

IBM Connect:Direct for z/OS
6.0

Documentation



This edition applies to Version 5 Release 3 of IBM® Connect:Direct and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1993, 2018.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. IBM Connect:Direct for z/OS Release Notes.....	1
Hardware and Software Requirements.....	1
POSIX Environment and System Libraries.....	1
MBCS Transfers.....	2
License Key File.....	2
New Product Name.....	2
What's New in This Release.....	2
Known Restrictions.....	3
Chapter 2. Connect:Direct for z/OS Special Product Considerations.....	5
Other Product Considerations.....	6
Transmission Control Queue Considerations.....	7
Miscellaneous Considerations.....	7
UDT Special Considerations.....	8
Connect:Direct Secure Plus Considerations.....	8
Connect:Direct File Agent Special Considerations.....	10
Chapter 3. Upgrading IBM Connect:Direct.....	11
Migrating an Existing Network Map.....	12
Migrating existing STATS files.....	12
Migrate Stats data to Version 6.....	12
Backout or Migrate STATS data back to Previous Version.....	13
Using the DGA#FXAL REXX Exec.....	13
Migrating CICS from Prior Releases.....	22
Upgrading Connect:Direct File Agent.....	23
Chapter 4. Configuration Guide.....	25
Before you Install.....	25
Installation Requirements.....	25
Virtual Storage Requirements.....	31
Preparing TCP/IP Configuration (Optional).....	33
Security Planning.....	35
Planning for Parallel Sessions and Process Recovery.....	37
Planning the Network Map.....	37
Planning for Disaster Recovery Testing.....	38
Planning for Connect:Direct File Agent.....	38
Traces in Startup JCL.....	38
Configuring IBM Connect:Direct for z/OS.....	39
Define the IBM Connect:Direct VSAM Files.....	39
Define VTAM Resources.....	40
Building a Test IBM Connect:Direct Configuration (optional).....	40
Building the Initialization Parameter File.....	41
Installing the ISPF IUI.....	43
Start IBM Connect:Direct.....	50
Signing On to IBM Connect:Direct.....	52
Customizing the Product.....	52
VTAM Definitions.....	53
Defining APPLID of Local DTF.....	53
APPLID for IUI and Batch Sessions.....	54
VTAM and NCP Parameters.....	57

VTAM Definitions for Full Networking IBM Connect:Direct.....	60
Network or Domain Name in Cross-Domain Network.....	62
Optional Configuration Tasks.....	65
Customizing the Product.....	65
Customizing the Spool Transfer Feature.....	69
Customizing VPS/CDI.....	70
HP OpenView for SNMP Traps Customization.....	71
NetView for SNMP Customization.....	72
Configuring IBM Connect:Direct without SNA Support.....	73
Overriding IBM Connect:Direct for z/OS Default Language Environment Run-Time Options.....	75
IBM Connect:Direct Enqueue Resource Management.....	77
Initialization Errors.....	77
Overriding IBM Connect:Direct Initialization Parameters.....	78
Initialization Errors.....	78

Chapter 5. CICS Administration and User Guide.....89

CICS Interface.....	89
CICS Screen Conventions.....	91
About the IBM Connect:Direct for z/OS CICS Documentation.....	92
Configuring the CICS Interface.....	92
Activate the CICS Component - Optional.....	92
Build the CICS Configuration File through ISPF.....	93
Load IBM Connect:Direct CICS Option into the CICS System Definition.....	95
Customize the CICS Interface Overview.....	95
Tune the CICS Interface.....	97
Using the Administration System.....	98
The Administration Primary Menu.....	99
Maintaining Configuration Information.....	100
Configuration Categories.....	101
Accessing the Configuration Screen.....	101
Updating the Control Record.....	102
Updating DTF Node Records.....	106
Updating Network Node Records.....	107
Working with the Administration Interface.....	108
Operating the CICS Interface.....	109
Viewing Node Status.....	111
Viewing the Work Queue Screen.....	113
Signon Defaults.....	114
Accessing the Signon Defaults Screen.....	114
Viewing User Status.....	115
Operational Considerations.....	116
Signing On to Multiple DTFs from a Single IUI.....	116
Signing On to a Single DTF from Multiple IUI Facilities.....	117
Performing an Immediate or Uncontrolled Shutdown.....	117
Restarting Task ABENDS.....	118
Accessing Accounting and Logging Information.....	118
Using the Extended Submit Facility (ESF).....	118
Specifying IBM Connect:Direct Signon Parameters.....	118
About IBM Connect:Direct CICS Data.....	119
Signing On and Off CICS.....	120
Signing On.....	120
Using the Primary Menu.....	122
Using the SN Option for Multiple Terminal Signon.....	123
Updating Your Signon Defaults.....	123
Using the ESF Session Mode Option.....	124
Signing Off.....	125
About Copying Files.....	125

Generating a Copy File from Connect:Direct for CICS.....	127
Copying a File from Your Node to Your Node.....	132
Building, Modifying, and Submitting Processes through CICS.....	133
Submit Process Screen.....	133
Security Override Screen.....	135
Selecting a Process through CICS.....	136
Determining the Status of a Submitted Process.....	136
Viewing Summary Information.....	138
Viewing Process Detail.....	139
Selecting Statistics through CICS.....	143
Viewing Process Statistics.....	143
Viewing Statistics Summary Information.....	145
Displaying CICS Messages.....	146
Displaying Product Messages.....	146
Displaying the Last Message.....	147
Displaying Your CICS User Profile.....	148
CICS Messages and Problem Isolation.....	149
Product Messages.....	149
Problem Isolation.....	156

Chapter 6. Administration Guide..... 157

Basic System Administrative.....	157
IBM Connect:Direct Native Command Structure.....	159
Managing Tasks.....	161
Connect:Direct Secure Plus Commands.....	166
Displaying Initialization Parameter Settings.....	168
Displaying IBM Connect:Direct/Plex Status.....	173
Stopping IBM Connect:Direct.....	173
Suspending and Resuming Quiesce and Trace Settings.....	176
Global Signon Defaults.....	181
Implementing Security.....	183
Security Exits.....	184
SIGNON Command Sequence.....	185
Implementing Security Exits.....	189
IBM Connect:Direct Secure Point-of-Entry.....	208
Trusted Node Security.....	209
Implementing a CA-ACF2 Environment.....	211
Implementing an IBM RACF Environment.....	212
Implementing a CA-TOP SECRET Environment.....	213
Configuring Firewall Navigation.....	214
Troubleshooting Security Errors.....	217
Maintaining User Authorization.....	220
Authorization File.....	221
Maintaining the Type File.....	234
Overriding File Attributes.....	234
INSERT TYPE and UPDATE TYPE Commands.....	235
DELETE TYPE Command.....	243
SELECT TYPE Command.....	244
Maintaining the Network Map.....	246
Local Node Entry.....	247
When to Define the Local Node as the Adjacent Node.....	248
API Signons.....	267
Examples of Local and Adjacent Node Records.....	268
How to Update the Network Map.....	274
Viewing the Network Map.....	281
Configuring a IBM Connect:Direct/Plex Environment.....	282
A New IBM Connect:Direct/Plex Environment.....	284

Advanced IBM Connect:Direct/Plex Configuration Considerations.....	286
Converting an Existing Stand-Alone Server to a Plex Environment.....	290
Converting Two Existing Stand-Alone Server Systems to a Plex Environment.....	293
Additional IBM Connect:Direct/Plex Configuration Examples.....	297
Configuring Extended Recovery.....	303
Setting Up Extended Recovery for a IBM Connect:Direct/Stand-Alone Server.....	303
Setting Up Extended Recovery for a IBM Connect:Direct/Plex Environment.....	304
Configuring SNMP Support.....	306
Identifying Trap Variables.....	307
Setting Up SNMP.....	312
IBM Connect:Direct Exits.....	315
Statistics Exit.....	316
Submit Exit.....	322
ALLOCATION EXIT.....	329
I/O Exit.....	335
Data Exit.....	339
WLM Exit.....	344
Tapemount Exit.....	345
Process Exit for Testing (DGAXPRCT).....	346
Custom Messages in the Message Library.....	354
Message IDs.....	355
Customizing Submit Screens.....	356
Step 1 - Modify the Existing Menu DGA@SM03.....	356
Step 2 - Define General Purpose Process.....	357
Step 3 - Provide a New Submit Screen.....	357
Administering Statistics.....	362
Tools to Monitor the Statistics Facility.....	362
Optimizing Statistics Files.....	364
Change the File Pair Configuration.....	365
Archiving Statistics.....	366
Displaying the Status of the Statistics Logging Facility.....	374
Displaying the Statistics Archive File Directory.....	375
Switching the Statistics File Pair.....	378
Recording Statistics for Specific Record Types.....	378
Notifying IBM Connect:Direct of Statistics File Archival.....	380
Managing the Transmission Control Queue.....	380
Configure the TCQ.....	381
How to Troubleshoot the TCQ.....	381
Supporting DBCS and MBCS.....	384
Overview of DBCS.....	384
Translation Tables.....	385
Customize Translation Tables.....	385
Alternate Logic Applied to DBCS and SBCS Translation (LOGIC =B (B,RC)).....	394
MBCS Conversions.....	396
Performance Tuning.....	397
Analyzing TCP/IP Performance.....	397
How to Improve BSAM Data Transfer Rates.....	400
Increasing Throughput and Decreasing CPU Utilization.....	401
Using zFBA for File Transfer.....	406
How zFBA works with IBM Connect:Direct.....	406
Special Considerations when using zFBA with IBM Connect:Direct.....	407
zFBA Error Handling.....	407
Using zEDC with IBM Connect:Direct.....	408
Simultaneous Session and Asset Tracking Reporting.....	409
SMF Record Layout.....	409
H2 Stat Record layout.....	411
Reporting.....	412
Isolating Problems.....	413

Providing Dumps to Customer Support.....	414
Resolving the Problem.....	418
Reviewing IBM Connect:Direct Messages and Sense Codes.....	419
Examine Output from Select Commands.....	420
Verify File Attributes.....	422
Common Errors.....	422
Diagnostic Tools.....	433
Global Initialization Parameters.....	446
ABEND.CODES.NODUMP=(ABEND code list).....	446
ABEND.RUNTASK = (<u>DUMP</u> ABEND.CODES.NODUMP).....	447
ALLOC.CODES = (allocation errors).....	447
ALLOC.MSG.LEVEL = <u>INFO</u> <u>WARN</u> <u>SEVERE</u>	448
ALLOC.RETRIES = number of retries.....	448
ALLOC.WAIT = hh:mm:ss.....	448
ALLOCATION.EXIT = modname.....	449
CDPLEX = <u>NO</u> <u>YES</u>	449
CDPLEX.MAXSERVER = maximum number of servers <u>4</u>	449
CDPLEX.TIMER = <u>5</u> number of minutes.....	449
CDPLEX.WLM.GOAL = (<u>NO</u> <u>YES</u> , exitname).....	449
CHECK.CERT.EXPIRE = <u>NO</u> <u>YES</u>	450
CHECK.CERT.EXPIRE.TIME = <u>00:00:00</u> HH:MM:SS.....	450
CHECK.CERT.EXPIRE.WARN.DAYS = <u>30</u> nnn.....	450
CKPT = nK nM.....	450
CKPT.DAYS = number of days.....	451
CKPT.MODE = (<u>RECORD</u> <u>BLOCK</u> <u>BLOCK</u> <u>RECORD</u> <u>PDS</u> <u>NOPDS</u> <u>NOPDS</u> <u>PDS</u> <u>VSAM</u> <u>NOVSAM</u> <u>VSAM</u> <u>NOVSAM</u>).....	451
COMPRESS.EXT = <u>ALLOW</u> <u>DISALLOW</u>	451
COMPRESS.NEGO.FAIL = <u>STEP</u> <u>PROCESS</u>	452
COMPRESS.NETMAP.OVERRIDE = <u>ALLOW</u> <u>DISALLOW</u>	452
COMPRESS.STD = <u>ALLOW</u> <u>DISALLOW</u>	452
CONFIRM.COLD.START = <u>YES</u> <u>NO</u>	452
CRC = (<u>OFF</u> <u>ON</u> , <u>YES</u> , No).....	453
CTCA = <u>NO</u> <u>YES</u>	453
CTCA.TIMER = number of seconds.....	453
DATEFORM = (<u>MDY</u> <u>DMY</u> <u>YMD</u> <u>YDM</u>).....	453
DEBUG = nnnnnnnn.....	454
DEFAULT.PERMISS = (text_permissions <u>644</u> , binary_permissions <u>755</u>).....	455
DESC.CRIT = (descriptor code).....	455
DESC.NORM = (n,n,...).....	455
DESC.TAPE = (n,n,...).....	455
DSNTYPE = <u>YES</u> <u>NO</u>	456
ECZ.COMPRESSION.LEVEL = <u>1</u> n.....	456
ECZ.MEMORY.LEVEL = <u>4</u> n.....	456
ECZ.WINDOWSIZE = <u>13</u> nn.....	456
ESF.WAIT = hh:mm:ss.....	457
EXPDT = (TT,DD,TD,DT) (if multiple values EXPDT = TT DD TD DT ALL <u>NONE</u> (if only one value).....	457
EXTENDED.RECOVERY = <u>NO</u> <u>YES</u>	458
FASP = (<u>NO</u> <u>SSP</u> , <u>NO</u> <u>SSP</u>).....	458
FASP.FILESIZE.THRESHOLD = nnn nK nM nG.....	458
FASP.BANDWIDTH = nnn nK nM nG.....	458
FASP.POLICY = <u>FAIR</u> <u>FIXED</u> <u>HIGH</u> <u>LOW</u>	459
FIPS = <u>NO</u> <u>YES</u>	459
GDGALLOC = <u>GENERATION</u> <u>DSNAME</u>	459
GDGENQ = <u>YES</u> <u>NO</u>	460
IMMEDIATE.SHUTDOWN = <u>I</u> <u>R</u> (I, nnn <u>60</u>) (R, nnn <u>60</u>).....	461
INITPARM.BACKUP.....	461
INITPARM.BACKUP = member.....	461

INVOKE.ALLOC.EXIT = SEND REC BOTH.....	462
INVOKE.ALLOC.EXIT.ON.RESTART = YES NO.....	462
INVOKE.SPOE.ON.SNODEID = NO YES.....	462
MAX.AGE = (nnn , * *(nnn) ALL ALL(nnn) status_type status_type (nnn) , list).....	462
MAX.AGE.TOD = time.....	464
MAXPRIMARY = number of PNODE sessions.....	464
MAXPROCESS = number of executing PNODE and SNODE Processes.....	464
MAXRETRIES = number of retries.....	464
MAXSECONDARY = number of SNODE sessions.....	465
MAXSTGIO = maximum storage used for sequential data set transfers for system-determined NCP, maximum I/O storage for user-specified NCP).....	465
MAX.TAPE = number of tape Processes NONE.....	465
MAXUSERS = number of users.....	466
MCS.CLIST = console operator CLIST library file name.....	466
MCS.SIGNON = (SIGNON USERID = (user ID,password) NETMAP = network map [optional parameters]).....	466
MULTI.COPY.STAT.RCD=not set CT MC M2.....	467
NETMAP.CHECK = NO (ALL TCP, ALL BOTH NODENAME, FAIL WARN PASS).....	467
NETMAP.CHECK.ON.CALL= YES NO.....	468
NODE.QUIESCE.OFF = NODENAME.....	469
NODE.QUIESCE.ON = NODENAME.....	469
NODE.TRACE.OFF = NODENAME.....	469
NODE.TRACE.ON = (NODENAME,nnnnnnnn).....	470
NON.SWAPABLE = YES NO.....	470
PDSE.SHARING = YES NO.....	470
PDSENQ = YES NO.....	471
PROCESS.RETENTION = YES NO.....	471
PRTYDEF = Process priority.....	472
QUIESCE = YES NO.....	472
QUIESCE.NODE = node name.....	472
REMOTE.DUMMY.PASSWORD = [YES INTERNAL].....	473
REQUEUE = YES NO	473
RESET.ORIGIN.ON.SUBMIT = YES NO.....	474
REUSE.SESIONS = YES NO.....	474
ROUTCDE.CRIT = (route code).....	474
ROUTCDE.NORM = (route code).....	475
ROUTCDE.TAPE = (route code).....	475
RUN.JOB.EXIT = modname.....	475
RUNJOBID = USER CD.....	475
RUN.TASK.EXIT = modname.....	475
RUNTASK.RESTART = YES NO.....	476
S+CMD.ENFORCE.SECURE.CONNECTION = YES NO.....	476
SECURE.DSN = filename.....	476
SECURE.SSL.PATH.PREFIX = prefix.....	476
SECURITY.EXIT SECURITY.....	477
SECURITY.NOTIFY = YES NO HOLD.....	477
SERVER.MSU = number 2147483647.....	478
SERVER.TYPE = [PROD TEST].....	478
SESSION.HIGHWATER.SMF = (133 128-255, 1-60) NO.....	478
SNA = YES NO.....	478
SNMP = YES NO.....	479
SNMP.DSN = data set name data set name (member).....	479
SNMP.MANAGER.ADDR = hostname IP address.....	479
SNMP.MANAGER.PORTNUM = port-number.....	479
SNODE.ARCH.RECALL.WAIT = NO YES.....	479
STAT.ARCH.CONFIRM = YES NO.....	480
STAT.BUFFER.ESDSDATA STAT.BUFFER.KSDSINDX STAT.BUFFER.KSDSDATA.....	480
STAT.ERROR = ABEND DISABLE.....	481

STAT.EXCLUDE = (record type list).....	481
STAT.INIT = <u>WARM</u> COLD.....	481
STAT.QUEUE.ELEMENTS = statistics record queue size.....	482
STAT.SNODEID = (<u>NO</u> YES,NO <u>YES</u>).....	482
STAT.SWITCH.SUBMIT = dsn [member].....	483
STAT.SWITCH.TIME = (hh:mm:ss , ...).....	483
STAT.TPREC = (start_time, end_time, snaps_per_hour).....	483
STAT.USER = (user ID, [password]).....	484
STATISTICS.EXIT = modname (modname[, <u>MANAGER</u> SERVER BOTH]).....	484
STRNO.MSG = number <u>10</u>	484
SUBMIT.EXIT = modname.....	485
SYSOUT = class.....	485
TAPE.PREMOUNT = YES <u>NO</u> LIST.....	485
TAPEIO = <u>BSAM</u> EXCP.....	485
TAPEMOUNT.EXIT = modname.....	486
TCP = OES NO.....	486
FASP.FILESIZE.THRESHOLD = nnn nK nM nG.....	486
TCP.API.LISTEN = ((addr , port) , (addrn , portn)).....	486
TCP.API.TIMER = <u>00:00:00</u> hh:mm:ss.....	487
TCP.CONNECT.TIMEOUT= 0 number of seconds.....	487
TCP.FMH.TIMER=hh:mm:ss.....	487
TCP.LISTEN = ((addr , port) , (addrn , portn)).....	488
TCP.RUNTASK.TIMER = hh:mm:ss.....	488
TCP.SOURCEIP = (IPv4 address or DNS IPv6 address or DNS)	488
TCP.SRC.PORTS.....	489
TCP.SRC.PORTS.LIST.ITERATIONS = number of scans.....	490
TCP.TIMER = wait time.....	490
TCQ = <u>WARM</u> COLD.....	490
TCQ.THRESHOLD = <u>NO</u> YES nn.....	490
THIRD.DISP.DELETE = <u>YES</u> NO.....	491
TRACE.BUFFER = nnn <u>2</u>	492
TRANS.SUBPAS = <u>YES</u> NO.....	492
UDP.SRC.PORTS.....	493
UDP.SRC.PORTS.LIST.ITERATIONS = number of scans.....	493
UDT = YES <u>NO</u>	493
UDT.MAXPROCESS = <u>2</u> number of executing UDT Processes.....	493
UDT33.LISTEN = ((ipaddr,port)[,(ipaddr,port)...]).....	493
UPPER.CASE = YES <u>NO</u>	494
V2.BUFSIZE = (maximum transmission buffer size, TCP/IP send/receive buffer size).....	494
WTMESSAGE = <u>NO</u> YES (YES,nnn).....	495
WTRETRIES = hh:mm:ss <u>00:03:00</u>	495
XCF.NAME = XCF group name.....	495
ZEDC = ON <u>OFF</u> PREFERRED	495
ZFBA = YES <u>NO</u>	496
ZIIP = <u>NONE</u> EXTCOMP SSLTLS ALL PROJECT.....	496
IBM Connect:Direct System File Initialization Parameters.....	496
Local Initialization Parameters.....	498
CDPLEX.INITPARM.BACKUP = member.....	499
CDPLEX.MANAGER = <u>NO</u> YES.....	499
CDPLEX.MSGID = <u>NONE</u> xx.....	499
CDPLEX.PLEXCLASSES = (*,plexclass,...,plexclass).....	499
CDPLEX.REDIRECT.....	500
CDPLEX.REDIRECT.EXCEPTION = ((Mgr-IP, Ext_Svr-IP, Ext_Svr-port, Exception-IP, Exception-port),...).....	500
CDPLEX.SERVER = IBM Connect:Direct/Server name.....	501
CDPLEX.SERVER.JOBDSN = data set name.....	501
CDPLEX.SERVER.JOBMEM = ((member name,server name),...).....	501
CDPLEX.SERVER.NODE = node name.....	502

CDPLEX.TIMER = 5 number of minutes.....	502
CDPLEX.VTAM = (VTAM-APPL,P/S-Node-APPL).....	502
Using the UDT Protocol.....	502
Initialization Parameters for UDT Connections.....	503
Network Map Parameters.....	503
PROCESS Statement Parameter.....	503

Chapter 7. User Guide.....505

Introduction to IBM Connect:Direct commands.....	505
Supported commands.....	505
Writing IBM Connect:Direct commands.....	506
The batch interface.....	512
Batch interface job requirements.....	512
Sample job stream to run the batch interface.....	513
Using the MAXDELAY keyword parameter to synchronize submitted Processes.....	516
The Interactive User Interface (IUI).....	519
Primary Options Menu.....	519
Operator Tables.....	524
Messages.....	524
The Application Program Interface.....	528
DGADCHLA Program.....	529
DGADCHLA Examples.....	537
Sample Job Stream for Executing the Program.....	538
Managing Sessions.....	539
Signing On to Connect:Direct for z/OS.....	539
Swap Node Command.....	549
Sign Off Command.....	550
Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands.....	551
Building, Modifying, and Submitting Processes.....	552
Process Routing.....	553
Process Queuing.....	554
SUBMIT Command.....	560
Submitting Processes through the IUI.....	569
Using the SB IUI Option to Submit a Predefined Process.....	570
Using the DF IUI Option to Create, Edit, and Submit Processes.....	572
Using the CF IUI Option to Generate a COPY Process.....	574
Controlling Processes in the TCQ.....	579
Controlling Processes with Commands.....	579
Modifying a Process in the TCQ with CHANGE PROCESS.....	579
Suspending, Flushing, and Deleting Processes.....	586
Examining Processes in the TCQ.....	590
Process Queuing and Recovery.....	597
Logical Queues.....	597
Process Recovery.....	604
Process Results and Statistics.....	609
Statistics Log Records.....	609
SELECT STATISTICS Command.....	611
SELECT MESSAGE Command.....	631
The Network Map.....	632
Retrieving Records from the Network Map File.....	632
Translating TCP/IP Host Names to Network Addresses.....	634
Utility Programs.....	636
User Notification Programs.....	636
VSAM AMS Interface Program (DGADTAMS).....	641
Symbolic Resolution Utilities (DGADTSUB and DGADGSUB).....	642
Batch Compression Utility (DGASACMP).....	645
Determine “High-Water Mark” for a Period (DGADVITL).....	648

