QAPI, which returned an error code.

Action:

Turn on CA XCOM Data Transport tracing as instructed by CA Technical Support.

Not authorized to make this request

Reason:

An operator request was made for which the requesting user was not authorized.

Action:

Users can generally view and modify requests that they submitted.

To authorize a particular user can be view and modify other transfers, add that user to one of the following groups:

- XCOMADM—To view and modify other users' transfers
- XCOMSADM—To view and modify transfers from all machines

Operator request failed, xxxxx.

Reason:

An operator request has failed for a reason other than those mentioned above.

Action:

A trace must be obtained to debug this problem.

1304E

Error creating decrypt object.

Reason:

This message is returned when one of the following occurs:

- LEAR_CIPHER or EAR_CIPHER is not valid.
- LEAR_KEY or EAR_KEY is not valid.

Action:

Verify the LEAR_CIPHER/EAR_CIPHER/LEAR_KEY/EAR_KEY parameters.

1305E

Error creating encrypt object.

Reason:

This message is returned when one of the following occurs:

- LEAR_CIPHER or EAR_CIPHER is not valid.
- LEAR_KEY or EAR_KEY is not valid.

Action:

Verify the LEAR_CIPHER/EAR_CIPHER/LEAR_KEY/EAR_KEY parameters.

1306E

Digest does not match.

Reason:

This message is returned when the user-supplied digest does not match with the CA XCOM Data Transport generated digest at the sending side.

Action:

1. Verify the LEAR_DIGEST/EAR_DIGEST/LEAR_HASH/EAR_HASH parameters.
2. Check if the file is tempered.

1307E

Error creating digest object.

Reason:

This message is returned when LEAR_HASH or EAR_HASH is not valid.

Action:

Verify the LEAR_HASH/EAR_HASH parameters.

1308E

Unable to initialize ETPKI Library.

Reason:

This message is returned when CA XCOM Data Transport cannot load the CAPKI (also known as ETPKI) library.

CA XCOM Data Transport requires the CAPKI library for the following reasons:

- Password encryption with cipher negotiation
- Encryption at rest
- CA XCOM Gateway transfers
- SSL transfers

The CA XCOM Data Transport installer installs the CAPKI library. A default path is updated in the xcom.glb library.

Action:

1. Verify if the ETPKI library is installed correctly.
2. Search libcapki.dll to verify if the CAPKIHOME global parameter is set to the directory where ETPKI is installed.

1309E

Invalid Cipher or Digest or fipsmode parameters.

Reason:

This message is retuned when one of the following occurs:

1. If FIPS mode is on and cipher is RC4/RC2.
2. If FIPS mode is on and hash is MD4/MD5.

Action:

1. Set FIPS_MODE=NO if you want to use RC4/RC2 cipher.
2. Set FIPS_MODE=NO if you want to use MD4/MD5 hash.

1310E

Transfer nnnnnn XXXXXXXXXX by user uuuuuuuu at ssssssss/iiii

Reason:

nnnnnn is the transfer number.

XXXXXXXXXX can be one of the following:

- HELD
- RELEASED
- TERMINATED
- CANCELLED
- PURGED
- SUSPENDED
- RESUMED
- ALTERED Start Time
- ALTERED Start Date
- ALTERED Schedule Priority

uuuuuuuu is the logged on user ID

ssssssss is the SYSTEM NAME as defined in the xcom.glb file (or xcomdflt on z/OS)

iiii is the SYSTEM ID as defined in the xcom.glb file (or xcomdflt on z/OS)

Action:

None

1312E

XTC parameters not supported for QUEUE=NO

Reason:

The XTC function works only if QUEUE=YES is specified. If you specify QUEUE=NO, CA XCOM Data Transport flags all XTC parameters as errors, and terminates the job.

Action:

Specify QUEUE=YES if you need to use XTC parameters.

1313E

Restart transfer is not supported when cipher mode is OFB.

Reason:

This message is returned when cipher mode is OFB and CA XCOM Data Transport restarts a transfer.

Action:

If you want to restart the transfer with encryption at rest, choose a different cipher.

1314E

Restart transfer is not supported when cipher is RC4.

Reason:

This message is returned when LEAR_CIPHER or EAR_CIPHER is RC4 and CA XCOM Data Transport restarts a transfer.

Action:

If you want to restart the transfer with encryption at rest, choose a different cipher.

1315E

Restart transfer is not supported when hash is set.

Reason:

This message is returned when LEAR_HASH or EAR_HASH is set and CA XCOM Data Transport restarts a transfer.

Action:

If you want to restart the transfer, remove LEAR_HASH or EAR_HASH.

1317E

Local discover request failed.

Reason:

An error was detected while attempting to send a discovery request. Possible reason could be the failure of Xrcvwt.

Action:

Check the Network connectivity and retry the transfer.

1325E

Remote Discover Request failed: reason

Reason:

Possible reason could be one of the following:

- Xrcvwt FAILED
- Xrcvwt/Xcnfrmed FAILED
- Xsnddta FAILED
- Xprprcv FAILED
- Cipher match not found
- Error occurred while matching the cipher
- Cannot initialize ETPKI library
- Error code : code

Action:

Depends on the reason that is returned in the message.

For the first four reasons, check the network connectivity and retry the transfer.

If the cipher match is not found, recheck the Local and Remote cipher preferences set in xcom.glb. Refer to How Transmission Password Encryption Works for more information about how to set the (STC)TRNENCRL_CIPHER/TRNENCRR_CIPHER parameters.

If ETPKI (CAPKI) Initialization fails, check whether the ETPKI (CAPKI) Library is installed properly and that the CAPKIHOME global parameter is set to the correct value.

1326E

Local ping failed.

Reason:

xcomtcp returns this error code when the required arguments are not supplied to the XCOM Ping request.

Action:

Provide the required arguments to the XCOM Ping request.

1327E

Transfer is not allowed without Password cipher negotiation.

Reason:

This error is produced when TRNENCRR_CIPHER on remote XCOM Server does not have COMPAT in the list of ciphers and

- Transfer is initiated with TRNENCRL_CIPHER=COMPAT.
- Transfer is initiated from CA XCOM Data Transport version which does not support Password cipher negotiation.

Action:

Add COMPAT to the TRNENCRR_CIPHER global parameter on remote XCOM server to permit the CA XCOM Data Transport proprietary cipher without cipher negotiation that is required for backward password compatibility with CA XCOM Data Transport versions before 11.6

or

Choose different cipher for TRNENCRL_CIPHER other than COMPAT, in order to encrypt the password when in flight.

1427E

Password cipher negotiation failed: reason

Reason:

Possible reason could be one of the following:

- Cipher match not found
- Partner returned cipher error
- Cannot initialize ETPKI library

Action:

Depends on reason.

If the cipher match is not found, recheck the Local and Remote cipher preferences set in xcom.glb. Refer to How Transmission Password Encryption Works for more information.

Recheck the remote partner cipher preferences. Refer to How Transmission Password Encryption Works for more information about how to set the (STC)TRNENCRL_CIPHER/TRNENCRR_CIPHER parameters.

Check whether the ETPKI (CAPKI) Library is installed properly and that the CAPKIHOME global parameter is set to the correct value.

3646E

Error in getting Meta Connection.[Descriptive message]

Exception Message=[Descriptive message]

Reason:

This error is produced when a request is made for a Meta Connection and it fails.

Action:

Verify the TCP/IP software requirements have been installed and configured on the system.

3648E

Error in Submitting the Configuration.[Descriptive message]

Exception Message=[Descriptive message]

Reason:

This error is produced when a request is made to submit a transfer from the GUI and it fails.

Action:

Verify the TCP/IP software requirements have been installed and configured on the system.

3680E

User is not authorized.

Reason:

This error is produced when a request is made for another user's history, and the logged on user is not authorized for this type of request.

Action:

To authorize a specific user to see transfers other than their own, add the user to the XCOMADM group.

3685E

Local CIPHER List Contains Invalid Ciphers or Invalid Characters.

Reason:

The entered Local Cipher List contains invalid characters and/or cipher values. The valid cipher values are: "ALL", "AES", "AES128", "AES192", "AES256", "DES", "3DES", "RC2", "AES128-CBC", "AES128-ECB", "AES128-CFB", "AES128-OFB", "AES192-CBC", "AES192-ECB", "AES192-CFB", "AES192-OFB", "AES256-CBC", "AES256-ECB", "AES256-CFB", "AES256-OFB", "DES-CBC", "DES-ECB", "DES-CFB", "DES-OFB", "3DES-CBC", "3DES-ECB", "3DES-CFB", "3DES-OFB", "RC2-CBC", "RC2-ECB", "RC2-CFB", "RC2-OFB", "XCOM", "RC4", "COMPAT" separated by : and optionally preceded by !, +, or - character.

Action:

Correct the list and repeat the action.

3686E

Remote CIPHER List Contains Invalid Ciphers or Invalid Characters.