ADRSSU Interface Program (DGADSIOX I/O Exit).....	648
DGASCONV – Secure Parameter File Conversion Utility.....	649
Using the TCQ/TCX Repair Utility (DGADTQFX).....	649
IBM Sterling Connect:Direct FTP+ for z/OS (CDFTP).....	650
Chapter 8. Facilities Guide.....	663
IBM Connect:Direct for z/OS Facilities Overview.....	663
ARS Report Examples.....	663
IBM Connect:Direct for z/OS Activity Report.....	663
IBM Connect:Direct for z/OS Summary Report.....	664
IBM Connect:Direct Exception Report.....	667
Security Violations Report.....	668
Connect:Direct for z/OS Function Reports.....	669
Activity Reporting System.....	676
ARS and IBM Connect:Direct.....	676
Business Solutions Using ARS.....	677
Requesting ARS Statistical Reports.....	677
Requesting Reports Using ARS Screens.....	678
Running the Job.....	682
Requesting Multiple ARS Reports or Scheduled Processing.....	683
Modifying the Sample Job Stream.....	683
ARS Record Layouts.....	686
Description of an SAS Variable.....	687
Authorization Event Record.....	687
Change Process Termination Record.....	688
Copy Termination Record.....	689
Delete Process Termination Record.....	693
Display Statistics Record.....	693
Flush/Suspend Process Termination Record.....	694
PDS Member Copy Record.....	695
Process Submit Statistics Record.....	696
Process Termination Record.....	697
Run Job Termination Record.....	697
Run Task Termination Record.....	698
Signon/Signoff Statistics Record.....	699
Start IBM Connect:Direct Command Record.....	700
Stop IBM Connect:Direct Statistics Record.....	700
Write to Operator (WTO) Statistics Record.....	701
The Operator Interface.....	701
Concatenation of Operator CLIST.....	702
Sample Connect:Direct for z/OS CLISTs.....	702
Rules for Setting Up Connect:Direct for z/OS CLISTs.....	703
Connect:Direct for z/OS Commands.....	704
CLIST Examples.....	704
Operation Messages.....	706
Stopping IBM Connect:Direct.....	706
About Tape Mount Messages.....	707
Tape Device Allocation.....	709
Event Services Support.....	710
ESS Concepts and Components.....	710
Event Services Support Architecture.....	711
ESS Data Flow Using CICS API.....	713
Deciding What Event Data to Collect.....	714
Using ESS with the CICS API.....	716
Using ESS with the ESS User Exit.....	716
Spool Transfer Facility.....	731
Spool Transfer Components.....	731

VTAM Printer Support.....	731
IBM Connect:Direct API.....	732
Symbolic Definitions.....	738
Customizing VPSSCDI.....	739
Transferring Data to the JES Reader or Spool.....	739
Output to the JES Reader.....	740
Examples of JCL Output.....	740
Output to JES Spool Files.....	741
IBM Connect:Direct Banners.....	741
IBM Connect:Direct COPY Process.....	742
COPY TO Syntax to Support Spool Transfer.....	743
Symbolic Definitions for the SYSOUT Keyword.....	744
Sample Process for SYSOUT.....	746

Chapter 9. Secure Plus for z/OS..... 749

Connect:Direct Secure Plus for z/OS Overview.....	749
Security Concepts.....	749
Security Protocols.....	749
IBM Connect:Direct Access to System Resources for SSL or TLS.....	756
Connect:Direct Secure Plus Tools.....	757
Prerequisites.....	760
Connect:Direct Secure Plus for z/OS Documentation.....	761
Plan Your Implementation of the SSL or TLS Protocol.....	762
Transport Layer Security Protocol and Secure Sockets Layer Protocol.....	762
TLS or SSL Protocol Processing.....	763
Self-Signed and CA-Signed Certificates.....	765
Server Certificates and IBM Connect:Direct.....	768
Using the SecurePlus Admin Tool and Populating the Parameter File.....	770
Starting the Administration Tool.....	770
Connect:Direct Secure Plus Configuration.....	774
Populating the Parameter File Using Quick Start.....	776
Manual Parameter File Creation.....	776
Create the Parameter File Manually for the SSL or TLS Protocol.....	776
Configuration Guidelines.....	776
Adding the Local Node Record to the Parameter File Manually for the SSL or TLS Protocol.....	777
Adding a Remote Node Record to the Parameter File Manually for the SSL or TLS Protocol.....	781
Additional Configuration Options.....	785
Adding a Remote Node Record for the External Authentication Server.....	786
Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server.....	787
Implementing Strong Password Encryption.....	788
Configure the Local Node Record Imported from the Network Map.....	791
Configuration Guidelines.....	791
Configuring the Local Node Record for the SSL or TLS Protocol.....	792
Configure Remote Node Records Imported from the Network Map.....	795
Configuration Guidelines.....	795
Configuring a Remote Node Record for the SSL or TLS Protocol.....	795
Disabling Connect:Direct Secure Plus in a Remote Node Record.....	799
Configuration for a Secure Connection between z/OS and OpenVMS Nodes.....	800
Enable and Validate Connect:Direct Secure Plus Operation	801
Saving and Submitting the Connect:Direct Secure Plus Parameter File.....	802
Preparing IBM Connect:Direct for Secure Plus Operations.....	803
Parameter File Saving After the Initial Setup.....	803
Validating and Testing Connections by Session.....	803
Override Settings in IBM Connect:Direct Processes.....	804
PROCESS Statement Overrides for Connect:Direct Secure Plus Defaults.....	805
COPY Statement Overrides for Connect:Direct Secure Plus Defaults.....	805
Security Settings Override Examples.....	806

Maintain Connect:Direct Secure Plus.....	807
Parameter File Maintenance.....	807
Node Record Maintenance.....	811
Connect:Direct Secure Plus Statistics.....	815
SSL or TLS Statistics Record.....	815
Troubleshooting.....	817
Certificate Parameter Definitions.....	823
RACF Application Certificate Parameter Definitions.....	823
GSSKYMANT Utility Certificate Parameter Definitions.....	825
CA-ACF2 Application Certificate Parameter Definitions.....	827
CA-Top Secret Application Certificate Parameter Definitions.....	829
Configuration Worksheets.....	831
Local Node Security Feature Definition Worksheet.....	831
Remote Node Security Feature Definition Worksheet.....	832
.EASERVER Node Security Feature Definition Worksheet.....	833
.CLIENT Node Security Feature Definition Worksheet.....	834
Chapter 10. Quick Reference Guide.....	835
Notational Conventions.....	835
IBM Connect:Direct Process Statements.....	835
Process Statement.....	836
Intrinsic Variables.....	837
COPY Statement.....	838
RUN JOB Statement.....	841
RUN TASK Statement.....	841
Run Task Utility Programs and Parameters.....	842
SUBMIT Statement.....	843
SYMBOL Statement.....	843
Modal Statements.....	843
IBM Connect:Direct Commands.....	844
Modify Command.....	844
Network Map Commands.....	845
Process Commands.....	847
Signon, Signoff, Stop, and Swap Commands.....	849
Task Commands.....	849
Type Commands.....	850
User Commands.....	851
Inquire Commands.....	852
Statistics Commands.....	853
Submit Command.....	855
Network Map Parameters.....	856
Initialization Parameter Overview.....	857
Global Initialization Parameters.....	857
System File Initialization Parameters.....	862
Local Initialization Parameters.....	862
Status Codes.....	863
Trace Types.....	864
Chapter 11. Connect:Direct for z/OS EAV support	867
EATTR parameter.....	867
COPY statement support.....	867
VIEW PROCESS support.....	868
TYPE file support.....	869
DGADTDYN support.....	870
z/OS operating system support of EAV.....	871
Notices.....	873

Trademarks.....	874
Terms and conditions for product documentation.....	875

Chapter 1. IBM Connect:Direct for z/OS Release Notes

The *IBM® Connect:Direct® for z/OS® Release Notes* document supplements IBM Connect:Direct for z/OS documentation. Release notes are updated with each release of the product and contain last-minute changes and product requirements. Read the document before installation.

Hardware and Software Requirements

Hardware and software requirements for IBM Connect:Direct for z/OS are in the *Connect:Direct for z/OS Configuration Guide* and the *Connect:Direct for z/OS Program Directory*. Connect:Direct Secure Plus requires the same hardware and software and the additional software listed in the following table.

Sterling Connect:Direct File Agent must be installed in the UNIX System Services component of z/OS and configured to communicate with the IBM Connect:Direct server. Use a PC for terminal emulation to create a configuration file. Additional Sterling Connect:Direct File Agent requirements include:

Component or Functionality	Software
Connect:Direct for z/OS	<ul style="list-style-type: none">• One or more online zIIP processors to use the zIIP Exploitation Feature• One or more online zEDC accelerators to exploit the zEDC feature.
Connect:Direct Secure Plus for z/OS	<ul style="list-style-type: none">• TCP/IP connection to use the SSL or TLS protocol• IBM SystemSSL toolkit• IBM Language Environment• For Secure Plus, ICSF FMID HCR77A0 or higher is required unless cryptographic coprocessors are available. If coprocessors are available, any ICSF level supported by the Connect:Direct for z/OS release is accepted.
Connect:Direct File Agent	<ul style="list-style-type: none">• Hummingbird Exceed® PC X server for WIN32, version 7.0 or later for terminal emulation• Java 2 Platform, Standard Edition (J2SE). Download the J2SE from the IBM web site• 31-bit JVM (Java Virtual Machine)

POSIX Environment and System Libraries

Install and set up the UNIX System Services (or POSIX) environment before you install IBM Connect:Direct. The POSIX requirement and use of C/C++ and LE require that the following data sets be available to IBM Connect:Direct through the STEPLIB or LINKLST. In addition to C/C++ and LE, XPLINK is now required to support the File Accelerator and alters the data set list from previous releases.

- CEE.SCEERUN (IBM Language Environment)
- CEE.SCEERUN2 (XPLINK Requirement)
- CBC.SCLBDLL (C/C++ Run-time)
- SYS1.SIEALNKE (System SSL Environment)

If IBM Connect:Direct Resource Access Control Facility (RACF) program control is implemented, define these data sets and the IBM Connect:Direct SDGALINK to the program class.

MBCS Transfers

To perform MBCS transfers, verify that the Natural Language Resources component of Language Environment is installed on your operating system.

License Key File

To exploit the FASP feature through SSP you must obtain and store a HSAO license file inside the Connect:Direct for z/OS file system. This HSAO license file must be available to the IBM Connect:Direct during product initialization via the CDASPLIC DD statement.

New Product Name

The product formerly known as IBM Sterling Connect:Direct for z/OS now reflects the IBM Corporation brand and logo, and product is now referred to as IBM Connect:Direct.

What's New in This Release

In this release of IBM Connect:Direct for z/OS, and its related software, several features have been added to enhance functionality. The following sections describe these changes in more detail.

New features added to this release

- Enhanced logic for MAXDELAY has been added to allow for closer monitoring of the Process and the ability to restart the API session when certain communication failures occur. See [Using the MAXDELAY](#) in the *User Guide* for more information.
- An enhanced SESSION.HIGHWATER.SMF support to enhance monitoring and recording of Session Highwater mark by generating SMF data with additional data about the Connect:Direct environment and recording within the Connect:Direct Statistics facility.
- zEDC hardware compression has been enhanced to set a precedence to favor zEDC over the zIIP feature. See ZEDC [“Global Initialization Parameters”](#) on page 446 in the *Administration Guide*.
- Enhanced zLIB compression support to allow for better buffering of data prior to compression to ensure more data is eligible for compression and eligible for zEDC compression.
- Support of the FASP protocol through the Sterling Secure Proxy to exploit the FASP protocol between SSP and a Connect:Direct Server running on Windows or UNIX. Please reference “Planning for Aspera FASP for IBM Connect:Direct for z/OS” in the *Configuration Guide*.
- Support to allow the Connect:Direct SDGALINK to be allocate via LIBDEF. Please reference the *Configuration Guide* for details.
- Enhanced Secure Plus support to:
 - Exploit of a default certificate from the Key store
 - Support for easy cloning of SecurePlus parameter file entries
 - Generate a report of the SecurePlus configuration
 - Exploit the zERT feature of the zOS Communication ServerPlease reference the *SeucrePlus Implementation Guide* for details.
- Enhanced ISPF IUI support that enables or disables UI functionality based on the authorization of the user.
- Support of a global TCP SourceIP. Please reference Global TCP SourceIP in the *Configuration Guide* for details.

- Support to allow for multiple Operator CLIST data set concatenation. Please reference the *Facilities Guide* for details.
- Removal of the authorization bit mask from the DGASECUR macro allowing for easy customization of authorized functionality within the Stage2 Security exit. Please reference the *Administration Guide* for details
- Stage2 Security Exit enhancement to force SPOE translation if performed if the remote Connect:Direct access is attempting using the special passwords of IUI, BATCH or STC. Please reference the *Administration Guide* for details.
- Passing of the address of the UICB to the submit exit to allow the exit to set user text in the UICB fields UIERRM1 and UIERRM2.
- Add 2 new Operator CLIST to perform the Certificate Expiration Check and Refresh S+ commands.
- Enhanced modal logic to allow for check of the highest return code of all steps within a Process, checking of variable data passed to the process and checking of message id from a previous step.
- Select Statistics Enhanced to allow for:
 - Selecting by SNODE, by PNODE, by FROM node, by TO node, by Local node, by Remote node and by Submitter node, and also allow selecting by any node - that is, if a node is any of the preceding.
 - Customized Select Statistics panel that consolidates the basic with extended selection criteria
 - Add additional information to several Stat record types
 - A new 'WHERE2' layout for the header portion of the CT, FI, MC/M2, PI/ZI, PS/SW, PT/ZT, RJ, and RT records
 - Information on the reasons for use or non-use of ZEDC and ZIIP, including ZIIP fallback
 - Connect:Direct Server version to the CT record
 - File DCB information to the CT record
 - Add Process Class to the CT, PT, SB, SE, RT and RJ records
 - Add a Session End (SE) record that should pair up with the Session Begin (SB) record
 - Added new Operator Clist (OP) record that reflects the CLIST and from which data set the CLIST was executed from
 - Added flags CTFXNODE, MCFXNODE and FIFXNODE (P/S node fields were corrected) to resolve inconsistency in the stats report for CT, MC and FI records when Connect:Direct for z/OS is acting as the SNODE. For sample code to handle the FX flag, refer to [member DGAXSCPU in \\$CD.SDGASAMP](#).
- Enhanced the V2.BUFSIZE parameter to support up to 2MB buffer sizes
- Allow for setting the Default File Permission when creating new files in zFS file system
- Enhanced the command processor to ensure that all commands issued produce a WTO that identified the Userid for which issued the command
- Relaxed the authorization restrictions for the Stage1 Security exit.

Known Restrictions

Connect:Direct for z/OS and its related software for this release have the following known restrictions:

- Data loss had been discovered to occur under the following conditions:
 - CKPT/Restart is specified
 - Restart has occurred
 - zEDC is specified on the receiving side
 - file size and buffer size combinations that are boundary conditions in the code

To mitigate the problem until a complete solution can be developed, Connect:Direct will disable zEDC on the receiving side of the transfer if checkpointing is specified on the Process COPY step. If zIIP fallback is possible on the receiving side of the transfer, then zIIP will be utilized. Otherwise, software zLIB

decompression will be used. zEDC compression can be utilized on the sending side of the transfer without any data integrity issues. zEDC decompression can be utilized on the receiving side of the transfer if the Process specifies the COPY statement keyword, CKPT=OK.

- In previous versions of Connect:Direct for z/OS, the default behavior for TCP Source IP was to bind to the first IP address defined in the TCP . LISTEN initialization parameter. The V6R0 global initialization parameter, TCP . SOURCEIP, now defaults to HOSTNAME, obtained during initialization and each Process will bind to that address as the Source IP address for the PNODE. This is a new behavior from prior versions and may cause the SNODE to fail NETMAP checking of the Source IP. You can specify the new Source IP address in the SNODE's adjacent node entry by specifying an alternate communication address or ALT.COM to define that new IP address if it differs from what was previously defined. An alternative method on the PNODE is the SOURCEIP NETMAP parameter to define a specific IP address for the SNODE or specify a more appropriate address for the TCP . SOURCEIP initialization parameters. For more information, see *Global TCP SourceIP in the Configuration Guide*.

Chapter 2. Connect:Direct for z/OS Special Product Considerations

Review the following considerations before installing the product:

Installation Consideration

- If the installation data sets have been loaded to a cataloged data set with an HLQ other than IBM.fmid, the SMP/E RECEIVE command must change the HLQ by specifying the RFPREFIX(hlq) parameter, for example, RECEIVE SELECT(HDGA510) RFPREFIX(hlq).

Exits Considerations

- If you assemble and link-edit the sample exits, DGAXPALL and DGAXPNUM in the \$CD.SDGASAMP library, the following assembly error will occur: ASMA140W END record missing. Ignore this error, caused by comment lines after the SCEND statement.
- The IBM Connect:Direct startup JCL has a user-definable DD statement (USRINFO) for displaying user-defined information from User Exits. If the USRINFO DD is allocated, then IBM Connect:Direct opens it during product initialization and initializes it with the string SITA523I USRINFO INITIALIZED.

Initialization Parameters Considerations

- In a IBM Connect:Direct/Plex environment, if you specify ANYADDR or ANYADDR6 for the TCP.LISTEN local initialization parameter associated with a IBM Connect:Direct Server but do not specify the CDPLEX.REDIRECT parameter, a default address is created as the IBM Connect:Direct/Plex redirection address. For the IPv6 protocol, the default is generated by adding a::FFFF: prefix to the IPv4 default. If you do not want this to be your default address, specify the value you do want using the CDPLEX.REDIRECT parameter.

If you use the dynamic VIPA resources, you may need to specify the SOURCEIP parameter in the network map if the remote node cannot perform network map checking. The SOURCEIP address is used to bind the socket and by the remote node to perform netmap checking.

- The following initialization parameters are obsolete and must be removed from the initialization parameter file:

Obsolete Parameters		
ABNLIGNR	ALLOC.OPTIONS	ALLOC.STORAGE
APCTHRSH	APDSN	API.SYSTEMS
APPLID	BUFND	BUFSIZE
CD.KEY	CD.NODE	CDPLEX.TCPIP
CDPLEX.TCPNAME	CONNECT.WAIT	DUMP
ESTAE	EXT.COMPRESS.DEFAULT	LINES
LOG	LOG.PRINTER	LU1.ASYNC
LU1.NETMAP.CHECK	LU1.SCRIPT.DSN	LU2.WAIT
MAXRESTAR	MAXWAIT	MP
NDM.KEY	NDM.NODE	NETEX.MAX.OFFERS
NETEX.MAXRU	NETEX.NRBTIME	NETEX.OFFERTIM

Obsolete Parameters		
NETEX.OLD.PREFERRED	NETEX.SSNM	NETEX.VER
NETEX.WINDOW	OS390.TASKING	PASSWORD
PASSWORD.REPLY	PC.ENABLE.CHECK	POA.APPL
POSIX	PRTYINT	RESIDENT.MOD
SECURITY.TYPE	STIMEOUT	STORAGE.PER.TASK
TAPE.DETACH	TAPE.RETPD	TCP.ADDR
TCP.API.PORTNUM	TCP.BUFSIZE	TCP.NAME
TCP.PORTNUM	TCP.SNS.SUBSYS	TCQDSN
TCXDSN	TPAM	XETEX
XETEX.MAX.OFFERS	XETEX.MAXRU	XETEX.NRBTIME
XETEX.OFFERTIME	XETEX.SSNM	XETEX.VER
XETEX.WINDOW	XLU1.ASYNC	XLU1.SCRIPT.DSN
ZEDC.EXCLUDE	ZIIP.EXTCOMP.DATASIZE.THRESHOLD	

Process Retention Considerations

- To enable Process Retention, increase the TCQ space before activating it. A Process is retained in the PR queue until manually deleted, space is required, or MAX.AGE value is reached. Increasing TCQ space reduces the likelihood that a Process in the PR Queue is automatically deleted when space is required. For information on enlarging the TCQ, see Planning the Installation.

Network Map Considerations

- If IBM Connect:Direct FTP+ for z/OS communicates with an earlier version of Connect:Direct for z/OS, the CDFTP.TEMPFILE specification in the netmap must be an zFS file unless appropriate maintenance has been applied to that version of IBM Connect:Direct for z/OS.
- Session Management of Adjacent Nodes - Use the SESS.SNODE.MAX network map parameter only with the TCP/IP and LU6.2 protocols.

Storage Considerations

- To maximize available storage below the 16-MB line, define sufficient storage above the 16-MB line, and maximize concurrent processing, define a region size of 0 MB for IBM Connect:Direct.

Note: Depending on your system-specific storage requirements, the MAXTHREADTASK and MAXASSIZE parameters in SYS1.PARMLIB member BPXPRMxx may also have to be altered to ensure maximum storage and concurrent processing within IBM Connect:Direct.

Other Product Considerations

Review the following other product considerations before installing the product:

- UNIX System Service (BPX) calls are executed in the IBM Connect:Direct IUI under the TSO or Batch User ID. BPX calls require that a user ID has an OMVS segment defined to it within the external security product, such as IBM RACF, CA-ACF2 or CA-TOP SECRET. The BPX calls resolve the TCP/IP name or address for reporting purpose in Select Statistics. If a user ID does not have an OMVS segment defined to it, an SEC6 ABEND or Trace Resolver messages under the TSO or batch user can occur when requesting Select Statistics.

Information about adding an OMVS segment to a user ID using an external security product, either individually or as a group assignment, is found in the external security product documentation.

- Due to an issue with the security toolkit on the i5/OS hardware, an SSL/TLS cipher suite can be negotiated during the SSL/TLS handshake that is not defined in the cipher suite list. IBM PMR, 35692, has been opened to address the i5/OS toolkit issue. When the PTF becomes available for i5/OS, apply it to ensure cipher suite negotiation occurs as expected.