Reason:

The entered Local Cipher List contains invalid characters and/or cipher values. The valid cipher values are: "ALL", "AES", "AES128", "AES192", "AES256", "DES", "3DES", "RC2", "AES128-CBC", "AES128-ECB", "AES128-CFB", "AES128-OFB", "AES192-CBC", "AES192-ECB", "AES192-CFB", "AES192-OFB", "AES256-CBC", "AES256-ECB", "AES256-CFB", "AES256-OFB", "DES-CBC", "DES-ECB", "DES-CFB", "DES-OFB", "3DES-CBC", "3DES-ECB", "3DES-CFB", "3DES-OFB", "RC2-CBC", "RC2-ECB", "RC2-CFB", "RC2-OFB", "XCOM", "RC4", "COMPAT" separated by : and optionally preceded by !, +, or - character.

Action:

Correct the list and repeat the action.

3689E

UNICODE Input Character Error Action is Invalid.

Reason:

The entered action is invalid. The valid actions are "FAIL","SKIP","REPLACE", or “REPLACE#nnnnnnn”. Where nnnnnnn is a number in the range of 0 through 1114111.

Action:

Correct the entered action and repeat the process.

3690E

UNICODE Convert Character Error Action is Invalid.

Reason:

The entered action is invalid. The valid action is "FAIL","SKIP","REPLACE", or “REPLACE#nnnnnnn”. Where nnnnnnn is a number in the range of 0 through 1114111.

Action:

Correct the entered action and repeat the process.

3691E

UNICODE Default Delimiter is Invalid.

Reason:

The entered default delimiter list contains invalid characters and/or delimiter values and/or mutually exclusive values. The valid delimiter values are “EBCDIC", "ASCII", "CR", "LF", "NL", "VT", "FF", "LS", "PS", "CRLF", "LFCR", "CRNL" separated by (:) character. For mutually exclusive values and the other rules related to the delimiter values, see DEFAULT_DELIM parameter description in the XCOM.GLB Parameters section of the Administration Guide

Action:

Correct the list and repeat the action.

3692E

Character-Set Input Error is Invalid.

Reason:

The entered value is invalid. The valid value is "FAIL","SKIP","REPLACE", or “REPLACE#nnnnnnn”. Where nnnnnnn is a number in the range of 0 through 1114111.

Action:

Correct the entered value and repeat the process.

3693E

Character-Set Convert Error is Invalid.

Reason:

The entered value is invalid. The valid value is "FAIL","SKIP","REPLACE", or “REPLACE#nnnnnnn”. Where nnnnnnn is a number in the range of 0 through 1114111.

Action:

Correct the entered value and repeat the process.

3694E

Local Record Delimiter is Invalid.

Reason:

The entered default delimiter list contains invalid characters and/or delimiter values and/or mutually exclusive values. The valid delimiter values are: “EBCDIC", "ASCII", "CR", "LF", "NL", "VT", "FF", "LS", "PS", "CRLF", "LFCR", "CRNL" separated by (:) character. For mutually exclusive values and the other rules related to the delimiter values, see the LOCAL_DELIM parameter description in the List of Parameters section of this guide.

Action:

Correct the list and repeat the action.

3695E

Remote Record Delimiter is Invalid.

Reason:

The entered default delimiter list contains invalid characters and/or delimiter values and/or mutually exclusive values. The valid delimiter values are: “EBCDIC", "ASCII", "CR", "LF", "NL", "VT", "FF", "LS", "PS", "CRLF", "LFCR", "CRNL" separated by (:) character. For mutually exclusive values and the other rules related to the delimiter values, see the REMOTE_DELIM parameter description in the List of Parameters section of this guide.

Action:

Correct the list and repeat the action.

Appendix C: Character Sets for Unicode Transfer

This section lists the character sets supported by CA XCOM Data Transport for Unicode Transfers.

IANA is the main source of converter aliases on the Internet. Since IANA does not specify the Unicode mappings for every codepage and alias, and every platform supports other aliases besides the IANA aliases, ICU (International Components for Unicode) is used by CA XCOM Data Transport to provide a way to target the codepage conversion based upon the standard or platform. This allows you to use the right converter name and implementation based upon which standard you are targeting.

The column marked as 'Canonical Name' is a Unique ICU converter name. The column marked as 'IANA' specifies the code page name as per IANA registry. The column marked as 'All Aliases' has all the aliases by which specified code page is known regardless of which standards support the converter alias name.

Character sets are categorized according to converter type. Character sets are separated with the space.

This section contains the following topics:

Unicode (see page 597)
LATIN_1 (see page 598)
US_ASCII (see page 599)
Multibyte character sets (MBCS) (see page 599)
Single Byte Character Sets (SBCS) (see page 602)
Double-byte character sets (DBCS) (see page 613)
ISO_2022 (see page 613)
HZ (see page 614)
Indian Script Code for Information Interchange (ISCII) (see page 614)
LMBCS_1 (see page 615)
EBCDIC_STATEFUL (see page 615)

Unicode

Canonical Name	IANA	All Aliases
UTF-8	UTF-8	UTF-8 ibm-1208 ibm-1209 ibm-5304 ibm-5305 ibm-13496 ibm-13497 ibm-17592 ibm-17593 windows-65001 cp1208 x-UTF_8J

UTF-16	UTF-16 ISO-10646-UCS-2	UTF-16 ISO-10646-UCS-2 ibm-1204 ibm-1205 Unicode csUnicode ucs-2
UTF-16BE	UTF-16BE	UTF-16BE x-utf-16be Unicode Big Unmarked ibm-1200 ibm-1201 ibm-13488 ibm-13489 ibm-17584 ibm-17585 ibm-21680 ibm-21681 ibm-25776 ibm-25777 ibm-29872 ibm-29873 ibm-61955 ibm-61956 windows-1201 cp1200 cp1201 UTF16_BigEndian
UTF-16LE	UTF-16LE	UTF-16LE x-utf-16le Unicode Little Unmarked ibm-1202 ibm-1203 ibm-13490 ibm-13491 ibm-17586 ibm-17587 ibm-21682 ibm-21683 ibm-25778 ibm-25779 ibm-29874 ibm-29875 UTF16_LittleEndian windows-1200

LATIN_1

Canonical Name	**IANA**	**All Aliases**
ISO-8859-1	ISO_8859-1:1987 ISO-8859-1 IBM819 cp819 latin1 csISOLatin1 iso-ir-100 l1	ISO-8859-1 ibm-819 IBM819 cp819 latin1 8859_1 csISOLatin1 iso-ir-100 ISO_8859-1:1987 l1 819

US_ASCII

Canonical Name	IANA	All Aliases
US-ASCII	ANSI_X3.4-1968 US-ASCII ASCII ANSI_X3.4-1986 ISO_646.irv:1991 ISO646-US us csASCII iso-ir-6 cp367 IBM367	US-ASCII ASCII ANSI_X3.4-1968 ANSI_X3.4-1986 ISO_646.irv:1991 iso_646.irv:1983 ISO646-US us csASCII iso-ir-6 cp367 ascii7 646 windows-20127 ibm-367 IBM367

Multibyte character sets (MBCS)

Canonical Name	IANA	All Aliases
gb18030	gb18030	gb18030 ibm-1392 windows-54936 GB18030
ibm-942_P12A-1999		ibm-942_P12A-1999 ibm-942 ibm-932 cp932 shift_jis78 sjis78 ibm-942_VSUB_VPUA ibm-932_VSUB_VPUA x-IBM942 x-IBM942C
ibm-943_P15A-2003	Shift_JIS MS_Kanji csShiftJIS windows-31j csWindows31J	ibm-943_P15A-2003 ibm-943 Shift_JIS MS_Kanji csShiftJIS windows-31j csWindows31J x-sjis x-ms-cp932 cp932 windows-932 cp943c IBM-943C ms932 pck sjis ibm-943_VSUB_VPUA x-MS932_0213 x-JISAutoDetect
ibm-943_P130-1999		ibm-943_P130-1999 ibm-943 Shift_JIS cp943 943 ibm-943_VASCII_VSUB_VPUA x-IBM943

ibm-33722_P12A_P12A-2004_U2	Extended _UNIX _Code _Packed _Format _for _Japanese EUC-JP csEUCPkdFmtJapanese	ibm-33722_P12A_P12A-2004_U2 ibm-33722 ibm-5050 EUC-JP Extended_UNIX_Code_Packed_Format_for_Japanese csEUCPkdFmtJapanese X-EUC-JP windows-51932 ibm-33722_VPUA IBM-eucJP
ibm-33722_P120-1999		ibm-33722_P120-1999 ibm-33722 ibm-5050 cp33722 33722 ibm-33722_VASCII_VPUA x-IBM33722 x-IBM33722A x-IBM33722C
ibm-954_P101-2007		ibm-954_P101-2007 ibm-954 EUC-JP Extended_UNIX_Code_Packed_Format_for_Japanese csEUCPkdFmtJapanese X-EUC-JP eucjis ujis x-IBM954 x-IBM954C
ibm-1373_P100-2002		ibm-1373_P100-2002 ibm-1373 windows-950
windows-950-2000	Big5 csBig5	windows-950-2000 Big5 csBig5 windows-950 x-windows-950 x-big5
ibm-950_P110-1999		ibm-950_P110-1999 ibm-950 cp950 950 x-IBM950
ibm-1375_P100-2007	Big5-HKSCS	ibm-1375_P100-2007 ibm-1375 Big5-HKSCS big5hk HKSCS-BIG5
ibm-5471_P100-2006		ibm-5471_P100-2006 ibm-5471 Big5-HKSCS MS950_HKSCS hkbig5 big5-hkscs:unicode3.0 x-MS950-HKSCS
ibm-1386_P100-2001		ibm-1386_P100-2001 ibm-1386 cp1386 windows-936 ibm-1386_VSUB_VPUA
windows-936-2000	GBK CP936 MS936 windows-936	windows-936-2000 GBK CP936 MS936 windows-936