Transmission Control Queue Considerations

Review the following Transmission Control Queue considerations before installing the product:

- Because of the changes in the Transmission Control Queue (TCQ) to support Processes containing up to 1 MB, review all SUBMIT exits. Modify any SUBMIT exit that:
 - Alters the size of the TCQE (change TQRECLN)
 - Uses TQCSPRD (pointer to first, that is, current step)
 - Uses TSHFTCQE (forward pointer to next step)
 - Uses TSHBTCQE (backward pointer to previous step)
 - Adds or changes PACCT or SACCT information

The TQCSPRD, TSHFTCQE, and TSHBTCQE fields contain offsets relative to the start of the TCQE for Processes that do not exceed 64 KB. For larger Processes, these fields must be multiplied by 16 before being used as an offset. In the TCQE, a flag indicates whether or not this processing should take place (the flag byte is TQFLAGA and the bit equate is TQGT64K).

SUBMIT exits are no longer allowed to change the size of the TCQE (modify TQRECLN). The sample SUBMIT exit DGAXACCT previously showed how to add the PACCT and SACCT fields to the end of the TCQE and extended the length. Because this is no longer allowed, fields have been reserved for the PACCT and SACCT information to allow the SUBMIT exit to add or modify accounting information as required. The sample SUBMIT exit DGAXACCT has been modified to show you how to manipulate this account information.

- Using IGGCSI00 to access files requires READ access to the master catalog.
- Only Version 2 Flows, which are used for the TCP/IP and SNA LU6.2 protocols, support the ability to checkpoint files larger than 4 GB. Version 1 Flows, which were used for VTAM SNA LU0, do not support this ability.
- To honor the permission setting for HFS files using the PERMISS keyword, you must set the z/OS UNIX System Services UMASK to 000 either by default or by using the runtime environment variable, _EDC_UMASK_DFLT. To set the environment variable, define the _EDC_UMASK_DFLT=000 variable in a RECFM=VB type file and allocate the ENVIRON DD in the IBM Connect:Direct startup JCL.
- Example: //ENVIRON DD DISP=SHR,DSN=\$CD.ENVIRON(TZ)

Miscellaneous Considerations

Review the following miscellaneous considerations before installing the product:

- Because the standby extended recovery job and the active IBM Connect:Direct node use the same VTAM APPLID, TCP/IP address, and port, the standby extended recovery job must run in the same z/OS image as the active IBM Connect:Direct node unless you use the Dynamic VIPA and Dynamic SNA information described in Preparing TCP/IP Configuration and Configuring Extended Recovery.
- If you copy a file that is in IBM-proprietary compressed format to a new file that is not, IBM Connect:Direct may not allocate enough space for the new file. Such COPYs may experience allocation failures (Sx37 type ABENDs), which will require SPACE to be manually coded in the Process.
- Scheduling or automated operations packages can result in attempts to process incomplete or empty files. Scheduling packages, such as CA-7, ZEKE, or CONTROL-M and certain automated operations packages, such as CONTROL-O, are often set to start jobs upon a file's closure and an SMF record being cut or based upon specific messages appearing on the system console. When using Connect:Direct for

z/OS, this method of starting jobs can result in attempts to process incomplete or empty files. This situation can occur when an active transfer is interrupted and must be restarted. The receiving file will be closed, but upon Process restart it will be written to by Connect:Direct for z/OS.

To prevent erroneous processing by such packages, set up Processes with conditional logic to verify that the copy successfully completes, and then use the scheduling or automation package's provided program (U7SVC, ZEKENDM, CTM@IF10) in a IBM Connect:Direct RUN TASK.

- Checkpointing is ignored when transferring HFS files.
- IBM C Program Containing MAIN() or Enclave - You cannot execute any IBM C program that contains a MAIN() section or enclave as a Run Task or Exit. FTP is an example of this type of program.
- LU6.2 Does not Support Redirection of Processes in IBM Connect:Direct/Plex Environment -The LU6.2 connection protocol does not support the feature, which allows the IBM Connect:Direct Manager to redirect work to one of its servers. The remote node must address the server on which you want to run an LU6.2 Process. To do this, specify the node name and VTAM address of the IBM Connect:Direct Server on which the Process is to run in the remote server's network map. Use the same CDPLEX.SERVER.NODE and CDPLEX.VTAM specified for the local initialization parameters for the IBM Connect:Direct Server you are trying to address.

Another stipulation related to the LU6.2 protocol involves a situation where one of the IBM Connect:Direct servers in a IBM Connect:Direct/Plex environment is communicating with a stand-alone IBM Connect:Direct DTF. In this case, all LU62 processes for this specific node must be directed to the same server and you can use a PLEXCLASS to accomplish this.

- IBM Connect:Direct cannot transfer LBI files using LU0 connections.

UDT Special Considerations

Review the following UDT considerations before installing the product:

- UDT requires three threads for each executing Process which may impact your storage and concurrent processing capabilities. If the number of available threads becomes critically low, you may need to increase the MAXTHREADS limit per PROCESS in the OMVS environment.
- To use UDT for file transfers on high-speed networks with latency, refer to the following table which summarizes performance differences between TCP and UDT:

Bandwidth	Latency	Results
45 Mbps	All latencies	TCP is equal to UDT
155 Mbps	Less than 20 ms	TCP outperforms UDT
	Greater than 20 ms	UDT outperforms TCP
1 Gbps	Less than 10 ms	TCP outperforms UDT
	Greater than 10 ms	UDT outperforms TCP

For information on UDT, see *Configuring Firewall Navigation* and *Using the UDT Protocol* in *IBM Connect:Direct for z/OS Administration Guide*.

Connect:Direct Secure Plus Considerations

Review the following Connect:Direct Secure Plus considerations before configuring the product:

- Support for the TLS 1.1 and TLS 1.2 protocols require certain version and release levels, as well as PTF service to use those protocols. See the Program Directory for detailed information on these requirements.
- The support for Security Policy SP800-131a and the Suite B profile requires that the Secure Plus and System SSL be in FIPS mode. See the information below and the System SSL Programming Guide regarding FIPS.

- Connect:Direct Secure Plus uses services from ICSF that require the Connect:Direct ID for Secure Plus and the TSO user ID of the Secure Plus Administrator to have read access defined in the RACF CSFSERV facilities class.
- The enhancements to Connect:Direct Secure Plus such as, new protocols added, support for new security policies, and removal of support for STS require a Secure Parmfile conversion. For more information see, [“DGASCONV – Secure Parameter File Conversion Utility”](#) on page 649.
- IBM Connect:Direct administrators need access to z/OS UNIX System Services in order to update SSL and TLS parameters.
- IBM Connect:Direct needs UNIX read permission to the key database. Use the UNIX CHMOD command to change permissions, if necessary.
- A default certificate must exist for the SSL or TLS protocol to successfully communicate with a trading partner. Use the IBM GSKKYMAN utility (or IBM RACF if using key rings) to set a trading partner certificate as the default. For information about creating and managing certificates, refer to the IBM documentation. To locate documentation, go the IBM web site and perform a search on GSKSSL10.
- Secure connections cannot be established using LU0 or PNODE=SNODE. Nonsecure connections operate normally using these protocols.
- You can only use the Quick Start option the first time you create and populate the Connect:Direct Secure Plus parameter file from the network map, and if your IBM Connect:Direct TCP/IP netmap entries use standard IPV4 addresses.
- While in FIPS mode, the following are important considerations for Connect:Direct Secure Plus:
 - The certificate store must be in FIPS mode and the certificate must meet size restrictions for encryption keys. The GSKKYMAN utility can be used to create a key database for FIPS mode. In addition, other requirements of RACF might be necessary. For more information, see [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#)
 - While in FIPS mode, IBM Connect:Direct Secure Plus can open a FIPS mode key database; however, initialization fails if the key database is not in FIPS mode. If the FIPS initialization parameter is specified as NO, IBM Connect:Direct Secure Plus can still open and use a FIPS-mode database.
 - When Connect:Direct FTP+ is in FIPS mode, the TLS protocol is the only supported protocol. If SSL is enabled in the secure parameter file, the connection to that remote node is rejected during the TLS handshake.
 - While in FIPS mode, only certain ciphers are supported. During the TLS handshake, any non-FIPS mode ciphers are ignored. The Secure Plus Admin tool provides the following textual representation of the ciphers available in FIPS mode:
 - SSL_RSA_AES_128_SHA
 - SSL_RSA_AES_256_SHA
 - SSL_RSA_WITH_3DES_EDE_CBC_SHA
 - The following errors cause Connect:Direct FTP+ to terminate with a U4079 abend. These are critical functions and indicate a severe problem requiring immediate attention:
 - CMSERR_BAD_RNG_OUTPUT**
Failure during random number generation
 - GSK_ERR_RNG**
Failure during random number generation
 - CMSERR_FIPS_KEY_PAIR_CONSISTENCY**
Failure when generating either an RSA or DSA keypair
 - CMSERR_KATPW_FAILED**
Failure was encountered by the gsk_perform_kat API when performing known answer test against the System SSL cryptographic algorithm

Connect:Direct File Agent Special Considerations

Connect:Direct File Agent has the following special considerations:

- Detect when VSAM files are created, but not when they are updated.
- When watching for a VSAM file, it triggers a Process three times because it detects the creation of the cluster, data, and index portions of the VSAM dataset. To prevent the Process from being triggered unnecessarily, append.DATA to the data set name. To see an example involving a VSAM data file, see the Connect:Direct File Agent help.
- Connect:Direct File Agent ignores PDSE data sets in a watch directory.
- Do not use the Search tab on the IBM Connect:Direct configuration GUI Help when running the GUI as a z/OS batch job. It may terminate the configuration GUI session.
- To avoid a parsing error when a substitution for a variable would include an open or closing parenthesis, enclose the variable in double quotes.

For example, if **%FA_FILE_FOUND.** is to be passed as the Process Argument &DSN, specify the variable as:

&DSN="%FA_FILE_FOUND."

Note: Variable values enclosed in quotes must not be concatenated with other values. Any attempt to do this will cause a submit failure or subsequent Process failure.

- Some Hummingbird Exceed releases do not allow you to close the Help windows.

Chapter 3. Upgrading IBM Connect:Direct

Before you begin

Review the following items before upgrading your existing IBM Connect:Direct system.

- You must re-assemble and re-link all user exits after you complete the installation procedure.
- If previous modifications were made to the DGASECUR and DGA\$MFLG, update the DGAXAUTH macro with those modifications.
- After installation, reassemble and link-edit your security module that uses the DMGSECUR macro (new element name is DGASECUR) to ensure that all enhancements are implemented. For information, see [Implementing Security](#).
- After IBM Connect:Direct files have been migrated, using these files with a previous release of IBM Connect:Direct results in the inability to view user comment fields in the network map. If you update the network map using an older version, any information stored in the comment fields may be lost.
- For archive JCL and Processes that use STAT archive files or STAT ESDS, update the LRECL to 2048 and update all archive jobs and procedures, which must change the logical record length. You can update the LRECLs in advance and use the LRECL updates with any release.
- To ensure above-the-line storage, calculate the difference between the maximum storage values used for BSAM data transfers (as specified by the MAXSTGIO initialization parameter) and change the JOBSTEP region accordingly. For additional information, see [Improving Performance](#).
- Performing an upgrade is similar to performing an initial installation. You can retain some or all data files created by the initial installation or as a result of the last upgrade.

To retain existing IBM Connect:Direct files when you migrate, use the information in the following table. These files are required to fall back to the previous version.

Data Set	Migration Considerations
NETMAP file	Unload, delete, reallocate, and reload the network map. For more information, see Migrating an Existing Network Map .
AUTH, CKPT, and TYPE files	Use IDCAMS REPRO to migrate the data to the new files. For the AUTH file, use this DCB information: <div>DCB=(DSORG=PS,RECFM=VB,LRECL=2052,BLKSIZE=24624)</div> For the CKPT (Checkpoint) file, use this DBC information: <div>DCB=(DSORG=PS,RECFM=VB,LRECL=20004,BLKSIZE=27998)</div> For the TYPE file, use this DCB information: <div>DCB=(DSORG=PS,RECFM=VB,LRECL=1934,BLKSIZE=29010)</div>
STATS files	See, Migrating of Existing STATS files steps below.
MSG file	Because new messages exist in the MSG file, use the updated message file. Any user-defined messages can be loaded using the DGAJMSGL JCL in the \$CD.SDGAJCL installation data set.

Data Set	Migration Considerations
IBM Connect:Direct initialization parameters file	<p>After migrating data from a previous release, specify the IBM Connect:Direct initialization parameters TCQ=WARM and STAT.INIT=WARM and attempt the initialization.</p> <p>Note: Make sure that all obsolete initialization parameters are removed before you start up. See the list of initialization parameters that are obsolete.</p>

Migrating an Existing Network Map

About this task

To use an existing network map when you migrate to this version of IBM Connect:Direct:

Procedure

1. Run the DGAJUNLD job stream to unload the network map file of your current release by having your UNLOAD DD statement point to your upgrade release CNTL library.
2. Make changes to the initialization parameters and edit the network map source, if necessary. Then run DGAJJDEF to redefine your network map, and DGAJLOAD to reload it.
3. If you change the name of your network map:
Update your NETDSN initialization parameter, DGAJBATC member in the \$CD.SDGAJCL data set, and ISPF main menu panel (usually ISR@PRIM).
Change the network map name in the MCS.SIGNON initialization parameter.

Migrating existing STATS files

There are considerable indexing improvements in the Statistics facility for Version 6 to allowing for the extended search capabilities by NODE.

Due to this, it is recommend that when migrating to Version 6 that you preserve your previous stats files by performing an archive on the previous version of Connect:Direct. Then perform a Stats Cold start using STAT.INIT=COLD command with the new version of Connect:Direct.

The same recommendation applies when backing out or migrating back to the previous release, archive the data and then Cold start.

Migrate Stats data to Version 6

About this task

If Cold start is not possible and you must have the Stat data active then the following procedure **MUST** be performed when migrating to Version 6 with Stats data from a previous version.

Procedure

1. Edit, customize and submit the DGAJBKEY member in the HDGA600.SDGAJCL data set to rebuild each STAT.KSDS from every STAT.ESDS from the previous release that contains active data.
2. Ensure that the STEPLIB uses the HDGA600.SDGALINK to execute the version 6 DGADBKEY. For example, if you have 2 Stat file pairs and both ESDS01 and ESDS02 contain active data then you would need to execute DGADBKEY for both ESDS01 and ESDS02 to rebuild both KSDS01 and KSDS02.

Example

If there are 2 STAT file pairs and both ESDS01 and ESDS02 contain active data then execute DGADBKEY for both ESDS01 and ESDS02 to rebuild both KSDS01 and KSDS02.

Backout or Migrate STATS data back to Previous Version

About this task

To rollback to the previous release, data collected by Version 6 may not be compatible with that previous release and the following procedure must be performed to rollback to the previous version:

Procedure

1. Edit, customize and submit the DGAJBKEY member in the old version HDGAnnn.SDGAJCL data set to rebuild each STAT.KSDS from the STAT.ESDS from version 6 that contains active data.
2. Ensure that the STEPLIB uses the old version HDGAnnn.SDGALINK to execute the old version DGADBKEY.

Using the DGA#FXAL REXX Exec

After updating a target library type, you can use the DGA#FXAL REXX exec to check the target library to ensure the members are all present and all aliases are assigned correctly, and to report on any extra members or aliases that are not in the list for that type. The default setting simply checks the PDS members against the list and reports on what is found. In addition, DGA#FXAL can fix or delete broken aliases. This exec uses the TSO DELETE and RENAME commands, which require exclusive control of the PDS. Any errors that cannot be fixed automatically by DGA#FXAL must be fixed manually by restoring the PDS from a backup or distribution library. These errors include deleted members, non-alias members that are supposed to be aliases, and aliases that are supposed to be non-alias members.

Syntax and Parameters

The DGA#FXAL REXX exec has the following format:

```
DGA#FXAL    pds    [type | . ]    [detaillevel | . ]    [fixlevel | . ]
```

The DGA#FXAL REXX exec has the following parameters. The pds parameter is required and must be first. All parameters after the pds parameter are optional and positional, and can take the default by specifying a "." placeholder.

Parameter	Description
pds	A fully qualified DSN with or without quotation marks. A partitioned IBM Connect:Direct dataset of one of the supported types. A value of '?' requests that the syntax be displayed. The syntax is also displayed if there are any parameter errors.
type	The pds type. Either specify one of the target library types listed or do not specify. If not specified, the lowest level qualifier of the PDS must be one of the listed target library types. For each type, you can specify either the new or old value – they produce the same result.
detaillevel	Controls detail reporting. The default is 1. The following values are valid for this parameter: <ul style="list-style-type: none">• 0 reports everything, error and non-error alike.• 1 reports only error level 1 and higher.• 2 reports only error level 2 and higher

Parameter	Description
fixlevel	<p>Controls what fixes are attempted. Only levels 2–8 errors can be fixed. Default is 0. Valid values are 0, 2–8, and combinations of 2–8. The default is 0.</p> <p>The following values are valid for this parameter:</p> <ul style="list-style-type: none"> • 0 does not fix anything • 2 fixes level 2 errors. • 3 fixes level 3 errors • ... fixes level 4–7 errors • 8 fixes level 8 errors

For each pair or solo member listed for the type, an error level is set. See “Error Level Explanation” on page 20. If the error level is specified in fixlevel and the fix action in the table is not None, the pair is added to a fix list. After all pairs have been checked, if the fix list is not empty, you are prompted to continue. If you reply "Y," the fixes are attempted. If the fix for any pair fails, the exec continues with the next pair. A report is generated and written to both the TSO screen and to a PS dataset. The detail listings of expected and unexpected members are controlled by detaillevel. All fix attempts are reported.

Examples

1. In this example, IBM Connect:Direct uses list for type SDGAPROC (PROCESS), reports everything, and fixes all that can be fixed.

```
%DGA#FXAL DGA.SDGAPROC . 0 2345678
```

2. In this example, IBM Connect:Direct uses list for type SDGAOPLS (OPLIST); and as a result of using all defaults, reports only errors and fixes nothing.

```
%DGA#FXAL 'DGA.SDGAOPLS'
```

3. In this example, Connect:Direct again uses list for type SDGAOPLS (OPLIST) and all defaults. But in this scenario, the pds dsn still has the legacy low-level qualifier.

```
%DGA#FXAL SYS5.CD.OPLIST
```

4. This example is the same as the previous two examples but the pds dsn has an unsupported low level qualifier. The following example shows three different ways of coding this scenario.

```
%DGA#FXAL SYS5.CD.OPERLIST OPLIST
or
%DGA#FXAL SYS5.CD.OPERLIST SDGAOPLS
or
Exec 'DGA.SDGAISPC(DGA#FXAL)' 'SYS5.CD.OPERLIST SDGAOPLS'
```

Sample Reports

In this section, the following two scenarios are demonstrated:

- The first report shows an example of how DGA#FXAL REXX is used to show Connect:Direct errors without fixing anything.
- The second report shows how DGA#FXAL REXX is used with Connect:Direct to report everything in SYS5.CD.SDGAMAP and to fix all errors that can be fixed automatically.

The reports are broken into chunks to make them easier to read, and explanations of the reports follow the report samples.

Example–Show Errors But Do Not Fix Anything

DGA#FXAL : 5.1.1 PM47625 20 Oct 2011 14:21:06
 Command entered : %DGA#FXAL SYS5.CD.SDGAMAP
 Rexx exec source : CDZ5100.STG1.ISPCLIB(DGA#FXAL) via System-level CLIST DDNAME=SYSPROC
 Report dsn : USERID.CDFXAL.D2011293.T142106.LIST

Pds : SYS5.CD.SDGAMAP
 Type : MAPLIB
 DetailLevel : 1
 FixLevel : 0

Solo members : 7
 TrueNames : 6
 Aliases : 14
 Orphans : 12

Detail Report - Expected Members

Error Level	Fix Req	Expected Member	Actual Status	Expected Alias	Actual Status
8		DGAAAEER	NotInPDS	CDAER	Solo
7		DGAACPTR	NotInPDS	CDCPTR	Alias
6		DGAACTIV	NotInPDS	ACTIVITY	Orphan
9		DGAACR	Alias	CDCTR	TrueName
10		DGAADPTR	Alias	CDDPTR	NotInPDS
11		DGAADTR	Alias	CDDTR	Solo
1		DGAAEXEC	NotInPDS	EXCEPT	NotInPDS
12		DGAAFMCR	Alias	CDFMCR	Alias
13		DGAAFPTR	Alias	CDFPTR	Alias
2		DGAANPDS	TrueName	NPDSCOPY	NotInPDS
3		DGAAPDS	Solo	PDSCOPY	Orphan
14		DGAAPPSR	Alias	CDPSSR	Orphan
15		DGAAPTR	Orphan	CDPTR	NotInPDS
16		DGAARJTR	Orphan	CDRJTR	Solo
17		DGAARTTR	Orphan	CDRTTR	Alias
4		DGAARUNJ	Solo	RUNJOB	Alias
5		DGAARUNT	TrueName	RUNTASK	Solo
18		DGAASDCR	Orphan	CDSDCR	Orphan
19		DGAASEC	Orphan	SECURITY	Orphan
1		DGAASFR	NotInPDS	CDSFR	NotInPDS
1		DGAASTDC	NotInPDS	CDSTDCR	NotInPDS
1		DGAASUB	NotInPDS	SUBMIT	NotInPDS
1		DGAASUM	NotInPDS	SUMMARY	NotInPDS

Requestable fixes: 7

Detail Report - Unexpected Members

Error Level	Fix Req	Member Name	Actual Status
22		DGAAT7	Solo
23		DGAAT1	TrueName
23		DGAAT2	TrueName
24		DGAAT8	Alias
24		DGAAT1A	Alias
25		DGAAT4A	Orphan
25		DGAAT9	Orphan

Summary Report

Error Level	Fix Req	Count	Description	Fix Action
0		1	TRU and ALI or SOL are correct	None
1		5	TRU and ALI or SOL are not in PDS	None
2		1	TRU is a TrueName or Solo; ALI is not in PDS	Assign ALI
3		1	TRU is a TrueName or Solo; ALI is an orphan	Delete and re-assign ALI
4		1	TRU is a TrueName or Solo; ALI is alien alias	Delete and re-assign ALI
5		1	TRU and ALI are TrueNames or Solos	Delete and re-assign ALI
6		1	TRU is not in PDS; ALI is an orphan	Delete ALI
7		1	TRU is not in PDS; ALI is an alias	Delete ALI
8		1	TRU is not in PDS; ALI is a TrueName or Solo	Delete ALI
9		1	TRU is an alias of TrueName ALI	None
10		1	TRU is an alias; ALI is not in PDS	None
11		1	TRU is an alias; ALI is a TrueName or Solo	None
12		1	TRU and ALI are aliases of the same TrueName	None
13		1	TRU and ALI are aliases of different TrueNames	None
14		1	TRU is an alias; ALI is an orphan	None
15		1	TRU is an orphan; ALI is not in PDS	None
16		1	TRU is an orphan; ALI is a TrueName or Solo	None
17		1	TRU is an orphan; ALI is an alias	None
18		1	TRU and ALI are orphans of the same TrueName	None
19		1	TRU and ALI are orphans of different TrueNames	None
20			TRU is an alias; ALI is not expected	None
21			TRU is an orphan; ALI is not expected	None
22		1	Unexpected member found - Solo	None
23		2	Unexpected member found - TrueName	None
24		2	Unexpected member found - Alias	None
25		2	Unexpected member found - Orphan	None

DGA#FXAL ending normally

Example-Report Everything and Fix All Fixable Errors

```
DGA#FXAL      : 5.1.1 PM47625 20 Oct 2011 14:50:26
Command entered : %DGA#FXAL SYS5.CD.SDGAMAP . 0 2345678
Rexx exec source : CDZ5100.STG1.ISPCLIB(DGA#FXAL) via System-level CLIST DDNAME=SYSPROC
Report dsn      : EPETE1.CDFXAL.D2011293.T145026.LIST
```

```
Pds           : SYS5.CD.SDGAMAP
Type          : MAPLIB
DetailLevel   : 0
FixLevel      : 2345678
```

```
Solo members  : 7
TrueNames     : 6
Aliases       : 14
Orphans       : 12
```

Detail Report - Expected Members

Error Level	Fix Req	Expected Member	Actual Status	Expected Alias	Actual Status
8	Yes	DGAAAER	NotInPDS	CDAER	Solo
7	Yes	DGAACPTR	NotInPDS	CDCPTR	Alias
6	Yes	DGAACTIV	NotInPDS	ACTIVITY	Orphan
9		DGAACR	Alias	CDCTR	TrueName
10		DGAADPTR	Alias	CDDPTR	NotInPDS
11		DGAADTR	Alias	CDDTR	Solo
1		DGAAEXEC	NotInPDS	EXCEPT	NotInPDS
12		DGAAFMCR	Alias	CDFMCR	Alias
13		DGAAFPTR	Alias	CDFPTR	Alias
2	Yes	DGAANPDS	TrueName	NPDS COPY	NotInPDS
3	Yes	DGAAPDS	Solo	PDSCOPY	Orphan
14		DGAAPPSR	Alias	CDPSSR	Orphan
15		DGAAPTR	Orphan	CDPTR	NotInPDS
16		DGAARJTR	Orphan	CDRJTR	Solo
17		DGAARTTR	Orphan	CDRTTR	Alias
4	Yes	DGAARUNJ	Solo	RUNJOB	Alias
5	Yes	DGAARUNT	TrueName	RUNTASK	Solo
18		DGAASDCR	Orphan	CDSDCR	Orphan
19		DGAASEC	Orphan	SECURITY	Orphan
1		DGAASFR	NotInPDS	CDSFR	NotInPDS
1		DGAASTDC	NotInPDS	CDSTDCR	NotInPDS
1		DGAASUB	NotInPDS	SUBMIT	NotInPDS
1		DGAASUM	NotInPDS	SUMMARY	NotInPDS
0		DGAAWTOS	TrueName	CDFWTOST	Alias

Requestable fixes: 7

Detail Report - Unexpected Members

Error Level	Fix Req	Member Name	Actual Status
22		DGAAT7	Solo
23		DGAAT1	TrueName
23		DGAAT2	TrueName
24		DGAAT8	Alias
24		DGAAT1A	Alias
25		DGAAT4A	Orphan
25		DGAAT9	Orphan

Summary Report

Error Level	Fix Req	Count	Description	Fix Action
0		1	TRU and ALI or SOL are correct	None
1		5	TRU and ALI or SOL are not in PDS	None
2	Yes	1	TRU is a TrueName or Solo; ALI is not in PDS	Assign ALI
3	Yes	1	TRU is a TrueName or Solo; ALI is an orphan	Delete and re-assign ALI
4	Yes	1	TRU is a TrueName or Solo; ALI is alien alias	Delete and re-assign ALI
5	Yes	1	TRU and ALI are TrueNames or Solos	Delete and re-assign ALI
6	Yes	1	TRU is not in PDS; ALI is an orphan	Delete ALI
7	Yes	1	TRU is not in PDS; ALI is an alias	Delete ALI
8	Yes	1	TRU is not in PDS; ALI is a TrueName or Solo	Delete ALI
9		1	TRU is an alias of TrueName ALI	None
10		1	TRU is an alias; ALI is not in PDS	None
11		1	TRU is an alias; ALI is a TrueName or Solo	None
12		1	TRU and ALI are aliases of the same TrueName	None
13		1	TRU and ALI are aliases of different TrueNames	None
14		1	TRU is an alias; ALI is an orphan	None
15		1	TRU is an orphan; ALI is not in PDS	None
16		1	TRU is an orphan; ALI is a TrueName or Solo	None
17		1	TRU is an orphan; ALI is an alias	None
18		1	TRU and ALI are orphans of the same TrueName	None
19		1	TRU and ALI are orphans of different TrueNames	None
20			TRU is an alias; ALI is not expected	None
21			TRU is an orphan; ALI is not expected	None
22		1	Unexpected member found - Solo	None
23		2	Unexpected member found - TrueName	None
24		2	Unexpected member found - Alias	None
25		2	Unexpected member found - Orphan	None

Do you wish to attempt the above 7 fixes?
Enter YES to attempt, NO to skip:
Reply is Y

Fix Report

Requested fixes: 7
Successful fixes: 7
Failed fixes: 0

Error Level	Expected TrueName	Expected Alias	Del RC	Ren RC	Message
8	DGAAER	CDAER	0		ALI deleted
7	DGAACPTR	CDCPTR	0		ALI deleted
6	DGAACTIV	ACTIVITY	0		ALI deleted
2	DGAANPDS	NPDS COPY		0	ALI assigned
3	DGAAPDS	PDSCOPY	0	0	ALI deleted and re-assigned
4	DGAARUNJ	RUNJOB	0	0	ALI deleted and re-assigned
5	DGAARUNT	RUNTASK	0	0	ALI deleted and re-assigned

DGA#FXAL ending normally

Report Explanation

The DGA#FXAL report contains the following information.

Line Number / Field / Report Section	Description
Line 1	The release level and maintenance level of DGA#FXAL, as well as the current date and time.
Line 2	The TSO command the user entered to invoke DGA#FXAL.
Line 3	The REXX exec source dataset and the ALTLIB DISPLAY line for the DGA#FXAL invocation.
Line 4	The DSN of the report dataset.
Utility Parameter Options	Taken either from the DGA#FXAL parameters or the defaults.
Total Counts per PDS Type	<p>The total counts for each of the four types of members in the PDS, where:</p> <ul style="list-style-type: none"> • Solo is a member without an alias • TrueName is a member with an alias • Alias is an alias that points to a TrueName • Orphan is an alias that points to a pds member that has been deleted
Detail Report - Expected Members	<p>This report is generated if at least one entry passes detaillevel filtering. For each entry in the DGA#FXAL exec list for the pds type, the actual status of the one or two member names is determined, and an error level is set based on the results. The error level is compared to detaillevel, and if it is greater than or equal to it, the detail line is written.</p> <p>For each detail line, the following information is provided:</p> <ul style="list-style-type: none"> • Error Level—A numeric error level. For more information, see “Error Level Explanation” on page 20. • Fix Req—Yes means a fix was requested for this error level. Else empty. • Expected Member—The first or only member name in the list entry. • Actual Status—One of the four PDS member types or “NotInPds” • Expected Alias—The second member name in the list entry, or empty • Actual Status—One of the four PDS member types, or “NotInPds”, or empty. <p>The total number of Expected Member detail lines with Yes in the Fix Req column.</p>
Detail Report - Unexpected Members	<p>This report is generated if at least one member exists in the pds that is not in the DGA#FXAL list for the pds type. An error level is set based on what kind of pds member it is.</p> <p>For each detail line, the following information is provided:</p> <ul style="list-style-type: none"> • Error Level—A numeric error level. For more information, see “Error Level Explanation” on page 20. • Fix Req—Always empty. • Member Name—The unexpected member name • Actual Status—One of the four PDS member types.

Line Number / Field / Report Section	Description
Summary Report	<p>This report is always generated.</p> <p>For each possible error level, the following information is provided:</p> <ul style="list-style-type: none"> • Error Level—A numeric error level. For more information, see “Error Level Explanation” on page 20. • Fix Req—Yes means a fix was requested for this error level • Count—The number of times this error occurred • Description—A short description of the error. See below • Fix Action—The action that will be taken to fix the error if Fix Req is Yes
Attempt to Fix section	<p>If there are any fixable errors, the attempt-to-fix prompt and the user's response are given.</p>
Fix Report	<p>If the response was 'Y', the Fix report is generated, starting with lines giving the number of fixes requested, successful, and failed, followed by one fix detail line for each fix attempted.</p> <p>Each detail line provides the following information:</p> <ul style="list-style-type: none"> • Error Level—A numeric error level. For more information, see “Error Level Explanation” on page 20. • Expected TrueName—The first member name in the list entry • Expected Alias—The second member name in the list entry • Del RC— The TSO DELETE command reason code, if it was attempted • Ren RC—The TSO RENAME command reason code, if it was attempted • Message—A description of the results of the attempted fix action
Utility execution status	<p>Indicates if the DGA#FXAL ended normally or provides an error message indicating where the utility had a problem in execution, for example, DGA#FXAL ending - cannot ALLOC &pds with DISP=OLD where the prefix '&' denotes a variable, which will be substituted in the actual message.</p>

Error Level Explanation

For all error level descriptions, the following meanings are in effect:

- SOL—The member name in a list entry that has only one member name
- TRU—The 1st member name in a list entry that has two member names
- ALI—The 2nd member name in a list entry that has two member names
- Solo—A member in pds that has no alias
- TrueName—A member in pds that has one or more aliases
- Alias—A member in pds that is an alias of a TrueName
- Orphan—A member in pds that once was an Alias, but its TrueName no longer exists
- Not in PDS—The member was not found in PDS

Error Level	Description
0	TRU and ALI or SOL are correct. For a list entry with TRU and ALI, TRU is a TrueName and ALI is its Alias in pds. For a list entry with SOL, SOL is a non-alias member of pds. No fix is needed.
1	TRU and ALI or SOL are not in PDS. For a list entry with TRU and ALI, neither one was found in pds. For a list entry with SOL, SOL was not found in pds. No automatic fix is possible.
2	TRU is a TrueName or Solo; ALI is not in PDS. For a list entry with TRU and ALI, TRU was found in pds, and it may have one or more aliases, but ALI is not in pds. The fix is to assign ALI as an alias of TRU.
3	TRU is a TrueName or Solo; ALI is an orphan. For a list entry with TRU and ALI, TRU and ALI were both found in pds, but though TRU may have one or more aliases, ALI is not one of them. ALI is an alias of a member that no longer exists. The fix is to delete ALI, then re-assign it as an alias of TRU.
4	TRU is a TrueName or Solo; ALI is alien alias. For a list entry with TRU and ALI, TRU and ALI were both found in pds, but though TRU may have one or more aliases, ALI is not one of them. ALI is an alias of another (i.e. an alien) TrueName. To find out what that alien TrueName is, invoke the TSO command "LISTDS pds MEM" and look for ALI in each alias list. The fix is to delete ALI, then re-assign it as an alias of TRU.
5	TRU and ALI are TrueNames or Solos. For a list entry with TRU and ALI, TRU and ALI were both found in pds. TRU may have one or more aliases, but ALI is not one of them. ALI is a Solo or TrueName member. The fix is to delete ALI, then re-assign it as an alias of TRU. Note: if ALI has any aliases, they will be orphaned.
6	TRU is not in PDS; ALI is an orphan. For a list entry with TRU and ALI, TRU was not found in pds. ALI is an alias of a member that no longer exists. The fix action is to delete ALI, though that only cleans up the orphan alias. Automatic recovery of the member is not possible.
7	TRU is not in PDS; ALI is an alias. For a list entry with TRU and ALI, TRU was not found in pds. ALI is an alias of another TrueName. The fix action is to delete ALI, though that only cleans up the alias. Automatic recovery of the member is not possible.
8	TRU is not in PDS; ALI is a TrueName or Solo. For a list entry with TRU and ALI, TRU was not found in pds. ALI is a Solo or TrueName member. The fix is to delete ALI, though that only cleans up the incorrect Solo or TrueName member. Note: if ALI has any aliases, they will be orphaned. Automatic recovery of the member is not possible.
9	TRU is an alias of TrueName ALI. For a list entry with TRU and ALI, TRU and ALI were both found in pds. ALI should be the alias and TRU should be the TrueName but, instead, their relationship is reversed. No automatic fix is supported.
10	TRU is an alias; ALI is not in PDS. For a list entry with TRU and ALI, TRU is an alias instead of a TrueName, and ALI was not found. No automatic fix is supported.
11	TRU is an alias; ALI is a TrueName or Solo. For a list entry with TRU and ALI, TRU is an alias instead of a TrueName, and ALI is a TrueName or Solo, instead of an alias of TRU. TRU is not an alias of ALI. No automatic fix is supported.
12	TRU and ALI are aliases of the same TrueName. For a list entry with TRU and ALI, TRU is an alias instead of a TrueName, and ALI is an alias of the same TrueName as TRU. No automatic fix is supported.

Error Level	Description
13	TRU and ALI are aliases of different TrueNames. For a list entry with TRU and ALI, TRU is an alias instead of a TrueName, and ALI is an alias of a different TrueName. No automatic fix is supported.
14	TRU is an alias; ALI is an orphan. For a list entry with TRU and ALI, TRU is an alias instead of a TrueName, and ALI is an alias of a member that no longer exists. No automatic fix is supported.
15	TRU is an orphan; ALI is not in PDS. For a list entry with TRU and ALI, TRU is an alias of a member that no longer exists instead of a TrueName, and ALI was not found in pds. No automatic fix is supported.
16	TRU is an orphan; ALI is a TrueName or Solo. For a list entry with TRU and ALI, TRU is an alias of a member that no longer exists instead of a TrueName, and ALI is a member with zero or more aliases instead of an alias of TRU. No automatic fix is supported.
17	TRU is an orphan; ALI is an alias. For a list entry with TRU and ALI, TRU is an alias of a member that no longer exists instead of a TrueName, and ALI is an alias of another TrueName instead of an alias of TRU. No automatic fix is supported.
18	TRU and ALI are orphans of the same TrueName. For a list entry with TRU and ALI, TRU and ALI are aliases of a single member that no longer exists instead of a TrueName. No automatic fix is possible.
19	TRU and ALI are orphans of different TrueNames. For a list entry with TRU and ALI, TRU is an alias of a member that no longer exists and ALI is an alias of a different member that no longer exists. No automatic fix is supported.
20	TRU is an alias; ALI is not expected. For a list entry with TRU only, TRU is an alias instead of a TrueName. No automatic fix is supported.
21	TRU is an orphan; ALI is not expected. For a list entry with TRU only, TRU is an alias of a member that no longer exists instead of a TrueName. No automatic fix is supported.
22	Unexpected member found - Solo. A Solo member was found in pds that does not appear in any list entry. No automatic fix is supported. This may or may not be an error.
23	Unexpected member found - TrueName. A TrueName member was found in pds that does not appear in any list entry. No automatic fix is supported. This may or may not be an error.
24	Unexpected member found - Alias. An alias was found in pds that does not appear in any list entry. No automatic fix is supported. This may or may not be an error.
25	Unexpected member found - Orphan. An alias of a member that no longer exists was found in pds that does not appear in any list entry. No automatic fix is supported.

Migrating CICS from Prior Releases

About this task

CICS uses different program names, map names, and transaction IDs. Refer to *Activate the CICS Component* in *IBM Connect:Direct for z/OS CICS Administration and User Guide* before continuing.

To use the current CONFIG file contents:

Procedure

1. Shut down the IBM Connect:Direct interface in the CICS region.

2. To close the CONFIG file so that the buffers are committed and written, issue CICS commands, for example, CEMT SET FILE(CONFIG) CLOSE.
3. Using IDCAMS, perform a REPRO of the CONFIG file to a new CONFIG file.
4. To reactive CICS, restart the API interface.
5. In a different CICS system, once the RDO updates have been applied and the Connect:Direct for z/OS Version 5.1 CICS modules loaded, run the Administrator transaction (DGAA) and go to the Configuration screen.

Go to the Control Record screen to set the Transaction codes to reflect the changes made to the RDO source \$CD.SDGACNTL(DGACCSD).

Enter the CDA transaction and start the interface.

6. Complete the CICS panels following the instructions in the procedure *Build the CICS Configuration File through ISPF* in *IBM Connect:Direct for z/OS CICS Administration and User Guide*.

Upgrading Connect:Direct File Agent

To take advantage of the command line in this version of Connect:Direct File Agent, modify the Java command line parameters after you install it.

If you are upgrading from an existing version:

Note: Files recorded in the checkpoint file are treated as if the new command line parameters had been active when the files were first discovered. You do not have to delete the checkpoint file.

1. Install Connect:Direct File Agent into an empty directory.
2. Copy the .ser and .ckpt files from the old to the new installation directory.
3. Rename the old installation directory to an archive name.
4. Rename the new installation directory to the primary name.

Chapter 4. Configuration Guide

Before you Install

Installation is easier and more effective if you complete your planning before you begin.

Before you begin your installation:

1. Read the IBM Connect:Direct for z/OS Release Notes for the latest product information.
2. Verify hardware and software requirements. Review [Installation Requirements](#) for hardware and software requirements.
3. Determine whether your system uses a Multi-Image Manager (MIM) or Global Resource Serialization (GRS) system.

Installation Requirements

Connect:Direct for z/OS requires the following hardware and software. However, this is not a complete list. See the Program Directory for the complete list of requirements, and for more information on several of the items listed below:

- Authorized Library - Connect:Direct for z/OS must run from an APF-authorized library. If it is not authorized, you receive message SITA117I during initialization, and it terminates.
- Connect:Direct for z/OS VSAM space requirements - For information, see [VSAM Files DASD Requirement and Description](#)
- High-Level Assembler - High-Level Assembler is required to assemble the sample Connect:Direct for z/OS exits in the SDGASAMP dataset.
- TSO with ISPF/PDF - You must have IBM Time Sharing Option (TSO) with ISPF version 6.0 or later.
- SNA Network Requirements - Network support for Connect:Direct for z/OS requires both of the following: VTAM Communications Server and IBM Advanced Communications Functions for Network Control Program (NCP)
- IBM Connect:Direct Spool Transfer - If you use the Spool Transfer feature for outbound spool transfers, you must have LRS (Levi, Ray, and Schoup, Inc.) VTAM Printer Support System (VPS) Release 6.2, Version 1.12 or later and LRS VPS/CDI Option installed. No additional software is required for inbound transfers.
- The CICS Interface - If you use the CICS interface, you must have at least 200 tracks of 3390 disk space available and be using CICS/TS version 3.01.00 or higher.
- All other requirements as listed in the Program Directory for IBM Connect:Direct for z/OS.

VSAM Files DASD Requirement and Description

Use IBM Connect:Direct VSAM files during testing, and migrate previous versions of the files as part of the product release implementation.

Note: If you use VSAM cache utilities, remove IBM Connect:Direct VSAM files from their control or unpredictable results may occur.

IBM Connect:Direct uses VSAM files to control and monitor execution. The following table describes the files and space requirements built during the configuration process. These values represent the minimum space requirement for each VSAM file.

dataset	DSORG	Approx. Install Supplied Size	Number of Entries	Approx. Minimum Size	Minimum Primary Entries	BLK SIZE (CI)
AUTH	KSDS	132 KB	100	88 KB	5	4096
CKPT	KSDS	600 KB	40	44 KB	5	4096
MSG	KSDS	3 MB	12000	3 MB	12000	4096
NETMAP	KSDS	88 KB	50	88 KB	2	4096
STATS archive directory ¹	ESDS	88 KB	500	88 KB	500	4096
STATS index1	KSDS	222 KB	5000	88 KB	750	4096
STATS index2 ²	KSDS	222 KB	5000	88 KB	750	4096
STATS log1	ESDS	1.8 MB	6750	197 KB	1000	4096
STATS log2 ²	ESDS	1.8 MB	6750	197 KB	1000	4096
TCQ	RRDS	200 KB	100	44 KB	21	1536
TCX	RRDS	44 KB	1	44 KB	1	1024
TYPE	KSDS	88 KB	20	88 KB	5	4096
CONFIG ³	KSDS	44 KB	1	44 KB	1	2048
USRPROF ³	KSDS	144 KB	1	144 KB	1	4096
EVENT ³	KSDS	44 KB	1	44 KB	1	2048
TOTAL		8.6 MB		4.4 MB		
<p>1. Optional dataset.</p> <p>2. The default configuration for the Statistics facility uses two statistics file pairs, which requires four VSAM files. Refer to “Statistics Files” on page 28 below for details about the VSAM files.</p> <p>3. Optional IBM Connect:Direct-CICS IUI facility file.</p>						

Authorization File (AUTH)

The Authorization file, a VSAM KSDS, can contain a record for each authorized IBM Connect:Direct user. If you use the IBM Connect:Direct Authorization Facility, calculate the size of your Authorization file requirements using the following formula:

$\text{Number of IBM Connect:Direct Users} \times 100 \text{ Bytes} = \text{Size of Authorization File}$
--

If you are not using the IBM Connect:Direct Authorization Facility, you still define an Authorization file.

Checkpoint File (CKPT)

The Checkpoint file is a VSAM KSDS that contains checkpoint information generated by the DTF during execution of a copy operation. The Checkpoint file consists of variable length records, one per Process that has checkpointing specified. The average record length is 256 bytes. The equivalent of one cylinder of space is allocated during the installation. The size of the Checkpoint file can be influenced by the number of days you retain checkpoint information according to the initialization parameter, CKPT.DAYS.

IBM Connect:Direct for z/OS Message File (MSG)

The Message file, a VSAM key-sequenced dataset (KSDS), holds all messages, except ISPF panel-related messages, used by IBM Connect:Direct. Each message record contains the issuing module name, short message text, and message explanation.

Network Map File (NETMAP)

The network map file is a VSAM KSDS that contains network definition information, including the network names for the local node and other (adjacent) nodes in the network, the communication addresses for use by the API, and various control information used by IBM Connect:Direct

To estimate the size of the network map, you must determine how many nodes are needed and the communications used for each node. This includes determining the node IDs for your network (including the local node name), all APPLIDs (SNA type), and all TCP/IP addresses (or DNS names) for the nodes.

The following chart defines the basic size of records for planning how large to make the network map:

Record	Size (in bytes)	Description/Comments
Control Records	250	Base records for NDM control
ADJACENT.NODE	124	Basic adjacent node information
ALT.COMM	32 + 59(n)	Alternate communications addresses. Each set adds 1 to “n”.
LDNS	278	Holder of the DNS name for an ADJACENT.NODE
CONTACT/ COMMENTS	74 + x	1 per node definition and each segment (x) adds up to: Conact Name - 40, Phone - 40, Description - 255
IPv6	232	This record exists for an ADJACENT.NODE if the node is defined with any of the following items specified: <ul style="list-style-type: none">• IPv6 address for TCP/IP (or UDT)• the API record specifies an IPv6 record• SOURCEIP is specified.
UDT	96	This record exists for an ADJACENT.NODE if the node is defined as UDT33.
TCP API	60	Used to specify the address for API
APPLIDs	40 + 8x	SNA APPLIDs for a node. Add 1 to x for each node ID specified
xNODE.LUS	28 + 8(x)	If either PNODE or SNODE.LUS is specified, add 1 to x for each node name specified.