ibm-1383_P110-1999	GB2312 csGB2312	ibm-1383_P110-1999 ibm-1383 GB2312 csGB2312 cp1383 1383 EUC-CN ibm-eucCN hp15CN ibm-1383_VPUA
ibm-964_P110-1999		ibm-964_P110-1999 ibm-964 EUC-TW ibm-eucTW cns11643 cp964 964 ibm-964_VPUA x-IBM964
ibm-949_P110-1999		ibm-949_P110-1999 ibm-949 cp949 949 ibm-949_VASCII_VSUB_VPUA x-IBM949
ibm-949_P11A-1999		ibm-949_P11A-1999 ibm-949 cp949c ibm-949_VSUB_VPUA x-IBM949C IBM-949C
ibm-970_P110_P110-2006_U2	EUC-KR csEUCKR	ibm-970_P110_P110-2006_U2 ibm-970 EUC-KR KS_C_5601-1987 windows-51949 csEUCKR ibm-eucKR KSC_5601 5601 cp970 970 ibm-970_VPUA x-IBM970
ibm-1363_P11B-1998	KS_C_5601-1987 KS_C_5601-1989 KSC_5601 csKSC56011987 korean iso-ir-149	ibm-1363_P11B-1998 ibm-1363 KS_C_5601-1987 KS_C_5601-1989 KSC_5601 csKSC56011987 korean iso-ir-149 cp1363 5601 ksc windows-949 ibm-1363_VSUB_VPUA x-IBM1363C
ibm-1363_P110-1997		ibm-1363_P110-1997 ibm-1363 ibm-1363_VASCII_VSUB_VPUA x-IBM1363
windows-949-2000		windows-949-2000 windows-949 KS_C_5601-1987 KS_C_5601-1989 KSC_5601 csKSC56011987 korean iso-ir-149 ms949 x-KSC5601

Single Byte Character Sets (SBCS)

Canonical Name	IANA	All Aliases
ibm-912_P100-1995	ISO_8859-2:1987 ISO-8859-2 latin2 csISOLatin2 iso-ir-101 l2	ibm-912_P100-1995 ibm-912 ISO-8859-2 ISO_8859-2:1987 latin2 csISOLatin2 iso-ir-101 l2 8859_2 cp912 912 windows-28592
ibm-913_P100-2000	ISO_8859-3:1988 ISO-8859-3 latin3 csISOLatin3 iso-ir-109 l3	ibm-913_P100-2000 ibm-913 ISO-8859-3 ISO_8859-3:1988 latin3 csISOLatin3 iso-ir-109 l3 8859_3 cp913 913 windows-28593
ibm-914_P100-1995	ISO_8859-4:1988 ISO-8859-4 latin4 csISOLatin4 iso-ir-110 l4	ibm-914_P100-1995 ibm-914 ISO-8859-4 latin4 csISOLatin4 iso-ir-110 ISO_8859-4:1988 l4 8859_4 cp914 914 windows-28594
ibm-915_P100-1995	ISO_8859-5:1988 ISO-8859-5 cyrillic csISOLatinCyrillic iso-ir-144	ibm-915_P100-1995 ibm-915 ISO-8859-5 cyrillic csISOLatinCyrillic iso-ir-144 ISO_8859-5:1988 8859_5 cp915 915 windows-28595
ibm-1089_P100-1995	ISO_8859-6:1987 ISO-8859-6 arabic csISOLatinArabic iso-ir-127 ECMA-114 ASMO-708 ISO-8859-6-I ISO-8859-6-E	ibm-1089_P100-1995 ibm-1089 ISO-8859-6 arabic csISOLatinArabic iso-ir-127 ISO_8859-6:1987 ECMA-114 ASMO-708 8859_6 cp1089 1089 windows-28596 ISO-8859-6-I ISO-8859-6-E x-ISO-8859-6S
ibm-9005_X110-2007	ISO_8859-7:1987 ISO-8859-7 greek greek8 ELOT_928 ECMA-118 csISOLatinGreek iso-ir-126	ibm-9005_X110-2007 ibm-9005 ISO-8859-7 greek greek8 ELOT_928 ECMA-118 csISOLatinGreek iso-ir-126 ISO_8859-7:1987 windows-28597 sun_eu_greek

ibm-813_P100-1995		ibm-813_P100-1995 ibm-813 ISO-8859-7 greek greek8 ELOT_928 ECMA-118 csISOLatinGreek iso-ir-126 ISO_8859-7:1987 8859_7 cp813 813
ibm-5012_P100-1999	ISO_8859-8:1988 ISO-8859-8 hebrew csISOLatinHebrew iso-ir-138 ISO-8859-8-I ISO-8859-8-E	ibm-5012_P100-1999 ibm-5012 ISO-8859-8 hebrew csISOLatinHebrew iso-ir-138 ISO_8859-8:1988 ISO-8859-8-I ISO-8859-8-E 8859_8 windows-28598 hebrew8
ibm-916_P100-1995		ibm-916_P100-1995 ibm-916 cp916 916
ibm-920_P100-1995	ISO_8859-9:1989 ISO-8859-9 latin5 csISOLatin5 iso-ir-148 l5	ibm-920_P100-1995 ibm-920 ISO-8859-9 latin5 csISOLatin5 iso-ir-148 ISO_8859-9:1989 l5 8859_9 cp920 920 windows-28599 ECMA-128 turkish8 turkish
iso-8859_10-1998	ISO-8859-10 iso-ir-157 l6 ISO_8859-10:1992 csISOLatin6 latin6	iso-8859_10-1998 ISO-8859-10 iso-ir-157 l6 ISO_8859-10:1992 csISOLatin6 latin6
iso-8859_11-2001		iso-8859_11-2001 ISO-8859-11 thai8 x-iso-8859-11
ibm-921_P100-1995	ISO-8859-13	ibm-921_P100-1995 ibm-921 ISO-8859-13 8859_13 windows-28603 cp921 921 x-IBM921
iso-8859_14-1998	ISO-8859-14 iso-ir-199 ISO_8859-14:1998 latin8 iso-celtic l8	iso-8859_14-1998 ISO-8859-14 iso-ir-199 ISO_8859-14:1998 latin8 iso-celtic l8
ibm-923_P100-1998	ISO-8859-15 Latin-9	ibm-923_P100-1998 ibm-923 ISO-8859-15 Latin-9 l9 8859_15 latin0 csisolatin0 csisolatin9 iso8859_15_fdis cp923 923 windows-28605

windows-874-2000		windows-874-2000 TIS-620 windows-874 MS874 x-windows-874
ibm-874_P100-1995	TIS-620	ibm-874_P100-1995 ibm-874 ibm-9066 cp874 TIS-620 tis620.2533 eucTH x-IBM874
ibm-1162_P100-1999		ibm-1162_P100-1999 ibm-1162
ibm-437_P100-1995	IBM437 cp437 437 csPC8CodePage437	ibm-437_P100-1995 ibm-437 IBM437 cp437 437 csPC8CodePage437 windows-437
ibm-720_P100-1997		ibm-720_P100-1997 ibm-720 windows-720 DOS-720 x-IBM720
ibm-737_P100-1997		ibm-737_P100-1997 ibm-737 IBM737 cp737 windows-737 737 x-IBM737
ibm-775_P100-1996	IBM775 cp775 csPC775Baltic	ibm-775_P100-1996 ibm-775 IBM775 cp775 csPC775Baltic windows-775 775
ibm-850_P100-1995	IBM850 cp850 850 csPC850Multilingual	ibm-850_P100-1995 ibm-850 IBM850 cp850 850 csPC850Multilingual windows-850
ibm-851_P100-1995	IBM851 cp851 851 csPC851	ibm-851_P100-1995 ibm-851 IBM851 cp851 851 csPC851
ibm-852_P100-1995	IBM852 cp852 852 csPCp852	ibm-852_P100-1995 ibm-852 IBM852 cp852 852 csPCp852 windows-852
ibm-855_P100-1995	IBM855 cp855 855 csIBM855	ibm-855_P100-1995 ibm-855 IBM855 cp855 855 csIBM855 csPCp855 windows-855
ibm-856_P100-1995		ibm-856_P100-1995 ibm-856 IBM856 cp856 856 x-IBM856
ibm-857_P100-1995	IBM857 cp857 857 csIBM857	ibm-857_P100-1995 ibm-857 IBM857 cp857 857 csIBM857 windows-857

ibm-858_P100-1997	IBM00858 CCSID00858 CP00858 PC-Multilingual-850+euro	ibm-858_P100-1997 ibm-858 IBM00858 CCSID00858 CP00858 PC-Multilingual-850+euro cp858 windows-858
ibm-860_P100-1995	IBM860 cp860 860 csIBM860	ibm-860_P100-1995 ibm-860 IBM860 cp860 860 csIBM860
ibm-861_P100-1995	IBM861 cp861 861 cp-is csIBM861	ibm-861_P100-1995 ibm-861 IBM861 cp861 861 cp-is csIBM861 windows-861
ibm-862_P100-1995	IBM862 cp862 862 csPC862LatinHebrew	ibm-862_P100-1995 ibm-862 IBM862 cp862 862 csPC862LatinHebrew DOS-862 windows-862
ibm-863_P100-1995	IBM863 cp863 863 csIBM863	ibm-863_P100-1995 ibm-863 IBM863 cp863 863 csIBM863
ibm-864_X110-1999	IBM864 cp864 csIBM864	ibm-864_X110-1999 ibm-864 IBM864 cp864 csIBM864
ibm-865_P100-1995	IBM865 cp865 865 csIBM865	ibm-865_P100-1995 ibm-865 IBM865 cp865 865 csIBM865
ibm-866_P100-1995	IBM866 cp866 866 csIBM866	ibm-866_P100-1995 ibm-866 IBM866 cp866 866 csIBM866 windows-866
ibm-867_P100-1998		ibm-867_P100-1998 ibm-867 x-IBM867
ibm-868_P100-1995	IBM868 CP868 csIBM868 cp-ar	ibm-868_P100-1995 ibm-868 IBM868 CP868 868 csIBM868 cp-ar

ibm-869_P100-1995	IBM869 cp869 869 cp-gr csIBM869	ibm-869_P100-1995 ibm-869 IBM869 cp869 869 cp-gr csIBM869 windows-869
ibm-878_P100-1996	KOI8-R csKOI8R	ibm-878_P100-1996 ibm-878 KOI8-R koi8 csKOI8R windows-20866 cp878
ibm-901_P100-1999		ibm-901_P100-1999 ibm-901
ibm-902_P100-1999		ibm-902_P100-1999 ibm-902
ibm-922_P100-1999		ibm-922_P100-1999 ibm-922 IBM922 cp922 922 x-IBM922
ibm-1168_P100-2002	KOI8-U	ibm-1168_P100-2002 ibm-1168 KOI8-U windows-21866
ibm-4909_P100-1999		ibm-4909_P100-1999 ibm-4909
ibm-5346_P100-1998	windows-1250	ibm-5346_P100-1998 ibm-5346 windows-1250 cp1250
ibm-5347_P100-1998	windows-1251	ibm-5347_P100-1998 ibm-5347 windows-1251 cp1251 ANSI1251
ibm-5348_P100-1997	windows-1252	ibm-5348_P100-1997 ibm-5348 windows-1252 cp1252
ibm-5349_P100-1998	windows-1253	ibm-5349_P100-1998 ibm-5349 windows-1253 cp1253
ibm-5350_P100-1998	windows-1254	ibm-5350_P100-1998 ibm-5350 windows-1254 cp1254
ibm-9447_P100-2002	windows-1255	ibm-9447_P100-2002 ibm-9447 windows-1255 cp1255
ibm-9448_X100-2005	windows-1256	ibm-9448_X100-2005 ibm-9448 windows-1256 cp1256 x-windows-1256S
ibm-9449_P100-2002	windows-1257	ibm-9449_P100-2002 ibm-9449 windows-1257 cp1257
ibm-5354_P100-1998	windows-1258	ibm-5354_P100-1998 ibm-5354 windows-1258 cp1258
ibm-1250_P100-1995		ibm-1250_P100-1995 ibm-1250 windows-1250
ibm-1251_P100-1995		ibm-1251_P100-1995 ibm-1251 windows-1251

ibm-1252_P100-2000		ibm-1252_P100-2000 ibm-1252 windows-1252
ibm-1253_P100-1995		ibm-1253_P100-1995 ibm-1253 windows-1253
ibm-1254_P100-1995		ibm-1254_P100-1995 ibm-1254 windows-1254
ibm-1255_P100-1995		ibm-1255_P100-1995 ibm-1255
ibm-5351_P100-1998		ibm-5351_P100-1998 ibm-5351 windows-1255
ibm-1256_P110-1997		ibm-1256_P110-1997 ibm-1256
ibm-5352_P100-1998		ibm-5352_P100-1998 ibm-5352 windows-1256
ibm-1257_P100-1995		ibm-1257_P100-1995 ibm-1257
ibm-5353_P100-1998		ibm-5353_P100-1998 ibm-5353 windows-1257
ibm-1258_P100-1997		ibm-1258_P100-1997 ibm-1258 windows-1258
macos-0_2-10.2	macintosh mac csMacintosh	macos-0_2-10.2 macintosh mac csMacintosh windows-10000 macroman x-macroman
macos-6_2-10.4		macos-6_2-10.4 x-mac-greek windows-10006 macgr x-MacGreek
macos-7_3-10.2		macos-7_3-10.2 x-mac-cyrillic windows-10007 mac-cyrillic maccy x-MacCyrillic x-MacUkraine
macos-29-10.2		macos-29-10.2 x-mac-centraleurroman windows-10029 x-mac-ce macce maccentraleurope x-MacCentralEurope
macos-35-10.2		macos-35-10.2 x-mac-turkish windows-10081 mactr x-MacTurkish
ibm-1051_P100-1995	hp-roman8 roman8 r8 csHPRoman8	ibm-1051_P100-1995 ibm-1051 hp-roman8 roman8 r8 csHPRoman8

ibm-1276_P100-1995	Adobe-Standard-Encoding csAdobeStandardEncoding	ibm-1276_P100-1995 ibm-1276 Adobe-Standard-Encoding csAdobeStandardEncoding
ibm-1006_P100-1995		ibm-1006_P100-1995 ibm-1006 IBM1006 cp1006 1006 x-IBM1006
ibm-1098_P100-1995		ibm-1098_P100-1995 ibm-1098 IBM1098 cp1098 1098 x-IBM1098
ibm-1124_P100-1996		ibm-1124_P100-1996 ibm-1124 cp1124 1124 x-IBM1124
ibm-1125_P100-1997		ibm-1125_P100-1997 ibm-1125 cp1125
ibm-1129_P100-1997		ibm-1129_P100-1997 ibm-1129
ibm-1131_P100-1997		ibm-1131_P100-1997 ibm-1131 cp1131
ibm-1133_P100-1997		ibm-1133_P100-1997 ibm-1133
ibm-37_P100-1995	IBM037 ebcdic-cp-us ebcdic-cp-ca ebcdic-cp-wt ebcdic-cp-nl csIBM037	ibm-37_P100-1995 ibm-37 IBM037 ibm-037 ebcdic-cp-us ebcdic-cp-ca ebcdic-cp-wt ebcdic-cp-nl csIBM037 cp037 037 cpibm37 cp37
ibm-273_P100-1995	IBM273 CP273 csIBM273	ibm-273_P100-1995 ibm-273 IBM273 CP273 csIBM273 ebcdic-de 273
ibm-277_P100-1995	IBM277 EBCDIC-CP-DK EBCDIC-CP-NO csIBM277	ibm-277_P100-1995 ibm-277 IBM277 cp277 EBCDIC-CP-DK EBCDIC-CP-NO csIBM277 ebcdic-dk 277
ibm-278_P100-1995	IBM278 ebcdic-cp-fi ebcdic-cp-se csIBM278	ibm-278_P100-1995 ibm-278 IBM278 cp278 ebcdic-cp-fi ebcdic-cp-se csIBM278 ebcdic-sv 278
ibm-280_P100-1995	IBM280 CP280 ebcdic-cp-it csIBM280	ibm-280_P100-1995 ibm-280 IBM280 CP280 ebcdic-cp-it csIBM280 280
ibm-284_P100-1995	IBM284 CP284 ebcdic-cp-es csIBM284	ibm-284_P100-1995 ibm-284 IBM284 CP284 ebcdic-cp-es csIBM284 cpibm284 284

ibm-285_P100-1995	IBM285 CP285 ebcdic-cp-gb csIBM285	ibm-285_P100-1995 ibm-285 IBM285 CP285 ebcdic-cp-gb csIBM285 cpibm285 ebcdic-gb 285
ibm-290_P100-1995	IBM290 cp290 EBCDIC-JP-kana csIBM290	ibm-290_P100-1995 ibm-290 IBM290 cp290 EBCDIC-JP-kana csIBM290
ibm-297_P100-1995	IBM297 cp297 ebcdic-cp-fr csIBM297	ibm-297_P100-1995 ibm-297 IBM297 cp297 ebcdic-cp-fr csIBM297 cpibm297 297
ibm-420_X120-1999	IBM420 cp420 ebcdic-cp-ar1 csIBM420	ibm-420_X120-1999 ibm-420 IBM420 cp420 ebcdic-cp-ar1 csIBM420 420
ibm-424_P100-1995	IBM424 cp424 ebcdic-cp-he csIBM424	ibm-424_P100-1995 ibm-424 IBM424 cp424 ebcdic-cp-he csIBM424 424
ibm-500_P100-1995	IBM500 CP500 ebcdic-cp-be csIBM500 ebcdic-cp-ch	ibm-500_P100-1995 ibm-500 IBM500 CP500 ebcdic-cp-be csIBM500 ebcdic-cp-ch 500
ibm-803_P100-1999		ibm-803_P100-1999 ibm-803 cp803
ibm-838_P100-1995	IBM-Thai csIBMThai	ibm-838_P100-1995 ibm-838 IBM838 IBM-Thai csIBMThai cp838 838 ibm-9030
ibm-870_P100-1995	IBM870 CP870 ebcdic-cp-roece ebcdic-cp-yu csIBM870	ibm-870_P100-1995 ibm-870 IBM870 CP870 ebcdic-cp-roece ebcdic-cp-yu csIBM870
ibm-871_P100-1995	IBM871 ebcdic-cp-is csIBM871 CP871	ibm-871_P100-1995 ibm-871 IBM871 ebcdic-cp-is csIBM871 CP871 ebcdic-is 871
ibm-875_P100-1995		ibm-875_P100-1995 ibm-875 IBM875 cp875 875 x-IBM875