For example, if the network map contains 25 nodes with API APPLIDs in each of the 25 nodes, it requires approximately the space illustrated in the following table.

Netmap Definitions	Number	Size (in bytes)
Control Records + Local node	3	362
ADJACENT.NODE	25	3100
APPLIDs	25 nodes with 20 APPLIDs	5350
Total	53 records	8812

For those installations that make frequent updates to the network map, it is recommended that the allocation be done on a cylinder boundary and be at least one cylinder more than currently needed. CA/CI splits may take place as additions and updates are done.

Statistics Directory of Archive Files

Allocate the STATS Archive Directory file if you plan to archive statistics records and maintain a directory of the archive files. Archiving is the process of copying statistics records from the IBM Connect:Direct statistics files to other datasets for long-term storage.

Use the directory to track the dataset names of the archive files, and the date and time range of the statistics records the archive files contain. Each record in the directory contains information about a single archive file. Therefore, the value of the RECORDS parameter that defines the directory determines how many archive files can be represented in the directory. Connect:Direct for z/OS provides facilities for maintaining and displaying the directory through the INQUIRE STATDIR command.

Statistics Files

The Statistics facility logs statistics to VSAM file pairs. Each file pair consists of the following:

- A VSAM entry-sequenced cluster
- A VSAM key-sequenced cluster

The default and minimum configuration uses two such file pairs, making four VSAM files. The maximum number of file pairs you can use is twenty. Specify the number of file pairs and the VSAM cluster names with the STAT.DSN.BASE and STAT.FILE.PAIRS initialization parameters.

The RECORDS parameter of the Access Methods Services DEFINE command for the ESDS cluster specifies the maximum number of records each file pair can contain. The maximum number of statistics records available is the sum of the values in the RECORDS parameters for all the statistics ESDS clusters. For example, if you determine that your system needs space for 15,000 statistics records, you can define three file pairs containing 5,000 records each, or two file pairs containing 7,500 records each.

Note: IBM Connect:Direct does not support extended-format, extended-addressing ESDS Statistics datasets.

Within each file pair, IBM Connect:Direct writes statistics records to the ESDS cluster. IBM Connect:Direct uses the KSDS cluster to maintain index information about the records in the ESDS. Whenever either file of the pair becomes full, the entire pair is considered full, and the system initiates a switch to the next pair in the sequence. Because the file pair is full when one file fills to capacity, it is important that you size the ESDS and KSDS in a file pair appropriately, relative to each other. Specifying appropriate file sizes reduces the waste of excessive space.

Statistic File Recommendation

The following tables describes minimum, optimum and maximum values for the VSAM attributes that impact the statistic file space and its performance.

CI Size	Minimum Size	Optimum Size	Maximum Size
ESDS (D)	2048	32768	32768
KSDS (D)	4096	4096	32768
KSDS (I)	2560	4096	32768

BUFSP (Cluster)	Minimum Size	Optimum Size	Maximum Size
ESDS (D)	2 * CI size (D)	2 * CI size (D)	Storage Limit
KSDS (D)	2 * CI size (D) + 1 * CI size (I)	2 * CI size (D) + 1 * CI size (I)	Storage Limit

Note: The actual maximum BUFSP is limited by your system's storage constraints. DEFINE allows you to specify BUFSP(16000000), but most programs would ABEND when they OPEN the file.

Note: Decreasing the BUFSP below the optimum size can cause significant performance degradation. Increasing the BUFSP far above the optimum provides no noticeable gain in performance. However, increasing it a little above the optimum provides some extra headroom.

The frequency with which IBM Connect:Direct writes records to the KSDS cluster of a file pair depends on usage patterns at each site and the ESDS CI size. The greater the ESDS CI size, the fewer records IBM Connect:Direct writes to the KSDS. However, a larger ESDS CI size also requires more statistic file buffer storage in the IBM Connect:Direct address space.

The following table summarizes results from running the KSDS rebuild program against ESDS clusters of various CI sizes. Each ESDS cluster was defined with the optimum BUFSP for its CI size, and each cluster contained identical statistic records. These results are suggestive of the performance differences due to ESDS CI size which the DTF will experience when writing statistics records and reading them for the SELECT STATISTICS command.

ESDS records	KSDS CI size Data/Index	ECDS CI Size	KSDS rebuild records (CD 5.2)	KSDS rebuild records (CD 6.0)	KSDS rebuild time (CD 5.2)	KSDS rebuild time (CD 6.0)
1,104,668	4K/4K	4K	705,573	1,360,564	08:26.46	15:33.28
1,104,668	4K/4K	8K	444,153	801,803	04:27.59	08:55.59
1,104,668	4K/4K	16K	282,623	477,526	02:57.60	05:28.93
1,104,668	4K/4K	32K	185,465	288,592	01:41.31	03:15.31

The default configuration provides space for 13,500 statistics records evenly divided between two file pairs. The minimum configuration provides space for 2,000 statistics records. The amount of activity in the IBM Connect:Direct system determines how frequently the statistics file pair list wraps around. It is recommended that you estimate your activity rate and allocate enough space so that the system records several days of records before a wraparound occurs. After running IBM Connect:Direct for several days, you may need to adjust the number or sizes of the statistics files to allocate the right amount of space.

Transmission Control Queue (TCQ) and Index (TCX)

The Transmission Control Queue (TCQ) is a VSAM RRDS file that stores Processes that are:

- Executing
- Queued for execution
- Held for retries
- Held for future execution
- Retained for reporting

The TCQ index (TCX) is also a VSAM RRDS file. It consists of a record that is a map for controlling the allocation of space in the TCQ.

The control interval (CI) size of the TCQ is from 1,536 to 30,720 bytes. Each Process defined in the TCQ occupies one or more CIs, depending on the number of statements within the Process. Because TCQ access characteristics prohibit secondary allocation, be sure to allocate enough space for the maximum number of potential Processes when you define the TCQ.

The TCX controls space use and access to the TCQ. The TCX average and maximum record sizes should be defined as the CISIZE - 7. To maximize the number of usable TCQ CIs, allocate a record size of 30,713 and a CI of 30,720 to the TCX since it occupies only one track. IBM Connect:Direct will use the number of CIs in the TCQ (rounded down to a multiple of 8) or 65536, whichever is smaller.

The largest TCQ that can be allocated and used is 2621 cylinders, on 3390 DASD device, with a CI size of 30,720 and a record size of 30,713. This size can accommodate 65,520 Processes in the queue if no single Process exceeds 30,713 bytes in its internal format.

The following formula shows the number of CIs that can be controlled:

$$\text{Maximum number of TCQ CIs} = ((\text{TCX Max Record Size} - 12) / 2) * 8$$

In the default TCX definition, the definition of the TCX CI Size is 1,024 bytes; maximum record size is 1,017. Substituting the default maximum record size of 1,017 bytes in the formula results in 4,016 TCQ CI records, as shown in the following calculation:

$$\text{Maximum number of TCQ CIs} = ((1,017 - 12) / 2) * 8 = 4,016$$

Using the default TCX definition, you can define the TCQ with up to 4,016 records, thereby allowing the TCQ to hold up to 4,016 Processes, depending on the number of statements in each Process. However, the default definition for the TCQ specifies 1,000 TCQ CIs but actually holds only about 500 simple one-step Processes).



CAUTION: Depending on your hardware configuration and load, it can take a long time to warm start IBM Connect:Direct with thousands of Processes in the TCQ. In general, it takes about 1 minute for every 1000 Processes.

Note: If you plan to use the Process retention feature, the size of the TCQ and TCX datasets should be increased, in addition to the factors you consider when determining space requirements for datasets.

Enlarging the TCQ

To enlarge the TCQ:

Procedure

1. **AVERAGE PROCESS SIZE.** Determine the size of the average Process for your site. To do this, you need to know the type and average number of steps in your average Process. Use the following table to calculate the TCQ space needed for your average Process. Take the number of steps and multiply that by the number of bytes for the Process component, then add the number of bytes for a Process header (1,616). If you do not know what the average Process looks like, assume that the average Process contains five COPY steps:

Process Component	Number of Bytes
Process Header	1,616
COPY Step	1,024
RUN TASK	128
RUN JOB	176
GOTO Step	96
EXIT Step	96
IF Statement	208

For example, if the average Process contains five COPY steps, the space required would be $1,616 + (5 * 1,024) = 6,736$ bytes.

Note: The largest Process can contain up to 1 MB and must fit within 43 TCQ records. A TCQ CISIZE of 24 KB is sufficient to hold the maximum size Process.

2. **TCQ CISIZE.** Calculate the appropriate TCQ CI size that should be defined for the average Process. The larger the TCQ CI size, the fewer I/O operations have to be done to read/write the TCQ entry. Ideally, the average Process should fit in a single TCQ entry, but if the average Process is greater than 30 KB, then select a TCQ CI size of 30 KB.
3. **TCQ CIs PER PROCESS.** Calculate the number of TCQ CIs required for a single Process. For example, if the average Process is 36 KB and you use a TCQ CI size of 30 KB, then each Process takes 2 TCQ CIs.

4. **NUMBER OF PROCESSES.** Calculate the number of Processes you expect to be in the TCQ at any one time, including Processes that are waiting for a connection or their turn to execute, Processes that are executing, Processes that have been held, and Processes that are going through retry. Double or triple this number, then multiply this number by the number of TCQ CIs required for each Process. For example, if you think that you'll have no more than 100 Processes in the queue at any one time, use 200 or 300 for the number and multiply it by the number of TCQ CIs required by each Process.
5. **TCQ RECORDS.** The result of this calculation is the number of records and CISIZE that should be defined in the TCQ VSAM definition. The average and maximum record size should be the CISIZE - 7.
6. **TCX CISIZE.** Now, make sure that the TCX definition will be able to map this number of TCQ entries. Take the number of records from Step 5, divide by 4, and add 12. Then, to obtain the average and maximum record size for the TCX VSAM definition, round this number up to the next valid CI Size (minus 7)

Assume that the average Process consists of 10 COPY steps, 10 IF statements and 10 RUN TASK statements and you want to allocate space to hold 1,000 of these Processes:

AVERAGE PROCESS SIZE: $1,616 + (10 * 1,024) + (10 * 208) + (10 * 128) = 15,216$

TCQ CISIZE: 16,384 (good value that would hold a complete average Process)

TCQ CIs PER PROCESS: 1

NUMBER OF PROCESSES: 1,000

TCQ RECORDS: 1,000

TCX CISIZE: 512 (this would map up to $((512 - 7 - 12) / 2 * 8) = 1,968$ TCQ RECORDS)

Note: If you change the TCX or TCQ definitions, you must COLD start the TCQ.

Type File (TYPE)

The Type file is a VSAM KSDS that consists of records containing file attribute defaults for the destination file allocations specified in the IBM Connect:Direct Copy statement. For information on the Type file contents, see *Maintaining the Type File* in the *IBM Connect:Direct for z/OS Administration Guide*.

CICS Files

The following three files are optional IBM Connect:Direct-CICS IUI facility Files:

- **CICS Configuration File (CONFIG)** - The Configuration file is used by the CICS IUI facility of IBM Connect:Direct-CICS only. It is a VSAM KSDS file. One file exists and is allocated for each CICS region. It is primed during IBM Connect:Direct-CICS installation and updated online through IBM Connect:Direct-CICS administrator functions. This file contains system parameters that control the IBM Connect:Direct-CICS environment. It also contains information about the IBM Connect:Direct nodes available to the IBM Connect:Direct-CICS and their network map definitions. Installed with default dataset name \$CDVAM.CONFIG.
- **CICS User Profile File (USRPROF)** - The User Profile or Signon Defaults file is allocated to the CICS region. It is a VSAM KSDS file with the CICS Userid as key. It is updated using the signon defaults function and used to set up auto-signon to IBM Connect:Direct Installed with default dataset name \$CDVAM.USRPROF
- **CICS Event Restart File (EVENT)** - The Event Restart file is used by the Event Services Support feature of IBM Connect:Direct. One file exists for each CICS system and is allocated to the CICS region. It is updated by the Event Services Support function, and is used for restarting ESS. Installed with default dataset name \$CDVAM.EVENT

Virtual Storage Requirements

Connect:Direct for z/OS executes with a REGION of 0 MB allocated for most environments. Using the default limits for the IEFUSI exit, IBM Connect:Direct may have enough virtual storage both above and below the line to run many Processes concurrently. However, your results may vary depending on the data type, block sizes, compression, communication buffer sizes, and other factors. IBM Connect:Direct

storage requirements also depend on the initialization parameters that you specify and the type of Process work being performed. In some cases, you may need to increase the size of the REGION or storage limits.

Note: Connect:Direct for z/OS may use above-the-bar storage. The in-storage trace table defaults to 2 MB above the bar storage, and is controlled by the TRACE.BUFFER initialization parameter. Each zFBA COPY step uses 32 MB of page-fixed above-the-bar storage for the duration of the step.

Specifying REGION=0M eliminates the need to determine the REGION needed, and does not use any more virtual storage than specifying just the needed amount.

The following initialization parameters affect storage allocation below and above the 16 MB line:

Parameters	Description
MAXUSER (default 6) MAXPRIMARY (default 6) MAXSECONDARY (default 6) MAXPROCESS (default 12)	These parameters determine the number of tasks that IBM Connect:Direct supports. Storage is obtained during initialization and remains for the duration of the JOB or started task. For each task, approximately 2 KB is allocated above the line and 1 KB below the 16 MB line.
V2.BUFSIZE (default 32K,128K)	The first positional parameter specifies the default maximum buffer size that IBM Connect:Direct uses for LU6.2 and TCP/IP data transmission. The default is 32K. (K means thousands of bytes.) The second positional parameter is used to alter the TCP/IP send and receive buffer sizes within TCP/IP. In general terms, the second positional parameter should be at least the same and not less than the first parameter and should be big enough to handle the largest V2.BUFSIZE override from the netmap. A good common practice is to have the second parameter be a multiple of the first parameter and at least twice or more than the first parameter.
TCP (default NO)	The TCP parameter affects the program storage required. This parameter allocates approximately 3280 KB above and 112 KB below the 16 MB line.
UDT (default NO)	The UDT parameter requires 4 MB above the line. Each concurrent UDT process requires an additional 50 MB above the line.
ALLOCATION.EXIT RUN.JOB.EXIT RUN.TASK.EXIT SECURITY.EXIT STATISTICS.EXIT	These exits are loaded at initialization and reside below the 16 MB line. The default value for these parameters is no exit. Samples are provided and described in the <i>IBM Connect:Direct for z/OS Administration Guide</i> .
STAT.QUEUE.ELEMENTS (default 1500)	Stat queue elements are 2 KB each and allocated above the 16 MB line.
MAXSTGIO (default 1M,1M)	This parameter limits the amount of I/O buffers for each COPY. The I/O buffers are allocated above the 16 MB line.

The following example shows reasonable settings for these initialization parameters:

```
MAXUSER=26
MAXPRIMARY=50
MAXSECONDARY=50
MAXPROCESS=100
MAXSTGIO=(1M,1M)
V2.BUFSIZE=32K
TCP=OES
ALLOCATION.EXIT=exitname
SECURITY.EXIT=exitname
STATISTICS.EXIT=exitname
STAT.QUEUE.ELEMENTS=999
TRACE.BUFFER = 2 (default)
```

Storage Requirements in a IBM Connect:Direct Plex Environment

In a IBM Connect:Direct Plex environment, additional storage is allocated based on the number of servers that the IBM Connect:Direct Manager can support. The maximum number of servers that a IBM Connect:Direct manager can support is 32. The IBM Connect:Direct Manager allocates approximately 510K above the 16 MB line for each server, regardless of whether the server is active or not.

Note: The IBM Connect:Direct Manager allocated storage to support and manage 32 servers, which is not needed for most environments resulting in inefficient usage of storage space. The CDPLEX.MAXSERVER global initialization parameter will control the maximum number of servers with the default being 4. If you have more than 4 servers in your environment, you must specify the number needed or you will receive an error. For more information, see [“Condition: Server Initialization Error \(SXTA101I\) with ABEND U1024” on page 79](#) and the information on the CDPLEX.MAXSERVER parameter in the *IBM Connect:Direct for z/OS Administration Guide*.

To calculate the amount of virtual storage above the line allocated for the queue holding the statistics records, the IBM Connect:Direct/Plex Manager multiplies the value for the STAT.QUEUE.ELEMENTS global initialization parameter by the maximum number of servers. If the resultant calculation is less than 5000, 5000 is used. If it is greater than 10000, 10000 is used. Each element takes 2048 bytes of storage.

Check the REGION parameter on the job card and specify REGION=0M.

Preparing TCP/IP Configuration (Optional)

Read this section if you are using TCP/IP support. IBM Connect:Direct supports IBM TCP/IP, Open Edition Sockets Interface, which you specify by using the OES value for the TCP initialization parameter. See Global Initialization Parameters. Review the NETMAP.CHECK parameter to determine if your site performs network map checking on TCP/IP nodes.

TCP/IP Support

The following procedures relate to the implementation of TCP/IP only:

- The HLQ.PROFILE.TCPIP dataset contains system operation and configuration information for the TCP/IP address space. The PORT statement reserves a port for a given user ID and identifies the protocol to be used on that port. It is not required that you reserve a port number for IBM Connect:Direct.
- To verify that a connection between nodes can be established, use the TCP/IP PING command. The PING command sends an echo request to a foreign host to determine if the computer is accessible.
- To verify that a connection exists to the remote IBM Connect:Direct, issue the following command:

```
TELNET ip-address,port-number
```

IBM Dynamic Virtual IP Address (Dynamic VIPA) Support

IBM SecureWay Communications Server provides for dynamic virtual IP addresses (dynamic VIPA). This feature enables you to define a TCP/IP stack so that a TCP/IP address is dynamic and exists only when the application that defines it is active. Following is an example:

```
:
VIPADynamic
    VIPARange      DEFINE      address_mask      network_prefix
ENDVIPADynamic
:
```

Note: Refer to IBM documentation for options and definitions.

To use Dynamic VIPA for IBM Connect:Direct, define a unique VIPA for each IBM Connect:Direct instance. An instance of IBM Connect:Direct is an “application” per the IBM SecureWay Communications documentation. When that instance is active, it defines the VIPA address, and when it terminates, it deactivates the VIPA address.

For IBM Connect:Direct Extended Recovery, define VIPA requirements the same way, but you must define the dynamic VIPA range in each TCP/IP stack. Each IBM Connect:Direct node must have a unique VIPA, meaning that in a IBM Connect:Direct/Plex environment, the manager and each server must have a unique VIPA to bind to. For HOT recovery, the standby server may have the same VIPA as the primary. The standby will not use the VIPA until it becomes the active server. In this way, it does not violate the VIPA rules as defined by IBM SecureWay Communications documentation.

Implementing TCP Stack Affinity

Connect:Direct for z/OS supports TCP/IP connectivity through multiple TCP/IP stacks and does not set stack affinity to any particular TCP/IP stack. You might want to run multiple TCP/IP stacks on the same system to provide network isolation for one or more of your applications. Establishing TCP/IP stack affinity binds all TCP/IP socket communications to that stack, which in turn allocates the proper host domain name resolution configuration datasets to IBM Connect:Direct. These datasets enable host name lookups to have the desired results.

If TCP/IP stack affinity is required, you must specify the stack using one of the following methods:

- Define the IP addresses and ports specified in the TCP.LISTEN initialization parameter to a particular TCP/IP stack using CINET
- Add the following step to the IBM Connect:Direct started task JCL before executing the DMINIT initialization module:

```
STEP 0      EXECUTE PGM=BPXTCAFF,PARTM=TCP_Stack_Name
```

Global TCP/IP SOURCEIP

The IBM Connect:Direct Server binds a local socket for out-bound processes and typically this ANYADDR or 0.0.0.0 unless the NETMAP parameter SOURCEIP is specified. A new global initialization parameter exists, TCP.SOURCEIP, to be a global default.

The intent of TCP.SOURCEIP is to define a global default IP address that the Connect:Direct for z/OS server can bind to for out-bound IP processes so that each individual NETMAP entry does not require the SOURCEIP parameter unless there is a requirement to override the default. Assigning ANYADDR and/or ANYADDR6 for this parameter, is accepted and used, but defeats the purpose of the parameter.

There are three ways to obtain the local IP (TALocal_IP/RATLADDR) which is the local IP address for the Bind:

- NETMAP SOURCEIP parameter
- TCP.SOURCEIP INITPARM

- Hostname

Note: Services do not exist to obtain a hostname for IPv6.

The bind port currently is obtained in one of these two ways:

- Bind to port zero(0) - default and most common
- As defined by the TCP.SRC.PORTS initialization parameter

The binding to port zero causes TCP/IP itself to assign the port from a defined range or pool of ports ranging from 1024 to 65365. This port is reference as an ephemeral port and as far as a IBM Connect:Direct servers it appears to be randomly assigned. The TCP.SRC.PORTS is used to define a smaller pool of ports to be used.

Exploit z/OS Encryption Readiness Technology (zERT)

IBM Connect:Direct with Secure Plus option can exploit z/OS Encryption Readiness Technology (zERT) capability provided by the z/OS V2R3 Communications Server. With zERT, the TCP/IP stack acts as a focal point in collecting and reporting the cryptographic security attributes of IPv4 and IPv6 application traffic that is protected using the TLS/SSL, SSH and IPsec cryptographic network security protocols. For more information, see z/OS® Encryption Readiness Technology for details and requirements.

Security Planning

IBM Connect:Direct supports signon security checking through its own Authorization Facility and through security exits interfacing with CA-ACF2 and CA-TOP SECRET by Computer Associates International, Inc., and Resource Access Control Facility (RACF) by IBM. Any of these packages can control access to IBM Connect:Direct functions. Read *Implementing Security in the IBM Connect:Direct for z/OS Administration Guide*.

If your system has z/OS UNIX System Services and RACF Program Control turned on, every JOBLIB/STEPLIB/LINKLIB DSN in the IBM Connect:Direct startup must be in the appropriate RACF Program Control list for HFS support to work correctly. If not, z/OS UNIX System Services considers the address space “dirty,” and setting thread-level security (which HFS support uses) fails with 0000008B xxxx02AF. IBM Connect:Direct initialization fails with the message SITA997I.

Note: The SP Admin tool is unable to open a Secure parameter file created prior to release 5.2.0. See [“DGASCONV – Secure Parameter File Conversion Utility”](#) on page 649 for more information.

Extended Submit Facility (ESF)

The Extended Submit Facility (ESF) enables Processes to be submitted even if the Connect:Direct DTF or the communications path between the API and DTF is not active.

The ESF is active because YES is the default parameter value for the ESF keyword on the Connect:Direct SIGNON command. An *AuthorizationRequired* error occurs if the logon ID where the API is running is not appropriately authorized when a Process is submitted through ESF. To prevent this error, do the following:

- If you submit Processes through ESF with CA-ACF2, ensure the logon ID is authorized through CA-ACF2 to update TCX and TCQ data sets.
- If you submit Processes through ESF with RACF, ensure the logon ID has control access authority for TCX and TCQ.

RACF Password Phrase (Passphrase)

IBM Connect:Direct for z/OS supports RACF Password Phrase(Passphrase) up to 64 characters in length. Any location within Connect:Direct where a password is accepted, a passphrase can be used in its place. For more information on RACF support of Password Phrase, see the Security Server RACF General User’s

Passphrases can contain characters that the Connect:Direct z/OS parser defines as "delimiter" characters:

Character	Description
	blank
<	less than
¬	logical not
,	comma
>	greater than
=	equal sign
/	forward slash
\	backward slash
'	single quote
"	double quote
(open parenthesis
)	close parenthesis

Passphrases can begin with a blank.

Passphrases can end with a blank.

Special Connect:Direct z/OS rules for Passphrase:

- Passphrases that contain a special character that is also a "delimiter" must be enclosed in double quotes or single quotes:

```
'This is<a>passphrase.'
```

or

```
"This is<a>passphrase."
```

- Passphrases that end with a blank must be enclosed with a combination of single quotes and double quotes:

```
' " Passphrase that contains blanks. "'
```

- Passphrases that contain one or more single quotes must be enclosed in double quotes:

```
"That's a passphrase, not his'ns."
```

Note: Passphrases that contain single quotes cannot be entered in the ISPF panels and should be avoided.

- Passphrases that contain one or more double quotes must be enclosed in single quotes:

```
'Passphrase for the "world".'
```

- Rules for entering a passphrase through the ISPF panels are the same as for entering the passphrase in a PROCESS statement. However, they are somewhat relaxed:
 - The ISPF code automatically encloses the passphrase in single quotes if it isn't entered enclosed in single or double quotes.

```
This is a <passphrase> and is "easy" to enter.
```

or

```
'This is a <passphrase> and is "easy" to enter.'
```

- Passphrase that end in a blank should be enclosed in double quotes (or the single/double quote - double/single quote pair).

```
"This is a passphrase that ends with a blank. "
```

or

```
'"This is a passphrase that ends with a blank. "'
```

Note: Passphrases that contain a single quote cannot be entered into the ISPF panels and should be avoided.

Note: If "delimiter" characters are avoided, entering the longer passphrase is the same as entering the password.

Summary

Passphrase	Enclosed within
Contains no Connect:Direct "delimiter"	none required
Contains Connect:Direct "delimiter" except single quote and/or double quote (see ending blank rule below)	' or "
Contains single quote *Cannot be entered with ISPF*	"
Contains double quote	'
Contains both single quote and double quote	*Not allowed*
Ends with blank, but has no single quote or double quote	"" ""
Ends with blank, and has a single quote or double quote	*Not allowed*

Planning for Parallel Sessions and Process Recovery

IBM Connect:Direct uses the parallel sessions capability to allow multiple Processes to execute simultaneously between any two IBM Connect:Direct nodes. Review *Building, Modifying, and Submitting Processes* in the *IBM Connect:Direct for z/OS User Guide* for more information on how to plan for parallel sessions.

IBM Connect:Direct provides facilities to recover from most errors that occur during Process execution. Recovery from the point of failure usually can be accomplished quickly. During the installation, you are asked to establish values for various parameters that affect Process recovery and checkpoint/restart. Review *Process Recovery and Checkpoint/Restart* in the *IBM Connect:Direct for z/OS User Guide*.

Planning the Network Map

The network map identifies the local IBM Connect:Direct node and the nodes where it can communicate. It consists of a local node entry and one or more adjacent node entries.

Each entry identifies the communications name and protocol associated with a IBM Connect:Direct node. A sample network map source can be found in member DGAXNTMP in the \$CD.SDGACNTL library. This

member should be tailored for and used as input to the network map load utility, DGADNTLD. This utility creates the VSAM form of the network map. It is invoked by the job DGAJLOAD in the \$CD.SDGAJCL library.

New Installations

1. Gather the information you will need to create your local and adjacent node entries.
2. To build a minimal netmap file with a local node and an initialization parameters file which uses all the default settings, follow the instructions in [Building a Test IBM Connect:Direct Configuration \(optional\)](#).
3. After completing the configuration process, update the network map to include other adjacent nodes in the network.

Existing Installations

To upgrade to a later version while retaining an existing netmap and Connect:Direct Secure Plus Parameters and Access files, see the *IBM Connect:Direct for z/OS Release Notes*.

Planning for Disaster Recovery Testing

When you back up a IBM Connect:Direct system for business continuity testing (also known as "disaster recovery testing or DR testing"), you create a snapshot of the DTF files at one point in time. If the DTF is running when the backup is taken, the captured files may not be in a synchronized known state.

To ensure that you perform a backup at a synchronized point in time to provide a "clean" starting point for the DTF, it is recommended that you put IBM Connect:Direct in a quiesced, or non-running state when you perform DR backups. Therefore, it is strongly recommended that you set the TCQ global initialization parameter to COLD when you start the DTF at the DR site.

If you do not know at what stage the backup was taken, you may have to deal with the following issues in a recovery situation:

- The TCQ and TCX datasets will probably be out of sync with each other, which will cause a failure when you start IBM Connect:Direct at a disaster recovery site. The DTF may terminate during initialization.
- If you need the contents of the TCQ to continue, run the DGADTQFX utility before you attempt to start the DTF (see *Managing the Transmission Control Queue* in the *IBM Connect:Direct for z/OS Administration Guide* for details). Because the system may not be in a known synchronized state, it is recommended that you set the TCQ initialization parameter to WARM and the QUIESCE initialization parameter to YES. Then you can delete Processes that have already executed before putting the DTF in the "Run" state.
- The NETMAP could also be corrupted if it was dynamically updated when the NETMAP was backed up. If you experience this with your system, unload the NETMAP, delete and redefine it, and then load it from the unloaded source.

Planning for Connect:Direct File Agent

Connect:Direct File Agent must be installed in a directory of the z/OS UNIX System Services component and configured to communicate with the IBM Connect:Direct server. Use a PC for terminal emulation when you are ready to create a configuration file using Connect:Direct File Agent. Connect:Direct File Agent uses mount points, which are HFS/zFS files requiring at least 20 cylinders of available space. See *IBM Connect:Direct for z/OS Release Notes* for software requirements for Connect:Direct File Agent.

Traces in Startup JCL

When a problem occurs while Connect:Direct is running, you can use a variety of traces to gather information to diagnose the problem and record events as they happen. Based on the trace specified, the Connect:Direct trace output is directed to various ddnames. For more information on traces, see *Isolating Problems*.

Connect:Direct provides the following DD statements in the DGAJCONN JCL member as the basic set of DDs to run your system including three automatic traces:

DDNAME	Function
STEPLIB	Connect:Direct SDGALINK
DMPUBLIB	Connect:Direct Process library
USRINFO	Standard display from User exits
NDMLOG	Automatic trace to list all initialization parameters read from the INITPARM dataset including obsolete parameters, which are indicated by SITA995I messages, and all modules, along with the last date on which they were modified, and related fix numbers.
ESTAE	Automatic trace to capture I/O errors, VTAM connection errors, ABEND control blocks, open and close errors, TCQ/TCX errors on adds and updates, and Statistics File write errors.
RPLERRCK	Automatic trace to capture VTAM and TCP/IP send and receive errors.
CDESTAE	Supplemental ESTAE Output
CEEOUT	Diagnostic messages from Language Environment programs (STDERR).

Configuring IBM Connect:Direct for z/OS

Define the IBM Connect:DirectVSAM Files

Edit and customize the following members in the \$CD.SDGACNTL dataset for your environment.

Procedure

1. Edit and customize the following members in the \$CD.SDGACNTL dataset for your environment:

- DGACVDEF – IDCAMS Define of the Connect:Direct VSAM files
- DGACVDEL – IDCAMS Delete of the Connect:Direct VSAM files
- DGACNDEF – IDCAMS Delete/Define of the Connect:Direct Network Map
- DGACAUTH – IDCAMS Repro of the AUTH File
- DGACMSGV – IDCAMS Repro of the MSG File
- DGACTYPE – IDCAMS Repro of the TYPE File

2. Edit, customize, and submit member DGAJVSAM in the \$CD.SDGAJCL dataset. This job defines all Connect:Direct VSAM files, and loads the MSG, TYPE and AUTH files.

Note: Connect:Direct does not support the extended format attribute in any of its VSAM ESDS control datasets. The following are Connect:Direct VSAM ESDS control datasets:

- MSG
- AUTH
- STATS
- TCQ
- TCX
- TYPE
- CKPT
- NETMAP
- PARMFILE

- ARCHDIR
- CONFIG
- EVENT
- USERPROF

Define VTAM Resources

Using the samples in dataset \$CD.SDGACNTL and VTAM definitions, define the VTAM resources appropriate for your environment.

Building a Test IBM Connect:Direct Configuration (optional)

IBM Connect:Direct for z/OS provides a series of menus that let you customize the installation. Both panel-level and field-level help are available by pressing the PF1 key. Panel-level help identifies required fields and provides general information requested on a panel. To see help for a specific field in a separate pop-up window, place the cursor on an individual field and press the PF1 key.

About this task

The installation panels are:

- The Connect:Direct for z/OS Installation Main Menu, which collects information about how you want to customize your installation, builds a basic test configuration, and provides the IBM Connect:Direct for z/OS Configuration Menu.
- The Connect:Direct for z/OS Configuration Menu displays information to build a default test configuration you can use as a preliminary version of the network map for testing. After the configuration process is complete, update the network map to include other adjacent nodes in the network. For more information on network maps, see *Maintaining the Network Map*. This step generates a JCL member (DGAJNETL) that you will run to build a minimal netmap file with a local node and an initialization parameters file which uses all the default settings.

Important: This procedure is intended for new users to assist them in building a test Connect:Direct for z/OS configuration—it is not for existing customers who already have netmaps populated with node information.

To display the installation panels:

Procedure

1. Request the TSO COMMAND Option (Option 6) from the ISPF Primary Option Menu and type the following:

```
=== > EXEC '$CD.SDGAISPC(DGA#CFG2)' '$CD'
```

2. When the Connect:Direct for z/OS Installation Main Menu is displayed, enter information in the **Permanent DASD Volume Serial No.** field. Press **Enter**.

```

IBM
Connect:Direct for z/OS      DATE-2015/02/13
----- Installation Main Menu ----- TIME-14:19
CMD ==>
C:D System High Level Qualifier ..... CSDQA1.HDGA510_____
Permanent DASD Device Type ..... SYSDA___
Permanent DASD Volume Serial No. ....
Temporary DASD Device Type ..... SYSDA___
Do you wish to configure a Test Connect:Direct... Y (Must be Y to continue)
Job Card Information
==> //CDINST JOB (CD-INSTALL),'CD INSTALL',CLASS=0,_____
==> // MSGCLASS=X,REGION=0M_____
==> //*_____
==> //*_____
Press ENTER to continue, PF1 for Help, PF3 to Terminate the Install

```

- When the Connect:Direct for z/OS Configuration Menu is displayed, enter information in the **Local Node Name** field to identify this node in the netmap (1–16 characters beginning with an alpha character).

Depending on your protocol, enter the following information:

- For SNA, specify the **VTAM SNA APPLID** and **VTAM SNA API APPLID** fields.
- For TCP, complete the **TCP IP Address or Hostname** field (supports IPV4, IPV6 as well as a Hostname).

```

IBM Connect:Direct
for z/OS      DATE-2011/03/04
----- Configuration Menu ----- TIME-13:55
CMD ==>
Connect:Direct configuration Information:
Local Node Name ..... N_____
SNA (Yes or No) ..... N_____
VTAM SNA APPLID .....
VTAM SNA API APPLID .....
TCP (Yes or No) ..... Y_____
TCP IP Address or Hostname ...
TCP Port Number .....
TCP API Port Number .....
Press ENTER to continue, PF1 for Help, PF3 to return

```

Press **Enter** to continue.

- When the Connect:Direct for z/OS JCL Generation Menu is displayed, press **Enter** to generate the JCL.

The following JCL members are generated in the resultant SDGAJCL dataset:

- DGAJNETL–JCL to build Netmap
- DGAJTST–Test Connect:Direct for z/OS JCL

- To build the test netmap, run the DGAJNETL job.

Once the job has run successfully, specify the NETDSN system file initialization parameter in the DGAJPARM member in \$CD.SDGACNTL. as the DSN created via the DGAJNETL job. This parameter specifies the file name of the Connect:Direct for z/OS VSAM network map file.

Building the Initialization Parameter File

New releases of IBM Connect:Direct for z/OS often implement new initialization parameters. In addition, initialization parameters frequently become obsolete. IBM Connect:Direct detects parameters, which should be retired, and issues messages to NDMLOG. If you receive a return code of 4 when you stop IBM Connect:Direct, review the NDMLOG for a list of obsolete parameters. You must remove obsolete parameters to eliminate SITA995I messages.

About this task

IBM Connect:Direct processes initialization parameters during startup to specify alternate values for various parameters. The IBM Connect:Direct initialization module processes the file that contains these parameters.

Note: In addition to modifying initialization parameter files directly, use IBM Control Center to manage these parameters for a IBM Connect:Direct stand-alone server or the global and local initialization parameters for a IBM Connect:Direct/Plex.

You can create backup copies of the global and local initialization parameter files to use in an emergency situation if IBM Connect:Direct cannot successfully initialize after initparm updates have been applied.

You must define all initialization parameters that specify the IBM Connect:Direct VSAM file names except the optional \$CD.STAT.ARCH.DIR parameter. Failure to take this step results in unpredictable behavior.

To specify your initialization parameters:

Procedure

1. Modify the parameters as required.

Find sample initialization parameters in the DGAINT01 member of the \$CD.SDGAPARM. Observe the following required or suggested values:

- For initial installation and testing, specify SECURITY.EXIT=OFF in member DGAINT01. When IBM Connect:Direct security exits are installed, change the SECURITY.EXIT installation keyword. This change prevents any problems with user definitions until the basic installation is verified.
- If you are using TCP/IP connectivity, you must specify TCP=OES. Also specify valid values for the TCP.LISTEN parameter.
- The UPPER.CASE initialization parameter controls what case initialization console messages are displayed in. The default is UPPER.CASE=NO, which means that all console messages are displayed in upper and lower case. If you want to display all console messages in upper case, you must specify UPPER.CASE=YES (this parameter can be modified as an override parameter in the DGADINIT execute statement in the IBM Connect:Direct startup job stream as well).
- If you generated the Test Configuration and successfully executed the DGAJNETL, then specify the NETDSN system file initialization parameter as the DSN created via the DGAJNETL job.

Note: You can initialize IBM Connect:Direct if SNA support is not available. See [Configuring IBM Connect:Direct without SNA Support](#).

2. Specify VSAM file names.

Specify all initialization parameters that indicate the IBM Connect:Direct VSAM file names. Locate following file names in the member DGACVDEF of the \$CD.CNTL library:

- \$CD.AUTH
- \$CD.CKPT
- \$CD.MSG
- \$CD.NETMAP
- \$CD.STATS
- \$CD.TYPE

3. Override the parameters during startup.

Override parameters during startup by specifying the parameter and its value in the PARM keyword of the EXEC statement in the startup JCL. See [Start IBM Connect:Direct](#) for sample JCL that shows an override for the UPPER.CASE=NO initialization parameter.

Note: If you are upgrading and have user exits in your IBM Connect:Direct installation, you must reassemble and link-edit those exits. For more information on user exits, see [Using IBM Connect:Direct Exits](#) in the IBM Connect:Direct for z/OS Administration Guide.

Installing the ISPF IUI

The IBM Connect:Direct IUI is an ISPF dialog. ISPF dialogs have three phases: allocation, invocation, and unallocation. Each of these phases can be done after TSO logon is complete, either outside or inside ISPF. Allocation can also be done through the TSO logon procedure. LIBDEF type ISPLLIB is now supported, so you can invoke the entire dialog from inside ISPF. In previous IBM Connect:Direct for z/OS releases, the IUI load library had to be allocated outside of ISPF.

Allocation Methods for IUI data sets

The following tables list the allocation methods and scopes of all IUI data sets. The changes for IBM Connect:Direct for z/OS 6.0 are marked in bold.

Note:

1. You cannot change STEPLIB after TSO LOGON
2. TSOLIB can only be invoked at TSO READY. The DDNAME can be anything
3. If you use LIBDEF ISPLLIB in IBM Connect:Direct for z/OS 5.2, DGADISTR abends S806.
4. Use DDNAME SYSPROC instead of SYSEXEC because SDGAISPC contains CLISTS
5. Use ALTLIB ACT APPLICATION(CLIST) because SDGAISPC contains CLISTS.

IBM Connect:Direct for z/OS 5.2	Allocated via TSO Logon Procedure		Allocated after TSO LOGON, Outside ISPF		Allocated after TSO LOGON, Inside ISPF	
	<i>DDNAME</i>	<i>ISPF Logical Screen Scope</i>	<i>Command</i>	<i>ISPF Logical Screen Scope</i>	<i>Command</i>	<i>ISPF Logical Screen Scope</i>
VSAM MSG	DMMSGFIL	All	ALLOCATE	All	ALLOCATE	All
Process	DMPUBLIB	All	ALLOCATE	All	ALLOCATE	All
SDGALINK	STEPLIB (1)	All				
			TSOLIB (2)	All		
	ISPLLIB	All	ALLOCATE	All	(3)	
SDGAMENU	ISPMLIB	All	ALLOCATE	All	LIBDEF	One
SDGAPENU	ISPPLIB	All	ALLOCATE	All	LIBDEF	One
SDGASENU	ISPSLIB	All	ALLOCATE	All	LIBDEF	One

IBM Connect:Direct for z/OS 6.0	Allocated via TSO LOGON Procedure		Allocated after TSO LOGON, Outside ISPF		Allocated after TSO LOGON, Inside ISPF	
	<i>DDNAME</i>	<i>ISPF Logical Screen Scope</i>	<i>Command</i>	<i>ISPF Logical Screen Scope</i>	<i>Command</i>	<i>ISPF Logical Screen Scope</i>
VSAM MSG	DMMSGFIL	All	ALLOCATE	All	ALLOCATE	All
Process	DMPUBLIB	All	ALLOCATE	All	ALLOCATE	All
SDGALINK	STEPLIB (1)	All				
			TSOLIB (2)	All		
	ISPLLIB	All	ALLOCATE	All	LIBDEF	One

IBM Connect:Direct for z/OS 6.0	Allocated via TSO LOGON Procedure		Allocated after TSO LOGON, Outside ISPF		Allocated after TSO LOGON, Inside ISPF	
SDGAMENU	ISPMLIB	All	ALLOCATE	All	LIBDEF	One
SDGAPENU	ISPPLIB	All	ALLOCATE	All	LIBDEF	One
SDGASENU	ISPSLIB	All	ALLOCATE	All	LIBDEF	One

TSO Logon Procedure Allocation Example

Modify the LOGON PROC used to log in to TSO by integrating IBM Connect:Direct libraries. The bold lines in the following example are for IBM Connect:Direct.

```
//TSO PROC
//*
//IEFPROC EXEC PGM=IKJEFT01,DYNAMBR=25,
//  PARM="PROFILE MODE WTPMSG MSGID"
//*
//DMSGFIL DD DSN=$CD.MSG,DISP=SHR
//DMPUBLIB DD DSN=$CD.SDGAPROC,DISP=SHR
//SYSPROC DD DSN=$CD.SDGAISPC,DISP=SHR
//          DD DSN=USR.ISPCLIB,DISP=SHR
//          DD DSN=ISP.SISPCLIB,DISP=SHR
//ISPLLIB DD DSN=$CD.SDGALINK,DISP=SHR
//          DD DSN=USR.ISPLLIB,DISP=SHR
//ISPMLIB DD DSN=$CD.SDGAMENU,DISP=SHR
//          DD DSN=USR.ISPMLIB,DISP=SHR
//          DD DSN=ISP.SISPMENU,DISP=SHR
//ISPPLIB DD DSN=$CD.SDGAPENU,DISP=SHR
//          DD DSN=USR.ISPPLIB,DISP=SHR
//          DD DSN=ISP.SISPPENU,DISP=SHR
//ISPSLIB DD DSN=$CD.SDGASENU,DISP=SHR
//          DD DSN=USR.ISPSLIB,DISP=SHR
//          DD DSN=ISP.SISPSLIB,DISP=SHR
//          DD DSN=ISP.SISPSENU,DISP=SHR
//ISPTABL DD DSN=USR.ISPTABL,DISP=SHR
//ISPTLIB DD DSN=USR.ISPTABL,DISP=SHR
//          DD DSN=ISP.SISPTENU,DISP=SHR
//ISPPROF DD DSN=USR.ISPPROF,DISP=SHR
//SYSHELP DD DSN=SYS1.HELP,DISP=SHR
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSPRINT DD TERM=TS,SYSOUT=A
//SYSTEM DD TERM=TS,SYSOUT=A
//SYSTSPRT DD TERM=TS,SYSOUT=A
//SYSIN DD TERM=TS,SYSOUT=A
//SYSTSIN DD DDNAME=IEFRDER
//IEFRDER DD TERM=TS,SYSOUT=A
```

Example of TSO CLIST Allocation Outside ISPF

Create a CLIST to allocate the ISPF data sets outside of ISPF and integrate the IBM Connect:Direct libraries. The bold lines in the following example are for IBM Connect:Direct.

```
ALLOC F(DMSGFIL) DA('$CD.MSG') SHR REU
ALLOC F(DMPUBLIB) DA('$CD.SDGAPROC') SHR REU
ALLOC F(SYSPROC) DA('$CD.SDGAISPC' -
'USR.ISPCLIB' -
'ISP.SISPCLIB') SHR REU
ALLOC F(ISPLLIB) DA('$CD.SDGALINK' -
'USR.ISPLLIB') SHR REU
ALLOC F(ISPMLIB) DA('$CD.SDGAMENU' -
'USR.ISPMLIB' -
'ISP.SISPMENU') SHR REU
ALLOC F(ISPPLIB) DA('$CD.SDGAPENU' -
'USR.ISPPLIB' -
'ISP.SISPPENU') SHR REU
ALLOC F(ISPSLIB) DA('$CD.SDGASENU' -
'USR.ISPSLIB' -
'ISP.SISPSLIB' -
'ISP.SISPSENU') SHR REU
ALLOC F(ISPTABL) DA('USR.ISPTABL') SHR REU
```

```

ALLOC F(ISPTLIB) DA('USR.ISPTABL' -
                   'ISP.SISPTENU') SHR REU
ALLOC F(ISPPROF) DA('USR.ISPPROF')  SHR REU
...
PDF

```

TSOLIB can be used in a CLIST outside ISPF instead of ALLOCATE FI(ISPLLIB):

```
TSOLIB ACTIVATE DATASET('$CD.SDGALINK')
```

Example of TSO REXX Allocation Outside ISPF

You can use REXX instead of CLIST for allocation. TSOLIB cannot be used inside a REXX EXEC.

```

/* REXX */
ADDRESS TSO
"ALLOC F(DMMSGFIL) DA('$CD.MSG') SHR REU"
"ALLOC F(DMPUBLIB) DA('$CD.SDGAPROC') SHR REU"
"ALLOC F(SYSPROC) DA('$CD.SDGAISPC' ,
                   "'USR.ISPCLIB' ,
                   "'ISP.SISPCLIB') SHR REU"
"ALLOC F(ISPLLIB) DA('$CD.SDGALINK' ,
                   "'USR.ISPLLIB') SHR REU"
"ALLOC F(ISPMLIB) DA('$CD.SDGAMENU' ,
                   "'USR.ISPMLIB' ,
                   "'ISP.SISPMENU') SHR REU"
"ALLOC F(ISPPLIB) DA('$CD.SDGAPENU' ,
                   "'USR.ISPPLIB' ,
                   "'ISP.SISPPENU') SHR REU"
"ALLOC F(ISPSLIB) DA('$CD.SDGASENU' ,
                   "'USR.ISPSLIB' ,
                   "'ISP.SISPSLIB' ,
                   "'ISP.SISPSENU') SHR REU"
"ALLOC F(ISPTABL) DA('USR.ISPTABL') SHR REU"
"ALLOC F(ISPTLIB) DA('USR.ISPTABL' ,
                   "'ISP.SISPTENU') SHR REU"
"ALLOC F(ISPPROF) DA('USR.ISPPROF') SHR REU"
...
PDF

```

Example of TSO CLIST Allocation Inside ISPF

You can allocate DMMSGFIL and DMPUBLIB inside ISPF by using the TSO ALLOCATE command, and free it by using the TSO FREE command. The ISPF application component data sets must be allocated by using LIBDEF, since ISP*LIB DDs cannot be altered from inside ISPF. Also, although SYSPROC can be reallocated inside ISPF, it is easier to use the TSO ALTLIB command to direct the IUI application to the Connect:Direct CLIST/REXX PDS.