ibm-918_P100-1995	IBM918 CP918 ebcdic-cp-ar2 csIBM918	ibm-918_P100-1995 ibm-918 IBM918 CP918 ebcdic-cp-ar2 csIBM918
ibm-1025_P100-1995		ibm-1025_P100-1995 ibm-1025 cp1025 1025 x-IBM1025
ibm-1026_P100-1995	IBM1026 CP1026 csIBM1026	ibm-1026_P100-1995 ibm-1026 IBM1026 CP1026 csIBM1026 1026
ibm-1047_P100-1995	IBM1047	ibm-1047_P100-1995 ibm-1047 IBM1047 cp1047 1047
ibm-1097_P100-1995		ibm-1097_P100-1995 ibm-1097 cp1097 1097 x-IBM1097
ibm-1112_P100-1995		ibm-1112_P100-1995 ibm-1112 cp1112 1112 x-IBM1112
ibm-1122_P100-1999		ibm-1122_P100-1999 ibm-1122 cp1122 1122 x-IBM1122
ibm-1123_P100-1995		ibm-1123_P100-1995 ibm-1123 cp1123 1123 x-IBM1123
ibm-1130_P100-1997		ibm-1130_P100-1997 ibm-1130
ibm-1132_P100-1998		ibm-1132_P100-1998 ibm-1132
ibm-1137_P100-1999		ibm-1137_P100-1999 ibm-1137
ibm-4517_P100-2005		ibm-4517_P100-2005 ibm-4517
ibm-1140_P100-1997	IBM01140 CCSID01140 CP01140 ebcdic-us-37+euro	ibm-1140_P100-1997 ibm-1140 IBM01140 CCSID01140 CP01140 cp1140 ebcdic-us-37+euro
ibm-1141_P100-1997	IBM01141 CCSID01141 CP01141 ebcdic-de-273+euro	ibm-1141_P100-1997 ibm-1141 IBM01141 CCSID01141 CP01141 cp1141 ebcdic-de-273+euro
ibm-1142_P100-1997	IBM01142 CCSID01142 CP01142 ebcdic-dk-277+euro ebcdic-no-277+euro	ibm-1142_P100-1997 ibm-1142 IBM01142 CCSID01142 CP01142 cp1142 ebcdic-dk-277+euro ebcdic-no-277+euro

ibm-1143_P100-1997	IBM01143 CCSID01143 CP01143 ebcdic-fi-278+euro ebcdic-se-278+euro	ibm-1143_P100-1997 ibm-1143 IBM01143 CCSID01143 CP01143 cp1143 ebcdic-fi-278+euro ebcdic-se-278+euro
ibm-1144_P100-1997	IBM01144 CCSID01144 CP01144 ebcdic-it-280+euro	ibm-1144_P100-1997 ibm-1144 IBM01144 CCSID01144 CP01144 cp1144 ebcdic-it-280+euro
ibm-1145_P100-1997	IBM01145 CCSID01145 CP01145 ebcdic-es-284+euro	ibm-1145_P100-1997 ibm-1145 IBM01145 CCSID01145 CP01145 cp1145 ebcdic-es-284+euro
ibm-1146_P100-1997	IBM01146 CCSID01146 CP01146 ebcdic-gb-285+euro	ibm-1146_P100-1997 ibm-1146 IBM01146 CCSID01146 CP01146 cp1146 ebcdic-gb-285+euro
ibm-1147_P100-1997	IBM01147 CCSID01147 CP01147 ebcdic-fr-297+euro	ibm-1147_P100-1997 ibm-1147 IBM01147 CCSID01147 CP01147 cp1147 ebcdic-fr-297+euro
ibm-1148_P100-1997	IBM01148 CCSID01148 CP01148 ebcdic-international-500+euro	ibm-1148_P100-1997 ibm-1148 IBM01148 CCSID01148 CP01148 cp1148 ebcdic-international-500+euro
ibm-1149_P100-1997	IBM01149 CCSID01149 CP01149 ebcdic-is-871+euro	ibm-1149_P100-1997 ibm-1149 IBM01149 CCSID01149 CP01149 cp1149 ebcdic-is-871+euro
ibm-1153_P100-1999		ibm-1153_P100-1999 ibm-1153 IBM1153 x-IBM1153
ibm-1154_P100-1999		ibm-1154_P100-1999 ibm-1154
ibm-1155_P100-1999		ibm-1155_P100-1999 ibm-1155
ibm-1156_P100-1999		ibm-1156_P100-1999 ibm-1156
ibm-1157_P100-1999		ibm-1157_P100-1999 ibm-1157
ibm-1158_P100-1999		ibm-1158_P100-1999 ibm-1158
ibm-1160_P100-1999		ibm-1160_P100-1999 ibm-1160
ibm-1164_P100-1999		ibm-1164_P100-1999 ibm-1164
ibm-5123_P100-1999		ibm-5123_P100-1999 ibm-5123

ibm-8482_P100-1999		ibm-8482_P100-1999 ibm-8482
ibm-4899_P100-1998		ibm-4899_P100-1998 ibm-4899
ibm-4971_P100-1999		ibm-4971_P100-1999 ibm-4971
ibm-9067_X100-2005		ibm-9067_X100-2005 ibm-9067
ibm-12712_P100-1998		ibm-12712_P100-1998 ibm-12712 ebcdic-he
ibm-16804_X110-1999		ibm-16804_X110-1999 ibm-16804 ebcdic-ar
ibm-37_P100-1995,swaplfnl		ibm-37_P100-1995,swaplfnl ibm-37-s390
ibm-1047_P100-1995,swaplfnl		ibm-1047_P100-1995,swaplfnl ibm-1047-s390 IBM1047_LF
ibm-1140_P100-1997,swaplfnl		ibm-1140_P100-1997,swaplfnl ibm-1140-s390
ibm-1141_P100-1997,swaplfnl		ibm-1141_P100-1997,swaplfnl ibm-1141-s390 IBM1141_LF
ibm-1142_P100-1997,swaplfnl		ibm-1142_P100-1997,swaplfnl ibm-1142-s390
ibm-1143_P100-1997,swaplfnl		ibm-1143_P100-1997,swaplfnl ibm-1143-s390
ibm-1144_P100-1997,swaplfnl		ibm-1144_P100-1997,swaplfnl ibm-1144-s390
ibm-1145_P100-1997,swaplfnl		ibm-1145_P100-1997,swaplfnl ibm-1145-s390
ibm-1146_P100-1997,swaplfnl		ibm-1146_P100-1997,swaplfnl ibm-1146-s390
ibm-1147_P100-1997,swaplfnl		ibm-1147_P100-1997,swaplfnl ibm-1147-s390
ibm-1148_P100-1997,swaplfnl		ibm-1148_P100-1997,swaplfnl ibm-1148-s390
ibm-1149_P100-1997,swaplfnl		ibm-1149_P100-1997,swaplfnl ibm-1149-s390
ibm-1153_P100-1999,swaplfnl		ibm-1153_P100-1999,swaplfnl ibm-1153-s390
ibm-12712_P100-1998,swaplfnl		ibm-12712_P100-1998,swaplfnl ibm-12712-s390

ibm-16804_X110-1999,swaplfnl		ibm-16804_X110-1999,swaplfnl ibm-16804-s390
ebcdic-xml-us		ebcdic-xml-us

Double-byte character sets (DBCS)

Canonical Name	IANA	All Aliases
ibm-5478_P100-1995	GB_2312-80 chinese iso-ir-58 csISO58GB231280	ibm-5478_P100-1995 ibm-5478 GB_2312-80 chinese iso-ir-58 csISO58GB231280 gb2312-1980 GB2312.1980-0
ibm-971_P100-1995		ibm-971_P100-1995 ibm-971 ibm-971_VPUA x-IBM971
ibm-16684_P110-2003		ibm-16684_P110-2003 ibm-16684 ibm-20780

ISO_2022

Canonical Name	IANA	All Aliases
ISO_2022,locale=ja,version=0	ISO-2022-JP csISO2022JP	ISO_2022,locale=ja,version=0 ISO-2022-JP csISO2022JP x-windows-iso2022jp x-windows-50220
ISO_2022,locale=ja,version=1	JIS_Encoding csJISEncoding	ISO_2022,locale=ja,version=1 ISO-2022-JP-1 JIS_Encoding csJISEncoding ibm-5054 JIS x-windows-50221
ISO_2022,locale=ja,version=2	ISO-2022-JP-2 csISO2022JP2	ISO_2022,locale=ja,version=2 ISO-2022-JP-2 csISO2022JP2
ISO_2022,locale=ja,version=3		ISO_2022,locale=ja,version=3 JIS7
ISO_2022,locale=ja,version=4		ISO_2022,locale=ja,version=4 JIS8
ISO_2022,locale=ko,version=0	ISO-2022-KR csISO2022KR	ISO_2022,locale=ko,version=0 ISO-2022-KR csISO2022KR
ISO_2022,locale=ko,version=1		ISO_2022,locale=ko,version=1 ibm-25546

ISO_2022,locale=zh,version=0	ISO-2022-CN	ISO_2022,locale=zh,version=0 ISO-2022-CN csISO2022CN x-ISO-2022-CN-GB
ISO_2022,locale=zh,version=1	ISO-2022-CN-EXT	ISO_2022,locale=zh,version=1 ISO-2022-CN-EXT
ISO_2022,locale=zh,version=2		ISO_2022,locale=zh,version=2 ISO-2022-CN-CNS x-ISO-2022-CN-CNS

HZ

Canonical Name	IANA	All Aliases
HZ	HZ-GB-2312	HZ HZ-GB-2312

Indian Script Code for Information Interchange (ISCII)

Canonical Name	IANA	All Aliases
ISCII,version=0		ISCII,version=0 x-ISCII91 x-iscii-de windows-57002 iscii-dev ibm-4902
ISCII,version=1		ISCII,version=1 x-iscii-be windows-57003 iscii-bng windows-57006 x-iscii-as
ISCII,version=2		ISCII,version=2 x-iscii-pa windows-57011 iscii-gur
ISCII,version=3		ISCII,version=3 x-iscii-gu windows-57010 iscii-guj
ISCII,version=4		ISCII,version=4 x-iscii-or windows-57007 iscii-ori
ISCII,version=5		ISCII,version=5 x-iscii-ta windows-57004 iscii-tml
ISCII,version=6		ISCII,version=6 x-iscii-te windows-57005 iscii-tlg
ISCII,version=7		ISCII,version=7 x-iscii-ka windows-57008 iscii-knd
ISCII,version=8		ISCII,version=8 x-iscii-ma windows-57009 iscii-mlm

LMBCS_1

Canonical Name	IANA	All Aliases
LMBCS-1		LMBCS-1 lmbcs ibm-65025

EBCDIC_STATEFUL

Canonical Name	IANA	All Aliases
ibm-930_P120-1999		ibm-930_P120-1999 ibm-930 ibm-5026 IBM930 cp930 930 x-IBM930 x-IBM930A
ibm-933_P110-1995		ibm-933_P110-1995 ibm-933 cp933 933 x-IBM933
ibm-935_P110-1999		ibm-935_P110-1999 ibm-935 cp935 935 x-IBM935
ibm-937_P110-1999		ibm-937_P110-1999 ibm-937 cp937 937 x-IBM937
ibm-939_P120-1999		ibm-939_P120-1999 ibm-939 ibm-931 ibm-5035 IBM939 cp939 939 x-IBM939 x-IBM939A
ibm-1364_P110-2007		ibm-1364_P110-2007 ibm-1364 x-IBM1364
ibm-1371_P100-1999		ibm-1371_P100-1999 ibm-1371 x-IBM1371
ibm-1388_P103-2001		ibm-1388_P103-2001 ibm-1388 ibm-9580 x-IBM1388
ibm-1390_P110-2003		ibm-1390_P110-2003 ibm-1390 x-IBM1390
ibm-1399_P110-2003		ibm-1399_P110-2003 ibm-1399 x-IBM1399

Appendix D: Problem Determination

With each implementation, the problem determination procedures vary slightly. This appendix is focused towards help desk personnel or a designated CA XCOM Data Transport troubleshooter with knowledge of the following:

- The operational characteristics of CA XCOM Data Transport for UNIX and Linux
- For transfers that use SNA, some knowledge of the structure and components of an SNA networking environment, including an understanding of the nature of SNA LUs, PUs, sessions, and conversations

This section contains the following topics:

Where to Find Information (see page 617)
General Procedures (see page 618)
Test Procedure (see page 619)
Problem Determination Worksheet (see page 620)
Contact Technical Support (see page 627)
Product Releases and Maintenance (see page 627)

Where to Find Information

CA XCOM Data Transport stores messages about the success and failure of all transactions in the following log file.

/usr/spool/xcom/xcom.log, except for the following:

- HP-UX PA RISC and HP-UX IA64 systems: /var/spool/xcom/xcom.log
- HP TRU64 (OSF Digital UNIX) system: /var/opt/xcom/xcom.log

When you have a problem, you should review the messages displayed on the screen as well as those recorded in this file. In many cases, you will be able to determine the cause of the problem from the messages. For more information about the log file, see Setting Up Log Files in the chapter "The Command Line Interface." For more information about CA XCOM Data Transport for UNIX and Linux messages, see the appendix "Messages."

General Procedures

This section contains a general test procedure to use for documenting problems.

Most problems encountered by users are the result of the following:

- Improper configuration
- Incorrectly performed procedures
- A faulty environment

The purpose of the procedures and worksheet in this Appendix is to help you to isolate the cause of problems and to correct them in an efficient and timely manner or to gather enough information for CA Technical Support to help you as quickly as possible.

Test Procedure

To be effective, you must follow these procedures in order and document the results of every test carefully. It is good practice to run through the tests at least twice to ensure that the results are consistent.

Follow these steps:

1. Document the precise actions that were performed immediately prior to the appearance of the problem.

   Note: The description of the actions performed should be detailed enough to allow for you to recreate the problem at some future time if necessary.

2. Document the symptoms of the problem. This includes:

   - Error messages
   - Unexpected events
   - Other details relevant to the malfunction

   Wherever possible, record any error messages that appear on the screen.

3. If a data file has been corrupted, save the data file in a safe place for later analysis by CA Technical Support. Any information that appears even remotely relevant may prove vital in achieving a quick resolution to the problem.

4. Retry the operation, isolating and eliminating as many extraneous factors as possible.

5. Use the following table to determine your next step.

Did the Problem Recur?	Then...
No	Record as much as you can from the problem's first occurrence (log files, and so forth) so that if you ever see it again, you will have additional evidence.
Yes	Continue with Step 6.

6. Is an error message issued?

   Yes

   Look up the error message in the appropriate CA XCOM Data Transport manual and follow the instructions.

No

If no error message was issued and the operation does not complete (for example, hangs), the problem may be related to a system malfunction (for example, remote system response time problems, and so forth).

Leave the function in the "hung state" for at least 10 minutes. If the problem persists, contact a person who can verify the current status of the system and backbone network. If the system or the network is not operating normally, retry the transfer at some later time.

7. Run a trace to pinpoint and trace problems in your communication lines.

Problem Determination Worksheet

This section contains worksheets for use in determining problems. When you call CA Technical Support, it is helpful to have the following information available so they can quickly and efficiently narrow down the cause of the problem. Use the following sheets to gather the information Customer Support will need.

General Information

Use the following table to provide general customer information.

Information	Your Response
Date of Incident	
Incident Number (to be supplied by CA Technical Support)	
Customer Name	
Site ID	
Customer Address	
Customer Telephone Number	
Customer email Address	
CA XCOM Data Transport Platform, Version, Release, and Maintenance	
Customer Support Contact (to be supplied by CA Technical Support)	

Environment Information

Use the following table to inventory information about your environment. Use additional sheets as necessary.

Information Type	Initiating System	Receiving System	(Receiving System)	Intermediary System
Platform				
Operating System				
Communications Subsystem				
Security Subsystem				
For SNA/APPC (Version/ Release/ Maintenance)				
Have there been any hardware or software changes (for example, hardware upgrades, operating system, CA XCOM Data Transport, or communications type changes)				

Transfer Type

Please check the box that applies to the file transfer:

Send File	Send Job	Send Report	Retrieve File

Problem Description

What is the customer attempting to perform or accomplish?

Provide a detailed problem description:

Use the following table to answer the questions below:

Question	YES	NO
Is the problem reproducible?		
Is this an immediate transfer (Foreground, TYPE=EXECUTE) or a scheduled transfer (Background, TYPE=SCHEDULE)?		
Has there been any hardware or software changes (for example, hardware upgrades, operating system, CA XCOM Data Transport, or communications type changes)		

If YES, please explain in detail.

Problem History

Use the following section to provide a description of the problem history.

	Yes	No
Is this the first occurrence of the problem?		

If No, please explain.

Error Messages

Please provide a list of error messages; include the message ID and all of the text.

Network Configuration Diagram

Network Configuration Diagram (How are each of the partners connected?)