Create a CLIST to allocate the IUI dialog data sets inside ISPF. TSOLIB cannot be used inside ISPF.

```

PROC 0
ALLOC F(DMMSGFIL) DA('$CD.MSG') SHR REU
ALLOC F(DMPUBLIB) DA('$CD.SDGAPROC') SHR REU
ALTLIB ACTIVATE DATASET('$CD.SDGAISPC') APPLICATION(CLIST)
ISPEXEC LIBDEF ISPLLIB DATASET ID('$CD.SDGALINK') STACK
ISPEXEC LIBDEF ISPMLIB DATASET ID('$CD.SDGAMENU') STACK
ISPEXEC LIBDEF ISPPLIB DATASET ID('$CD.SDGAPENU') STACK
ISPEXEC LIBDEF ISPSLIB DATASET ID('$CD.SDGASENU') STACK
EXIT

```

Example of TSO REXX Allocation Inside ISPF

Create a REXX to allocate the IUI dialog data sets inside of ISPF.

```

/* REXX */
ADDRESS TSO
"ALLOC F(DMMSGFIL) DA('$CD.MSG') SHR REU"
"ALLOC F(DMPUBLIB) DA('$CD.SDGAPROC') SHR REU"
"ALTLIB ACTIVATE DATASET('$CD.SDGAISPC') APPLICATION(CLIST)"
ADDRESS ISPEXEC
"LIBDEF ISPLLIB DATASET ID('$CD.SDGALINK') STACK"
"LIBDEF ISPMLIB DATASET ID('$CD.SDGAMENU') STACK"
"LIBDEF ISPPLIB DATASET ID('$CD.SDGAPENU') STACK"

```

```
"LIBDEF ISPSLIB DATASET ID('$CD.SDGASENU') STACK"
EXIT 0
```

Invoke the IUI Main Dialog (DGADISTR)

Typically, IUI invocation is done inside ISPF, since doing it outside ISPF via ISPSTART makes it difficult to use the other ISPF options. Connect:Direct for z/OS IUI invocation must be done by using ISPF SELECT with PGM, not SELECT with CMD. The SELECT can be in a panel, a CLIST, or REXX, or in an ISPF command table entry. The only prerequisite is that all required DDNAMEs are allocated.

A new ISPF APPLID for Connect:Direct for z/OS has to be specified by using NEWAPPL(applid). If LIBDEF was used to allocate the ISPF application component data sets, and NEWAPPL was specified in the SELECT, then you must also specify PASSLIB or the LIBDEFs are not searched.

See the table for IUI invocation examples.

Panel	&ZSEL = TRANS(TRUNC (&ZCMD,!) ... CD,'PGM(DGADISTR) PARM(cd-netmapdsn)..)
CLIST	ISPEXEC SELECT PGM(DGADISTR) PARM(cd-netmap-dsn)
REXX	ADDRESS ISPEXEC "SELECT PGM(DGADISTR) PARM(cd-netmap-dsn)"
CMD Table	ZCTACT = SELECT PGM(DGADISTR) PARM(cd-netmap-dsn)

Invoke the IUI MSG Dialog (DGADGDSP)

The IUI main dialog includes the message (MSG) dialog. If you want to look up a Connect:Direct for z/OS message without having to go through the IUI login process, you can use the message dialog. You must allocate the same ISPF components as for the IUI dialog, except that you do not need to allocate them there DMMSGFIL or DMPUBLIB DDs. The message dialog is invoked through SELECT PGM in the same ways as the IUI dialog, but it uses a different program name and parameter. The parameter is the same VSAM message file DSN you would allocate to DMMSGFIL when you invoke the IUI Main Dialog.

Panel	&ZSEL = TRANS(TRUNC (&ZCMD,!) ... CD,'PGM(DGADGDSP) PARM(cd-msgfile-dsn) ...)
CLIST	ISPEXEC SELECT PGM(DGADGDSP) PARM(cd-msgfile-dsn)
REX	ADDRESS ISPEXEC "SELECT PGM(DGADGDSP) PARM(cd-msgfile-dsn)"
CMD Table	ZCTACT = SELECT PGM(DGADGDSP) PARM(cd-msgfile-dsn)

A new ISPF APPLID for Connect:Direct for z/OS can also be specified by using NEWAPPL(applid). If LIBDEF was used to allocate the ISPF application component data sets, and NEWAPPL was specified in the SELECT, you must also specify PASSLIB, or the LIBDEFs are not searched.

See the following example of the invocation of ISPF primary option panel with IUI and MSG dialogs:

```
%----- ISPF PRIMARY OPTION MENU -----
%OPTION ==>_ZCMD+
%%
+USERID - &ZUSER
```



```

% 0+SPF PARMS - Specify ... +TIME - &ZTIME
% 1+BROWSE - Display ... +DATE - &ZDATE
% 2+EDIT - Create ... +JULIAN - &ZJDATE
% 3+UTILITIES - Perform ... +TERMINAL - &ZTERM
% 4+FOREGROUND - Compile,... +PF KEYS - &ZKEYS
% 5+BACKGROUND - Compile, Assemble, or Link Edit
% 6+COMMAND - Enter TSO command or CLIST
% 7+SUPPORT - Test dialog or convert menu/message fmts
% M+C:D-MSGs - Display IBM Connect:Direct messages
% N+C:D - IBM Connect:Direct IUI
% X+EXIT - Terminate ISPF using List/Log defaults
%%
)INIT
.HELP=TTUTOR
&ZHTOP=TTUTOR
&ZHINDEX=TINDEX
)PROC
&ZSEL=TRANS( TRUNC (&OPT, '.'))
0, 'PANEL(ISPOPT)'
1, 'PGM(ISPBRO)'
2, 'PGM(ISPEDIT)'
3, 'PANEL(ISPUTIL)'
4, 'PANEL(ISPFORA)'
5, 'PANEL(ISPJOB)'
6, 'PGM(ISPTSO)'
7, 'PANEL(ISPQTAC) NEWPOOL'
M, 'PGM(DGADGDSP) PARM($CD.MSG)NEWAPPL(CD) PASSLIB'
N, 'PGM(DGADISTR) PARM($CD.NETMAP) NEWAPPL(CD)PASSLIB'
X, 'EXIT'
)END

```

Unallocation of IUI Data Sets

Unallocation usually occurs at the same level as allocation. For example, if the allocations are done outside ISPF, they are freed outside ISPF. It is not necessary to free data sets allocated via the ALLOCATE command, but it is customary to remove any LIBDEF and ALTLIB allocations when the application ends. Failure to do so can have unpredictable results.

Example of TSO CLIST Unallocation Inside ISPF

This example unallocates the allocations made in [“Example of TSO CLIST Allocation Inside ISPF”](#) on page 45.

```

PROC 0
FREE F(DMMSGFIL DMPUBLIB)
ALTLIB DEACTIVATE APPLICATION(CLIST)
ISPEXEC LIBDEF ISPLLIB
ISPEXEC LIBDEF ISPLMLIB
ISPEXEC LIBDEF ISPLPLIB
ISPEXEC LIBDEF ISPLSLIB
EXIT

```

Example of TSO REXX Unallocation Inside ISPF

This example unallocates the allocations made in [“Example of TSO REXX Allocation Inside ISPF”](#) on page 45.

```

/* REXX */
ADDRESS TSO
"FREE F(DMMSGFIL DMPUBLIB)"
"ALTLIB DEACTIVATE APPLICATION(CLIST)"
ADDRESS ISPEXEC
"LIBDEF ISPLLIB"
"LIBDEF ISPLMLIB"
"LIBDEF ISPLPLIB"
"LIBDEF ISPLSLIB"
EXIT 0

```

Using a Panel Option and Screen Prompts with Permanent Allocations to Implement the IUI

About this task

Follow these steps to implement the IUI by using a panel option and screen prompts with permanent allocations.

Procedure

1. Change the TSO LOGON PROC by following the example in [“TSO Logon Procedure Allocation Example” on page 44.](#)
2. Alternatively, create a REXX or CLIST to be invoked outside or inside ISPF that does the allocations. For more information, see the TSO CLIST and TSO REXX allocation examples.
3. Change the ISPF Primary Option menu by following the example in [“Invoke the IUI Main Dialog \(DGADISTR\)” on page 23](#) or in [“Invoke the IUI MSG Dialog \(DGADGDSP\)” on page 24.](#)
[“Invoke the IUI Main Dialog \(DGADISTR\)” on page 46](#) or
[“Invoke the IUI MSG Dialog \(DGADGDSP\)” on page 46.](#)
4. Get into ISPF, navigate to the modified panel, and enter one of the chosen command strings.

Invoking a CLIST Inside ISPF

About this task

Follow these steps to invoke a CLIST inside ISPF.

Procedure

1. Start by creating a CLIST to be invoked inside ISPF that does the allocations. See [“Example of TSO CLIST Allocation Inside ISPF” on page 45](#) for an example of this allocation. Remove the EXIT statement. Customize with your CD DSNs (and applid, if NEWAPPL was added).
2. Add the call to DGADISTR after the allocations. For an example of the CLIST syntax, see [“Invoke the IUI Main Dialog \(DGADISTR\)” on page 46.](#) Customize with your NETMAP DSN.
3. Add the unallocation statements. For an example of unallocation, see [“Unallocation of IUI Data Sets” on page 47.](#)
4. Invoke the CLIST from ISPF 6.

Using a REXX EXEC Invoked Inside ISPF

About this task

Follow these steps to invoke a REXX EXEC inside ISPF.

Procedure

1. Start by creating a REXX EXEC to be invoked inside ISPF that does the allocations. Follow the example of allocating TSO REXX inside ISPF, included in [“Example of TSO REXX Allocation Inside ISPF” on page 45.](#) Remove the EXIT statement. Customize with your Connect:Direct DSNs and applid, if NEWAPPL was added.
2. Add the call to DGADISTR after the allocations. For an example of the REXX syntax, see [“Invoke the IUI Main Dialog \(DGADISTR\)” on page 46.](#) Customize with your NETMAP DSN.
3. Add the unallocation statements. For an example of TSO REXX unallocation, see, [“Unallocation of IUI Data Sets” on page 47.](#)

4. From ISPF 6, invoke the REXX EXEC.

Multiple Concurrent IUI Sessions

It is possible to have one IUI session per ISPF logical screen. To start the first IUI session, you might use a script similar to the one below.

```
/* REXX */
ARG NETMAP APPL
IF NETMAP = '' THEN NETMAP = "CDA.NETMAP"
IF APPL = '' THEN APPL = "CDA"
ADDRESS TSO
"ALLOC F(DMMSGFIL) DA('$CD.MSG') SHR REUSE"
"ALLOC F(DMPUBLIB) DA('$CD.SDGAPROC') SHR REUSE"
"ALTLIB ACTIVATE DATASET('$CD.SDGAISPC') APPLICATION(CLIST)"
ADDRESS ISPEXEC
"LIBDEF ISPLLIB DATASET ID('$CD.SDGALINK') STACK"
"LIBDEF ISPLMLIB DATASET ID('$CD.SDGAMENU') STACK"
"LIBDEF ISPLPLIB DATASET ID('$CD.SDGAPENU') STACK"
"LIBDEF ISPLSLIB DATASET ID('$CD.SDGAENU') STACK"
"SELECT PGM(DGADISTR) PARM("netmap") NEWAPPL("appl") PASSLIB"
ADDRESS TSO
"FREE F(DMMSGFIL DMPUBLIB)"
"ALTLIB DEACTIVATE APPLICATION(CLIST)"
ADDRESS ISPEXEC
"LIBDEF ISPLLIB"
"LIBDEF ISPLMLIB"
"LIBDEF ISPLPLIB"
"LIBDEF ISPLSLIB"
EXIT 0
```

Invoked with no arguments and no other IUI sessions active in the TSO LOGON, the IUI is started with the default netmap and applid, and when exited, cleanup is complete and leaves the TSO session in the same state as before the REXX EXEC was invoked. The LIBDEF and ALTLIB commands are limited in scope to the ISPF logical screen they are invoked on. But the scope of the TSO ALLOCATE command is the entire TSO session. So if before exiting, the ISPF SPLIT command is invoked, and the same REXX is invoked (say, specifying a different netmap DSN) on the second ISPF logical screen, the DMMSGFIL and DMPUBLIB allocations will be re-done. Since REUSE was specified, the ALLOCATE commands will complete and both IUI sessions will function properly. But when either IUI session is exited, it will free DMMSGFIL and DMPUBLIB, effectively pulling the rug out from underneath the other IUI session. The simplest way to avoid this problem is to remove the FREE command and manually unallocate them as required.

Note:

ISPF supports up to 32 logical screens. Your installation can reduce that number. Each IUI session requires a certain amount of storage (below and above the line). You may need to increase the TSO user's region to run multiple IUI sessions. Any ISPF logical screen can be used for any IUI session. You are allowed only one IUI session per ISPF logical screen. If you invoke more than one IUI session on one ISPF logical screen, the results are unpredictable. IUI sessions started with the same ISPF APPLID share the same ISPF profile pool. Each IUI session can utilize multi-session signon.

Different Release Levels used by Multiple Concurrent IUI Sessions

Multiple concurrent IUI sessions cannot use different IBM Connect:Direct release levels.

The first IUI session will load reentrant load modules into storage, and because they are reentrant, subsequent concurrent IUI sessions started in other logical screens will use them instead of loading a new copy from the ISPLLIB DD, the LIBDEF ISPLLIB, or the LINKLIST. Thus only one IBM Connect:Direct release can be used at a time, regardless of the number of concurrent IUI sessions.

IUI Load Module Search

In Connect:Direct for z/OS for z/OS 6.0 and later, the IUI calls load modules in two ways. The IUI uses the ISPF SELECT PGM service call when it calls modules that are limited to IUI usage. The IUI uses the MVS macros LINK or LOAD with the DCB parameter when it calls modules common to the entire IBM Connect:Direct .

When you do not employ LIBDEF ISPLLIB with the IUI session, the IUI passes a zero in the DCB parameter, which is the same as not specifying the DCB parameter at all. When you employ LIBDEF ISPLLIB, the IUI passes the DCB of the LIBDEF ISPLLIB concatenation to LINK and LOAD. A nonzero DCB changes the MVS load module search order by skipping the requesting task's task library and all the unique task libraries of its preceding tasks (for example, ISPLLIB and TSOLIB concatenations). It also skips the STEPLIB concatenation. If you employ LIBDEF ISPLLIB, all IUI load modules must be in either the LIBDEF ISPLLIB concatenation or the LINKLIST. For more information about the MVS load module search order, see “The search for the load module” *z/OS MVS Programming: Assembler Services Guide*. For more information about the SELECT service load module search order, see “Application data element search order” in *ISPF Services Guide, LIBDEF service*.

Checking for IUI ISPLLIB LIBDEF Support

If you create a REXX or CLIST to invoke the IUI using LIBDEF ISPLLIB, you should check that load module DGADR14 exists in the LIBDEF ISPLLIB library before doing the SELECT PGM. If it does not exist, LIBDEF ISPLLIB is not supported. An example of how to do this (using REXX) follows:

```
...
if sysdsn('$CD.SDGALINK(DGADR14)') \= "OK" then LIBDEF ISPLLIB is not supported by this CD
release.
The CD load module library must be allocated to DD ISPLLIB using ALLOCATE outside ISPF."
exit 4
end
...
"SELECT PGM(DGADISTR) ..."
...
```

IUI Stage 1 Exits

In Connect:Direct for z/OS 6.0 and later, the Stage 1 exits with the IUI can be in a separate library and/or concatenation from the rest of the IUI load modules, subject to the restrictions above. Whether they are together or separate from the rest of the IBM Connect:Direct load modules, the Stage 1 exits must be in an APF-authorized library. This library can be concatenated with APF unauthorized libraries. The Stage 1 exits are no longer required to be linked with AC(1), or be NONRENT or NONREUS. They should be linked with AMODE 31 since the parameter list passed to them is above the line.

Start IBM Connect:Direct

The \$CD.SDGAJCL(DGAJCONN) member contains the sample startup job stream to run the IBM Connect:Direct DTF as a batch job. It can run as a batch job or as a started task.

About this task

Important: If you generated the Test Configuration JOB, DGAJNETL, use that JOB instead of the DGAJCONN JOB or modify the DGAJCONN JOB to perform the same as DGAJNETL.

If you use Program Access to datasets (PADS) functionality in your security system, include all datasets in the IBM Connect:Direct JCL STEPLIB DD concatenation in your Program Control List (PCL). See *Implementing Security in the IBM Connect:Direct for z/OS Administration Guide* for more information.

To start IBM Connect:Direct:

Procedure

1. Execute IBM Connect:Direct from an authorized library. The installation makes the DTF load module (DGADINIT) APF-authorized with AC(1).
2. Submit the startup job stream. While Connect:Direct for z/OS is initializing, a series of messages display the sequence of events during initialization. If an initialization error occurs, note the last message issued. The problem most likely occurred during the step indicated by the last message or on the step indicated by the next message that should have been issued. For more information on specific error messages, see [“Initialization Errors” on page 77](#)

Following is an example of messages you may see. The specific startup messages generated when you start Connect:Direct for z/OS depend on the following factors:

- What communication protocols are used in your system
- Whether Connect:Direct Secure Plus is part of your system and the version
- Whether any initialization parameters are being overridden in the startup JCL
- Whether any certificates have expired or will soon expire

Note: The SITA460I and SITA462I messages related to the Strong Password Encryption (SPE) feature are displayed even if no encryption is possible. To determine if this feature is in effect, go to the Secure + Create/Update Panel - SPE Parameters panel where SPE is enabled.

```
SITA001I IBM Connect:Direct for z/OS initialization has begun.
SITA002I Connect:Direct parameter file allocated and open.
SITA618I CD zIIP support is enabled.
SITA617I zIIP processor online.
SITA658I CD zFBA support is enabled.
SITA120I Keyword INITPARM Value Overridden: TCQ
SITA120I Keyword INITPARM Value Overridden: STAT.INIT
SITA120I Keyword INITPARM Value Overridden: DEBUG
SITA740I Connect:Direct DGAOPLS Found.
SITA022I Loading Connect:Direct modules.
SITA601I The TCP server modules are loaded.
SITA067I MESSAGE file is open.
SITA628I SNMP Trap Agent Initialization Complete.
SITA023I Initializing Connect:Direct storage.
SITA024I Building the TCA chain.
SITA026I Creating the system tasks (master and timer).
SITA025I Building the ECB address list.
SITA027I Building the DDN table.
SITA069I NETWORK MAP file is open
SITA028I SECURE+ Initialization complete
SITA029I Statistics facility being initiated.
SITA996I STATS WARM Start being performed.
SITA998I Acquiring storage for STAT.QUEUE.ELEMENTS.
SSTL026I Statistics File Pair 01 is now active.
SSTL019I Statistics facility successfully initialized.
SITA068I AUTHORIZATION file is open.
SITA460I Strong Password Encryption Initiated; $CDZ.AUTH
SITA462I Strong Password Encryption Completed; $CDZ.AUTH
SITA134I TYPE DEFAULTS file is open.
SITA074I CHECKPOINT file is open.
SITA030I PCQ/TCQ being built.
SITA400I Enqueuing TCX from NETMAP for ESF processing.
SITA996I TCQ WARM Start being performed.
SITA460I Strong Password Encryption Initiated; $CDZ.TCQ
SITA462I Strong Password Encryption Completed; $CDZ.CDXTST.TCQ
SITA401I Dequeuing TCX from NETMAP for ESF processing.
SITA034I VTAM services being initiated APPLID = M1DEVMW0.
SVTJ019I SNA Support is Now Available.
SITA439I Global INITPARM member successfully backed up to $BACKUP
SITA977I Product Registration was Successful
SITA899I Connect:Direct Node : MY_LOCAL_CDZ
SITA370I ZLIB version : 1.2.7
SITA374I zEDC Express Accelerator status is AVAILABLE
SITA036I Connect:Direct 6.00.00 5655-X11 Initialization Complete.
SITA195I Secure+ TLS FIPS Mode initialization complete.
SITA146I TCQ automatic update has been initiated.
SITA147I TCQ automatic update has deleted 0 queue entries.
STC0109I TCP/IP Server Subtask Attached
STC0110I TCP/IP Interface Initializing
STC0140I TCP/IP IPV6 Support Initializing
STC0102I TCP/IP Issuing BIND Call for address 4199;0.0.0.0
STC0103I TCP/IP Issuing LISTEN Call: 4199;0.0.0.0
STCA102I TCP/IP API Issuing BIND Call for address 4198;0.0.0.0
STCA103I TCP/IP API Issuing LISTEN Call: 4198;0.0.0.0
STC0111I TCP/IP Interface Initializing Complete
STCP104I TCP Support is Available.
SNOI000I Operator Interface - initialized, MCS.CLIST open
CSPA601E ERROR Cert: CDZCERTA expired on: 01/01/2010-05:59:59
CSPA601E ERROR Cert: CDCertB expired on: 12/09/2010-20:41:43
```

The IEC161I, 062-086, or 056-084 messages can also display at OPEN of the VSAM files. These messages indicate that the files were not closed properly the last time IBM Connect:Direct was brought down. These are normal VSAM verification messages.