Environmental Information Inventory

Use the following table to inventory information about your environment. (Fill in the boxes of platforms involved in transfer)

Platform	Operating System	Communications Subsystem	Security Subsystem	APPC Version/ Release/ Maintenance
z/OS				
zVM				
zVSE				
AS/400				
Open VMS				
Stratus				
HP Non Stop				
HP-UX PA RISC				
HP-UX IA				
Linux				
zLinux				

AIX
Solaris
Windows 2003, 2008
Windows XP, Vista, 7
NetWare
Other

Initiating Platform _____

Check the column that applies to the file transfer:

	Send File	Send Job	Send Report	Receive File
This is a				

Documentation

The following is a starting point for various error conditions on various platforms. More documentation may be required to resolve the error condition:

Conditions	Platforms	Documentation
Abend	z/OS	Full Dump JES2 or JES3 log Xcom log, Batch job stream, or ISPF parameters
	VM	GCS Dump Virtual machine Console log Dest Table Default File CA XCOM Data Transport parameter file
	VSE	Partition DUMP,POWER log CA XCOM Data Transport log Batch Job stream Dest and Default Tables
	HP Non Stop	Abend message What was being attempted Transfer parameters VPROC of XCOM62

Conditions	Platforms	Documentation
Error in Transfer	z/OS	JES2 or JES3 log CA XCOM Data Transport log Batch Job Stream or ISPF parameters Xcomcntl and default file VTAM Buffer/IO trace
	zVM	VTAM Buffer/IO trace Virtual Machine Console log Rexx exec or Clist Dest and default file
	zVSE	POWER Log CA XCOM Data Transport log Batch Job Stream VTAM Buffer/IO trace Dest and default tables
	AS/400	Error message received Sense codes Job logs
	HP Non Stop	SNA Trace CA XCOM Data Transport Config Parameters used for transfer Local initiator log Remote initiator log
	Open VMS	SNA Trace CA XCOM Data Transport Config Parameters used for transfer Local initiator log Remote initiator log
	Stratus	Syserr_log messages Network definitions on both sides of the connection Monitor_SDLC log Status of station, link, line, LU, and session

Conditions	Platforms	Documentation
	NetWare	Syserr_log messages Network definitions on both sides of the connection Monitor_SDLC log Status of station, link, line, LU, and session
	Other	

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Product Releases and Maintenance

Customers are requested to operate only under currently supported releases of the product.

Customers with current maintenance agreements also receive ongoing maintenance. When a new release of the system is available, a notice is sent to all current customers.

Index

#

#!ENCRYPT • 246

A

AGE_TIME • 45, 354
ALLOCATION_TYPE • 167, 355
allocation_type, API • 261
API • 81
    API example • 329
    API return values • 321
    API, accessing the queue • 81
apitest.c file, API • 329
ASCII/EBCDIC conversion • 228
ATOE_FILENAME • 45, 355
AUTH_TYPE • 46

B

BLKSIZE • 168, 356
blksize, API • 262
Brixton
    Brixton, API • 315

C

C structures, API • 261
CA certificate for SSL • 343
    CA certificate for SSL, expiration • 342
CARRIAGE_CONTROL_CHARACTERS • 186, 359
carriage_control_characters, API • 263
CARRIAGE_FLAG • 229, 360
carriage_flag, API • 264
changing encrypted parameter values • 247
character conversion • 228
    ASCII to EBCDIC • 231
    EBCDIC to ASCII • 231
    hexadecimal numbers • 232
checkpoint and restart • 255
    chxcom62 command • 148, 149, 255
    performance issues • 255
    xcomtcp command • 148, 149, 255
checkpoint count, API • 264
checkpoint parameters
    CHECKPOINT_COUNT • 256
    NUMBER_OF_RETRIES • 256
CHECKPOINT_COUNT • 256, 361
chxcom62 command • 148, 149, 255
    options • 151
    SNA/APPC protocol example • 153
    syntax • 150
CLASS • 186, 362
class, API • 297
CLEANLOG • 84
client certificate for SSL • 344
CODE_FLAG • 230, 364
code_flag, API • 265
CODETABL • 365
codetabl, API • 266
communications parameters
    REMOTE_SYSTEM • 438
    REMOTE_SYSTEM_RF • 439
    REMOTE_SYSTEM_SJ • 439
    REMOTE_SYSTEM_SR • 440
    setting • 158
    VERSION • 160
COMPRESS • 159, 365
compress, API • 267
configuration files
    configxcom.glb • 45, 142
    creating • 143
    using • 142
    xcom.cnf • 142, 158
    xcom.ses • 80
configuring CA XCOM Data Transport
    SSL client • 346
    SSL server • 345
configxcom.glb • 45, 142
connection_profile • 80
CONTROL • 367
control block structure, API • 322
CONVERT_CLASSES • 48, 367
convert_classes, API • 269
COPIES • 187, 367
copies, API • 269
CREATE_DIR • 162, 367
custom character sets • 230
customizing notification • 97

D

daemon • 141
    daemon process • 43

data transfer parameters • 365
CODETABL • 365
COMPRESS • 159
MAXRECLEN • 160
TRUNCATION • 160
DATACLAS • 171, 369
dataclas, API • 271
DEBUG_FLAG • 369
debug_flag, API • 271, 369
debugging scripts • 40, 88
DEN • 177
den, API • 271
DESTINATION • 187
destination, API • 272
DESTINATION_TYPE • 373
directory considerations • 40
DISPOSITION • 187, 374
disposition, API • 272
DOMAIN • 244, 374
domain name, fully qualified • 154
domain, API • 272
DSNTYPE • 172, 375
dsntype, API • 273

E

EAR_CIPHER • 376
EAR_DIGEST • 377
EAR_HASH • 378
EAR_KEY • 378
EBCDIC conversion • 228
encrypted parameter values • 247
EOL_CLASSES • 52, 379
eol_classes, API • 278
error messages, system identifier • 484
ETOA_FILENAME • 53, 380
EXPDT • 178, 380
expdt, API • 278
EXPIRATION_TIME • 53, 380

F

FCB • 381
fcb, API • 278
FCB, Send Report parameter • 188
file conversion
default character sets • 232
file custom character sets • 230
file conversion parameters
CARRIAGE_FLAG • 229
CODE_FLAG • 230
file type conversion • 228
FILE_OPTION • 163, 382
file_option, API • 279
FILE_OPTION_RF • 182, 383
file_type, API • 279
FIPS_MODE • 53
FORM • 188
form, API • 280
fully qualified domain name • 154

G

global parameters
AGE_TIME • 45
ATOE_FILENAME • 45
CONVERT_CLASSES • 48
EOL_CLASSES • 52
ETOA_FILENAME • 53
EXPIRATION_TIME • 53
INTERNAL_CONVERSION_TABLES • 55
MAX_QUEUE_ENTRIES • 57
MAX_REMOTE_TCP • 57
MAX_SESSIONS_ENTRIES • 57
METACODE_CLASSES • 57
PROTOCOL • 58
QUEUE_PATH • 59
REMOVE_TRAIL_BLANKS • 59
SAVE_PASSWORD_IN_CNF • 59
SESSIONS_FILE • 60
SHELL_CMD • 60
SOCK_DELAY • 61
SOCK_RCV_BUF_SIZE • 61
SOCK_SEND_BUF_SIZE • 61
STAT_FREQUENCY • 62
TCP_CMD • 63
TEMPDIR • 63
TP_CMD • 63
TRACE_PATH • 63
TXPI_BUF_SIZE • 71
TXPI_SEND_CHECK_FREQ • 71
TXPI_TERM_TIMEOUT • 72
UMASK • 72
XCOM_PASSWORD • 74
XCOM_TRUSTED_OVR • 75
XCOM_USERID, global parameter • 75
XCOMPRE_LOCAL • 77
XENDCMD • 77
XLPCMD • 78

XNOTIFYCMD • 78
XPPCMD • 78
XPRECMD • 79
XTRACE • 79

H

hold, API • 280
HOLDFLAG • 188, 390

I

IBM mainframe file creation parameters
ALLOCATION_TYPE • 167
BLKSIZE • 168
DATACLAS • 171
DSNTYPE • 172
LRECL • 173
MGMTCLAS • 173
NUM_OF_DIR_BLOCKS • 174
PRIMARY_ALLOC • 174
RECORD_FORMAT • 174
SECONDARY_ALLOC • 175
SECURE_SOCKET • 446
STORCLAS • 176
UNIT • 176
UNIT_RF • 184
VOLUME • 177
VOLUME_RF • 185
IBM mainframe, creating a file on • 166
indirect transfer • 242
INTERNAL_CONVERSION_TABLES • 55, 391

L

LABEL • 178, 392
label, API • 281
LABELNUM • 179, 393
labelnum, API • 281
lclntfy, API • 282
LCLNTFYL • 251, 393
LEAR_CIPHER • 394
LEAR_DIGEST • 395
LEAR_HASH • 396
LEAR_KEY • 396
link libraries, API • 321
LOCAL_FILE • 164
local_file, API • 289
LOCAL_FILE_RF • 183
local_file_rf, API • 290
LOCAL_FILE_SJ • 193
local_file_sj, API • 290
LOCAL_FILE_SR • 189
local_file_sr, API • 290
LOCAL_NOTIFY • 250
LOCAL_RECEIVE • 321
LOCAL_SEND • 321
locally initiated transfers • 247
localnotify, API • 291
log files • 226
logging • 226
logging parameter, XLOGFILE • 227
LRECL • 173, 404
lrecl, API • 291

M

machine code characters, reports • 106
mailx • 97
mainframe file creation • 166
managing queue • 81
MAX_QUEUE_ENTRIES • 57, 405
MAX_REMOTE_TCP • 405
MAX_REMOTE_TCP • 57
MAX_SESSIONS_ENTRIES • 57, 406
MAXRECLEN • 160, 406
maxreclen, API • 291
member names, API • 261
METACODE_CLASSES • 57, 408
metacode_classes, API • 293
MGMTCLAS • 173, 408
mgmtclas, API • 293
mode • 341
multiple transfers • 367

N

NEWXFER • 367
notification parameters
LCLNTFYL • 251
LOCAL_NOTIFY • 250
NOTIFY_NAME • 252
NOTIFY_TERM • 252
NOTIFYL • 251
NOTIFYR • 253
RMLNTFYL • 254
using • 250
notification script, xcomntfy • 98
NOTIFY_NAME • 252, 409
notify_name, API • 294
NOTIFY_TERM • 252, 409

notify_term, API • 295
NOTIFYL • 251, 410
notifyl, API • 294
NOTIFYR • 253, 411
notifyr, API • 295
NUM_OF_DIR_BLOCKS • 174, 411
num_of_dir_blocks, API • 296
NUMBER_OF_RETRIES • 256, 412
number_of_retries, API • 296

O

OEDATE • 412
OETIME • 413
OFILETYPE • 415
OFLMAX • 416
OFLMIN • 417
OINIT • 418
OLMSG • 420
OREQ • 422
OSDATE • 423
OSTIME • 424
OTYPE • 426
OTYPEREQ • 426

P

PAM_PATH • 58
parameter format • 41, 352
parameters • 40
    encrypting • 245
PASSWORD • 244, 428
password, API • 296
path considerations • 40
permissions • 43
PORT • 428
port, API • 297
post processing parameters • 88
post transfer script • 91
pre-allocation exit, xcompre • 85
PRIMARY_ALLOC • 174, 428
primary_alloc, API • 296
printing processes • 95
printing script, xcomlp • 96
PRIORITY • 248, 428
problem tracing • 224
PROTOCOL • 58, 429
protocol, API • 297
protocols • 40
PU 2.0 devices and store and forward • 242

Q

QUEUE • 248, 430
queue corruption • 44
queue manager, xcomqm • 81
queue parameters
    QUEUE • 248
    START_DATE • 249
    START_TIME • 249
    using • 247
queue, API • 298
QUEUE_PATH • 59, 430

R

receiving files • 181
RECORD_FORMAT • 174, 430
record_format, API • 298
remote systems • 155
REMOTE_FILE • 165, 437
    send file parameter • 300
remote_file, API • 300
REMOTE_FILE_RF • 184, 438
    retrieve file parameter • 301
remote_file_rf, API • 301
REMOTE_SYSTEM • 438
remote_system, API • 301
REMOTE_SYSTEM_RF • 184, 439
REMOTE_SYSTEM_SJ • 193, 439
REMOTE_SYSTEM_SR • 189, 440
REMOVE_TRAIL_BLANKS • 59, 440
remove_trail_blanks, API • 305
REPORT_TITLE • 189
report_title, API • 306
reports, machine code characters • 106
resolving symbolic names, TCP/IP • 155
restart parameters
    RESTART_SUPPORTED • 256
    RETRY_TIME • 257
RESTART_SUPPORTED • 256, 441
restart_supported, API • 306
restarting transfers • 255
RETPD • 179, 442
retpd, API • 306
Retrieve File command • 181
    chxcom62 command • 148, 149, 255
    parameters • 181
    xcomtcp command • 148, 149, 255
Retrieve File parameters
    FILE_OPTION_RF • 182

LOCAL_FILE_RF • 183, 290
REMOTE_FILE_RF • 184, 301
Retrieve File parameters, REMOTE_SYSTEM_RF • 184
RETRY_TIME • 257, 442
return codes, system restrictions • 225
return values • 321
RMLNTFYL • 254
RMTNTFYL • 442
rmtntfyl, API • 298

S

sample API program • 329
SAVE_PASSWORD_IN_CNF • 445
global parameter • 59
script changes and shell • 88
scripts
xcomend • 92
xcomlp • 96
xcomntfy • 98
xcompp • 90
xcompre • 85, 87
SECONDARY_ALLOC • 175, 445
secondary_alloc, API • 307
secure_socket, API • 307
security parameters
DOMAIN • 244
PASSWORD • 244
TRUSTED • 244
USERID • 244
using • 243
Send File command
chxcom62 command • 148, 149, 255
parameters • 161
xcomtcp command • 148, 149, 255
Send File parameters
CREATE_DIR • 162
FILE_OPTION • 163
LOCAL_FILE • 164
REMOTE_FILE • 165, 300
Send Job command
chxcom62 command • 148, 149, 255
parameters • 192
xcomtcp command • 148, 149, 255
Send Job parameters
LOCAL_FILE_SJ • 193
REMOTE_SYSTEM_SJ • 193
Send Report command
chxcom62 command • 148, 149, 255
IBM machine code characters • 191
parameters • 185
xcomtcp command • 148, 149, 255
Send Report parameters
CARRIAGE_CONTROL_CHARACTERS • 186
CLASS • 186
COPIES • 187
DESTINATION • 187
DISPOSITION • 187
FCB • 188
FORM • 188
HOLDFLAG • 188
LOCAL_FILE_SR • 189
REMOTE_SYSTEM_SR • 189
REPORT_TITLE • 189
SPOOL_FLAG • 191
server certificate for SSL • 344
session control • 80
SESSIONS_FILE • 60, 446
SHELL_CMD • 60, 446
SMTP_SERVER • 60
SNA/APPC
performing transfers using • 152
protocols • 153
SOCK_DELAY • 61, 447
SOCK_RCV_BUF_SIZE • 61, 447
SOCK_SEND_BUF_SIZE • 61, 447
specifying tables • 155
SPOOL_FLAG • 191, 448
spool_flag, API • 307
SSL CA certificate • 343
expiration • 342
SSL client
certificate • 344
configuring • 346
SSL server • 345
certificate • 344
configuring • 345
START_DATE • 249, 449
start_date, API • 307
START_TIME • 449
start_time, API • 308
START_TIME, queue parameter • 249
starting states, API • 321
starting xcomd • 44
STAT_FREQUENCY • 62, 451
stat_frequency, API • 308
STORCLAS • 176, 454

storclas, API • 308
store and forward • 242
structure of API • 261
SunLink, API • 315
system in error messages • 484
SYSTEM_USER_DATA • 455

T

TAPE • 179, 456
tape, API • 308
TAPEDISP • 180
TCP/IP
    name resolution • 155
    port • 428
    protocols • 154
    transfers • 80, 154
TCP_CMD • 63, 456
Technical Support • 627
TEMPDIR • 63
tempdir, API • 309
TP_CMD • 63, 457
TRACE_PATH • 63, 457
tracing • 224
    parameter, XTRACE • 79, 226
    return codes • 225
transfer protocols • 40
TRANSFER_NAME • 457
TRANSFER_TYPE • 458
TRANSFER_USER_DATA • 458
transferring files • 148, 149
    command prompt • 152, 154
troubleshooting
    purpose of procedures • 618
    test procedure • 619
TRUNCATION • 160, 464
truncation, API • 311
TRUSTED • 244, 464
trusted, API • 311
TXPI_BUF_SIZE • 71, 467
TXPI_RECEIVE_TIMEOUT • 467
TXPI_SEND_CHECK_FREQ • 71, 467
TXPI_TERM_TIMEOUT • 72, 468

U

UMASK • 72, 468
UNIT • 176, 468
unit, API • 312
UNIT_RF • 184
UNITCT • 180, 469
unitct, API • 312
USE_TP_SECURITY • 469
USERID • 244, 469
userid, API • 312

V

VERSION • 160, 470
version, API • 312
VOLCT • 180, 470
volct, API • 313
VOLSQ • 180, 470
volsq, API • 313
VOLUME • 177, 471
volume, API • 313
VOLUME_RF • 185

W

wildcard characters • 149
wildcard transfers
    FILE_OPTION_RF • 182
    LOCAL_FILE • 164, 289
    LOCAL_FILE_RF • 183, 290
    REMOTE_FILE_RF • 184, 301
    wildcard transfers, REMOTE_FILE • 165, 300
WRITER • 471

X

xcom.cnf • 142, 158
xcom.ses • 80
XCOM_PASSWORD • 473
    global parameter • 74
xcom_show_cipher • 74, 314
XCOM_TRUSTED_OVR • 474
    global parameter • 75
XCOM_USERID • 474
    global parameter • 75
XcomAPI • 259, 321
XCOMAPI return values • 321
XcomAPI structure • 261
xcomapi.h • 330
xcomend • 91
xcomlp • 95
xcomntfy • 97
xcompp • 89
xcompre • 85, 87
XCOMPRE_LOCAL • 474
    global parameter • 77

XcomQAPI • 260, 326
queue request • 334
return codes • 339
xcomqm • 81
XCOMSEND • 95
xcomtcp command • 148, 149, 255
options • 151
performing transfers using • 154
port and protocol • 156
syntax • 150
XENDCMD • 77, 474
XIDEST • 243, 475
xidest, API • 314
XLOGFILE • 227, 475
xlogfile, API • 314
XLPCMD • 78, 475
XLUNAME • 476
xluname, API • 315
XMODE • 476
xmode, API • 315
XNODESPEC • 476
xnodespec, API • 315
XNOTIFYCMD • 78, 476
XPPCMD • 78, 477
XPRECMD • 79, 477
XTRACE • 226, 481
xtrace, API • 319
XTRACE, testing and tracing parameter • 79