3. Log on to TSO again.

After the TSO IUI is installed, log on to TSO again so that your new logon procedure or signon CLIST is in effect. You can then sign on to IBM Connect:Direct through the IUI, Operator Interface, or Batch Interface.

Remember: You can initialize IBM Connect:Direct if SNA support is not available. See [Configuring IBM Connect:Direct without SNA Support](#).

Signing On to IBM Connect:Direct

You can sign on to IBM Connect:Direct through the IUI, the Operator, or the Batch interface. For sign-on instructions for the Batch interface, refer to the *IBM Connect:Direct for z/OS User Guide*.

Signing On through the IUI Interface

Sign on to IBM Connect:Direct uses one of the following methods:

Procedure

- If the IBM Connect:Direct Authorization Facility is in effect, use the user ID SUPERUSR and the password supplied for the superuser in the network map local node definition.
 - If you are running with a IBM Connect:Direct security exit, use a user ID and password that meet your security requirements.
 - For an SNA environment, you must specify SNA=YES and define enough APPLIDS to handle both the IUI and Operator Interface sessions in the Netmap ADJACENT.NODE for LOCAL.NODE.
- For additional information on the IUI and procedures to automate the Signon process, refer to the IBM Connect:Direct for z/OS User Guide.

Signing On through the Operator Interface

When you use the Operator Interface, the operator is automatically signed on to IBM Connect:Direct when the first command or CLIST is issued. You must specify the MCS.SIGNON and MCS.CLIST initialization parameters in the appropriate initialization parameter file.

Customizing the Product

After you verify the installation, you may want to customize files, screens, and Processes. The following list guides you to additional instructions for customizing your installation:

Procedure

1. Fully define all nodes to the network map file. For information on how to update the network map file, see *Maintaining the Network Map*.
2. Set up the IBM Connect:Direct/Plex environment, if applicable.
3. Select the appropriate step to set up security:
 - If you use the IBM Connect:Direct Authorization Facility, add users to the User Authorization file.
 - If you use external security software such as RACF, ACF2, or TOP SECRET, see *Implementing Security* for more information.
4. Add types to the Type file. See *Maintaining the Type File* to add records to the Type file.
5. Customize the Messages file. See *Customizing IBM Connect:Direct* in *IBM Connect:Direct for z/OS Administration Guide*.
6. Customize the SUBMIT screen.

7. Customize the sample Processes. Information on IBM Connect:Direct Processes is available in the *IBM Connect:Direct for z/OS Process Language Reference Guide* and *IBM Connect:Direct for z/OS User Guide*.

VTAM Definitions

Before starting IBM Connect:Direct, the VTAM application definitions must be active, the mode table entries must be completed, and the cross-domain resources must be active. This section explains how to set up VTAM definitions for IBM Connect:Direct.

The following VTAM definitions may be required for each node. The member name of each definition is listed in parentheses. The sample definitions are in the file \$CD.SDGACNTL.

- VTAM definitions for IBM Connect:Direct DTF (DGACAPPL)
- VTAM definitions for IUI or batch interface (DGACIAPP)
- Mode table used with IBM Connect:Direct (DGACMODT)
- Cross-domain resource manager node (DGACCDRM)
- Cross-domain resource definition for other nodes (DGACCDRS)
- VTAM definition for PNODE=SNODE, also known as loop-back, processing (DGACAPPL)

If you are setting up VTAM definitions for mainframe-to-PC-only connections, disregard the information about multiple z/OS or VM sites within the IBM Connect:Direct network and VTAM cross-domain definitions.

Note: If you went through the installation process and generated the network map, refer to the member DGAJNET0 in \$CD.SDGACNTL for network map definitions that use the VTAM definitions.

Defining APPLID of Local DTF

Define the APPLID of the local IBM Connect:Direct DTF. A sample definition for the APPLID of the local DTF follows.

This example is located in the member DGACAPPL of \$CD.SDGACNTL.

CDAPP4 APPL	ACBNAME=CDAPP4, VTAM APPLICATION ID	X
	APPC=YES, ENABLE TO RUN LU6.2 SESSIONS	X
	AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VPACE),	X
	EAS=n, APPROXIMATE # OF CONCURRENT SESS	X
	MODETAB=CDMTAB, MODE TABLE NAME	X
	SONSCIP=NO, NO UNBIND IN SCIP EXIT	X
	SRBEXIT=NO, NO SRB PROCESSING	X
	VPACING=n, RECEIVE PACING	X
	DLOGMOD=CDLOGM, MODE TABLE ENTRY	X
	PARSESS=YES, PARALLEL SESSIONS CAN BE USED	X
	DSESLIM=n, # OF CONCURRENT LU6.2 SESSIONS	X
	DMINWNL=n, # OF LOCAL CONTENTION WINNERS	X
	DMINWNR=n, # OF REMOTE CONTENTION WINNERS	X
	AUTOSES=n, # OF AUTOMATIC LU6.2 SESSIONS	X
	DDRAINL=ALLOW, ALLOW CNOS TO DRAIN SESSIONS	X
	DRESPL=ALLOW, DEF RESPONSIBILITY FOR LOCAL CNOS	X
	LMDENT=n, SIZE OF HASH TABLE	X
	VTAMFRR=NO	

Optimal settings are:

- When using IBM Connect:Direct Extended Recovery, change CDAPP4 to CDAPP* to make the APPL dynamic. This change enables the IBM Connect:Direct extended recovery standby to monitor the active IBM Connect:Direct image from a different z/OS image and to take over for that active IBM Connect:Direct image if it fails. This setting applies to both stand-alone servers and IBM Connect:Direct/Plex environments
- APPC=YES enables LU6.2 support and is required.

- EAS represents the approximate number of concurrent sessions that this APPL has with all other APPLs (Connect:Direct adjacent nodes and interactive users). The default value is 509. Change the value to one that is realistic for your environment.
- MODETAB must point to the mode table that contains entries for LU0, LU6.2, and SNA Service Manager mode (SNASVCMG). See [Define Logmode Table Entries](#) for the definition of the mode table.
- Set VPACING to the number of RUs that this DTF receives before being obligated to send a pacing response. For best results, use a minimum value of 7. Larger values increase throughput when receiving data from another DTF.

Do not specify a value of 0, or omit this parameter, which causes no pacing and can lead to a VTAM buffer shortage.

DLOGMOD (the default log mode) must refer to an LU0 logmode entry to ensure compatibility with prior releases of IBM Connect:Direct. The LU6.2 logmode name is extracted from the network map.

- PARSESS=YES parameter is required.
- Set DSESLIM to the sum of the values specified in DMINWNL and DMINWNR so that DSESLIM is equal to or greater than the largest PARSESS max value in the network map.
- Set DMINWNL to a value that is no larger than one-half the value of DSESLIM.
- Set DMINWNR to a value that is no larger than one-half the value of DSESLIM.
- Set AUTOSSES to 1 if this DTF is used in a mixed environment of sending and receiving files. Set AUTOSSES to 0 if this DTF is used primarily to receive files.

Note: Use this DTF to send files if you set AUTOSSES to 0.

- DDRAINL=ALLOW enables Change Number of Sessions (CNOS) to drain sessions.
- DRESPL=ALLOW enables the application program to accept responsibility for deactivating sessions.
- LMDENT specifies the number of entries to be used for this application program's hash table of remote LUs. The default value is 19.

APPLID for IUI and Batch Sessions

Define APPLID for IUI and batch session definition.

Note: Add ENCR=NONE to VTAM APPLs used for IUI sessions to avoid signon problems.

The following example shows three VTAM application definitions for the IBM Connect:Direct IUI. These application IDs must match those specified in the IBM Connect:Direct network map. The examples are located in member DGACIAPP of \$CD.SDGACNTL.



CAUTION: In the example below, ACBNAME and the label names are representations. You need to change them to represent your own installation.

NAI01	APPL	ACBNAME=NAI01, DLOGMOD=CDLOGM, MODETAB=CDMTAB ENCR=NONE	X X
*			
NAI02	APPL	ACBNAME=NAI02, DLOGMOD=CDLOGM, MODETAB=CDMTAB ENCR=NONE	X X
*			
NAI03	APPL	ACBNAME=NAI03, DLOGMOD=CDLOGM, MODETAB=CDMTAB ENCR=NONE	X X

Defining APPLID for Loop-Back Processing

If you perform PNODE=SNODE (loop-back) processing, define the APPLID to use for the processing.

The following figure shows an additional IBM Connect:Direct DTF APPL (CDAPP2) defined to VTAM that enables you to perform loop-back processing. This multiple definition is required because with PNODE=SNODE processing the communications name (APPLID) for the adjacent node defined in the network map must be different than the communications name (APPLID) for the local node in the network map. This example is located in member DGACAPPL of \$CD.SDGACNTL.

CDAPP2	APPL	ACBNAME=CDAPP2,	VTAM APPLICATION ID	X
		AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VSPACE),		X
		EAS=n,	APPROXIMATE # CONCURRENT SESS	X
		MODETAB=CDMTAB,	MODE TABLE NAME	X
		SONSCIP=NO,	NO UNBIND IN SCIP EXIT	X
		SRBEXIT=NO,	NO SRB PROCESSING	X
		VPACING=7,	RECEIVE PACING OF 7	X
		DLOGMOD=CDLOGM,	MODE TABLE ENTRY	X
		PARSESS=YES,	PARALLEL SESSIONS CAN BE USED	X
		VTAMFRR=NO		

Defining Logmode Table Entries

Define the logmode table entries used with each APPLID. This step is required if you communicate with a node defined to VTAM as a PU TYPE 4 node.

The following sections provide examples from the mode table used with IBM Connect:Direct. The examples are located in member DGACMODT.

You can use the optional name (CDMTAB) as an assembler CSECT name for the mode table. You can assemble the table separately from other mode tables or you can insert the IBM Connect:Direct entry into an existing mode table.

Note: Modify COS, PACING, and RUSIZE values only.

z/OS Nodes

Use the following entry with other Connect:Direct for z/OS nodes.

Note: Connect:Direct Secure Plus is not supported for SNA connections.

CDLOGM	MODEENT	LOGMODE=CDLOGM,	ENTRY NAME	X
		TYPE=1,	NON-NEGOTIABLE BIND	X
		FMPROF=X'04',	FUNCTION MGMT. PROFILE 4	X
		TSPROF=X'04',	TRANS SERVICE PROFILE 4	X
		PRIPROT=X'B3',	PRIMARY PROTOCOL	X
		SECPROT=X'B3',	SECONDARY PROTOCOL	X
		COMPROT=X'6080',	COMMON PROTOCOL	X
		PSNDPAC=X'06',	PRIMARY SEND PACING	X
		SRVPAC=X'06',	SECONDARY SEND PACING	X
		RUSIZES=X'8989',	4K MAX RU SIZE FOR PRI AND SEC	X
		PSERVIC=X'000000000000000000000000'		

OpenVMS and HP NonStop (Tandem) Nodes

Use the following entry with OpenVMS and HP NonStop (Tandem) nodes.

CDVMS	MODEENT	LOGMODE=CDVMS,		X
		TYPE=1,		X
		FMPROF=X'04',		X
		TSPROF=X'04',		X
		PRIPROT=X'B1',		X
		SECPROT=X'B1',		X
		COMPROT=X'7080',		X
		SRCVPAC=X'04',	SRCVPAC,SSNDPAC, AND PSNDPAC	X
		SSNDPAC=X'04',	VALUES CAN BE TUNED	X
		PSNDPAC=X'04',		X
		RUSIZES=X'8989',	4K SEND AND RECV RUSIZE	X
		PSERVIC=X'000000000000000000000000'		

i5/OS SNUF (LU0) Nodes

Use the following entry with i5/OS SNUF (LU0) nodes.

SNUF4K	MODEENT LOGMODE=SNUF2K,	X
	COS=NJE,	X
	FMPROF=X'04',	X
	TSPROF=X'04',	X
	PRIPROT=X'B1',	X
	SECPROT=X'B1',	X
	COMPROT=X'7080',	X
	SRCVPAC=X'07', SRCVPAC,SSNDPAC, AND PSNDPAC	X
	SSNDPAC=X'07', VALUES CAN BE TUNED	X
	PSNDPAC=X'07',	X
	RUSIZES=X'8888', 2K SEND AND RECV RUSIZE	X
	PSERVIC=X'000000000000000000000000'	

LU6.2 Nodes

Use the following entry with LU6.2 independent nodes.

CD624K	MODEENT LOGMODE=CD624K,	X
	TYPE=1,	X
	COS=NJE,	X
	FMPROF=X'13',	X
	TSPROF=X'07',	X
	PRIPROT=X'B0',	X
	SECPROT=X'B0',	X
	COMPROT=X'D0B1',	X
	PSNDPAC=X'04',	X
	SRVCPAC=X'04',	X
	SSNDPAC=X'04',	X
	RUSIZES=X'8989', 4K SEND AND RECV RUSIZE	X
	PSERVIC=X'060200000000000000000000300'	

Use the following entry with LU6.2 dependent nodes.

CD624K	MODEENT LOGMODE=CD624K,	X
	TYPE=1,	X
	COS=NJE,	X
	FMPROF=X'13',	X
	TSPROF=X'07',	X
	PRIPROT=X'B0',	X
	SECPROT=X'B0',	X
	COMPROT=X'50B1',	X
	RUSIZES=X'8989', 4K SEND AND RECV RUSIZE	X
	PSERVIC=X'0602000000000000000000002C00'	

LU6.2 SNA Services Manager

LU6.2 requires the following SNA Services Manager mode table entry definition. This LOGMODE is for VTAM use. You must make this definition in the Mode Table, but do not specify it as the LOGMODE in the network map.

SNASVCMG	MODEENT LOGMODE=SNASVCMG,	X
	TYPE=1,	X
	COS=NJE,	X
	FMPROF=X'13',	X
	TSPROF=X'07',	X
	PRIPROT=X'B0',	X
	SECPROT=X'B0',	X
	COMPROT=X'D0B1',	X
	RUSIZES=X'8585', 256 SEND AND RECV RUSIZE	X
	PSERVIC=X'060200000000000000000000300'	

Cross-Domain Resource Manager Minor Nodes

Define the cross-domain resource manager nodes. Following are the IBM Connect:Direct Cross-Domain Resource Manager node definitions. The sample is located in member DGACCDRM.

```
CDRMA    VBUILD TYPE=CDRM
AAAA     CDRM  SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
BBBB     CDRM  SUBAREA=30,ELEMENT=1,ISTATUS=ACTIVE
CCCC     CDRM  SUBAREA=40,ELEMENT=1,ISTATUS=ACTIVE
```

Each CDCDRM entry defines another VTAM domain or equivalent with which this domain communicate, in the domain in which this IBM Connect:Direct is being defined.

Defining Cross-Domain Resources

Define the cross-domain resources. The following figure shows a IBM Connect:Direct Cross-Domain Resource (CDRSC) definition. The sample is located in member DGACCDRS. Each CDRSC entry defines an applications (another IBM Connect:Direct) in another domain.

```
CDRSCNA  VBUILD TYPE=CDRSC
CDAPP2   CDRSC CDRM=BBBB
CDAPP3   CDRSC CDRM=CCCC
```

IBM Connect:Direct does not use the Unformatted Systems Services (USS) table; however, if you define a USS table for the LUs to be used with AS/400 communication devices, or with OpenVMS or Tandem logical units, you cannot include a MSG10 definition in the table.

Defining APPLIDS for Multiple DTFs

One VTAM APPL is required for each IBM Connect:Direct DTF, along with an optional APPL for PNODE=SNODE processing. If the configuration consists of two or more DTFs, or if a IBM Connect:Direct/Plex is configured, multiple VTAM APPLs must be defined.

When defining APPLs for multiple DTFs or a IBM Connect:Direct/Plex, the APPLs can all reside in the same application major node (VBUILD TYPE=APPL) in SYS1.VTAMLST if necessary.

Defining Network or Domain Names

The VTAM APPL definitions for IBM Connect:Direct nodes are in the supplied examples (in \$CD.SDGACNTL) with the name field value (column 1) equal to the ACBNAME value. However, these names may be different. See [Network or Domain Name in Cross-Domain Network](#) for an example of VTAM and network map Cross Domain definitions and the rules that govern the definitions.

VTAM and NCP Parameters

This step explains how to set up VTAM and NCP parameters for IBM Connect:Direct operation. You must perform the following evaluations to prepare for IBM Connect:Direct operation:

- Determine correct RU size for SNA sessions.

The VTAM RUSIZES parameter is located in the VTAM MODEENT macro. It specifies the transmission buffer size and can affect IBM Connect:Direct operation. See [Selecting RU Size for SNA Sessions](#) that follows.

- Determine the effect of NCP parameters.

Several parameters located in the NCP macros used during NCP generation (GEN) can also affect IBM Connect:Direct operation. See [Effects of NCP Parameters](#).

Selecting RU Size for SNA Sessions

The request/response unit (RU) size for IBM Connect:Direct is specified in the VTAM RUSIZES parameter in the VTAM log mode table. The following example shows a mode table entry with a secondary and primary logical unit RU size of 1,024 bytes.

CDLOGM	MODEENT LOGMODE=CDLOGM, RUSIZES=X'8787'	X
--------	--	---

When selecting an RU size for IBM Connect:Direct, it is important to know how RU size relates to the NCP MAXDATA parameter.

The NCP MAXDATA value is specified in the NCP physical unit (PU) macro definition. It specifies, in bytes, the maximum amount of data that the NCP can receive from the PU at one time. This amount includes the transmission header and the request/response header, totaling 29 bytes for IBM Connect:Direct.

For example, if the IBM Connect:Direct VTAM log mode table entry specifies 4,096 bytes, the NCP MAXDATA value must be at least 4,125 bytes for IBM Connect:Direct to function correctly. In other words, the NCP MAXDATA value must be at least 29 bytes larger than the RU size specified in the IBM Connect:Direct VTAM log mode table.

- The minimum RU size value for IBM Connect:Direct API-to-DTF transmissions is 512 bytes. The maximum RU size value is 64 KB bytes. For best results, use a value of 7 KB.

Note: For locally attached non-SNA 3270 connections, specify RUSIZES=X'87C7'.

Understanding RU Sizing

The following table shows how to find the appropriate value to specify for the RUSIZES parameter. For the formula RUSIZES=X'abab', the first ab pair applies to the secondary logical unit (SLU). The second ab pair applies to the primary logical unit (PLU). The table shows the values for a and b. The number of bytes is found where the a column and the b column intersect in the table.

For example, the bold a and b numbers in the table indicate how it is used to determine the RUSIZES=87F8 for SNA connections. Because the primary LU is able to receive 3,840 bytes, the secondary LU can send 3,840 bytes at a time (the intersection of F and 8). The primary LU can send 1,024 bytes at a time because the secondary LU is able to receive 1,024 bytes (the intersection of 8 and 7).

	a							
b	8	9	A(10)	B(11)	C(12)	D(13)	E(14)	F(15)
0	8	9	10	11	12	13	14	15
1	16	18	20	22	24	26	28	30
2	32	36	40	44	48	52	56	60
3	64	72	80	88	96	104	112	120
4	128	144	160	176	192	208	224	240
5	256	288	320	352	384	416	448	480
6	512	576	640	704	768	832	896	960
7	1,024	1,152	1,280	1,408	1,536	1,664	1,792	1,920
8	2,048	2,304	2,560	2,816	3,072	3,328	3,584	3,840
9	4,096	4,608	5,120	5,632	6,144	6,656	7,168	7,680
A(10)	8,192	9,216	10,240	11,264	12,288	13,312	14,336	15,360
B(11)	16,384	18,432	20,480	22,528	24,576	26,624	28,672	30,720

	a							
b	8	9	A(10)	B(11)	C(12)	D(13)	E(14)	F(15)
C(12)	32,768	36,864	40,960	45,056	49,152	53,248	57,344	61,440
D(13)	65,536	73,728	81,920	90,112	98,304	106,496	114,688	122,880
E(14)	131,072	147,456	163,840	180,224	196,608	212,992	229,376	245,760
F(15)	262,144	294,912	327,680	360,448	393,216	425,984	458,752	491,520

The table is derived from the following calculations:

The RUSIZE of ab means RUSIZE equals $a \times (2^{**}b)$. The b of each ab pair is used as an exponent of base two. The resulting value is multiplied by a to get the RUSIZE for that logical unit.

Using the same example, RUSIZES=87F8, the SLU and PLU values are determined as follows.

if RUSIZES=X'87F8' then

the SLU RUSIZE is 87

which indicates $8 \times 2^{**7} = 8 \times 128 = 1024$ (8 times 2 to the 7th power)

the PLU RUSIZE is F8

which indicates $F \times 2^{**8} = 15 \times 2^{**8} = 15 \times 256 = 3840$ (15 times 2 to the 8th power)

Effects of NCP Parameters

Review the following NCP parameters carefully during the installation:

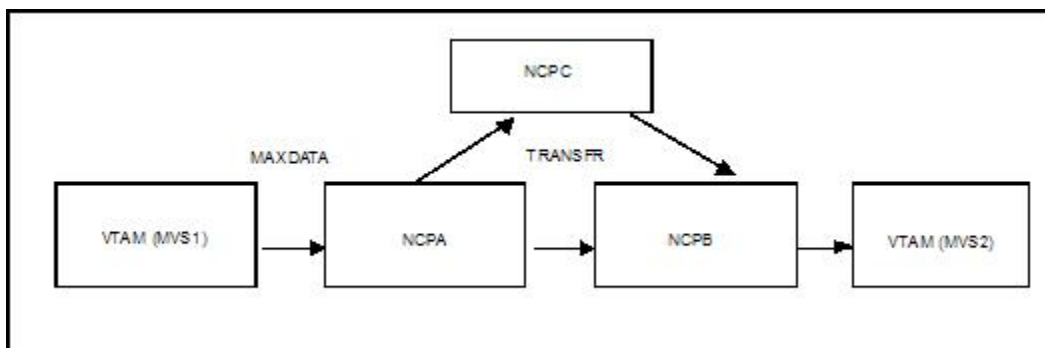
- BFRS in the BUILD GEN macro that defines the size of NCP buffers
- BFRPAD in the HOST GEN macro that defines the number of pad characters inserted by NCP
- MAXBUFRU in the HOST GEN macro that defines the number of buffers the access method (VTAM) allocates to receive data from the NCP
- UNITSZ in the HOST (VTAM) GEN macro that defines the size of access method buffers used for data transfer from NCP to IBM Connect:Direct
- TRANSFR in the LINE or BUILD GEN macro that defines the number of NCP buffers corresponding to the maximum amount of data NCP can receive from another NCP

MAXDATA in the PU unit GEN macro that defines the maximum path information (PIU) size

The following figure illustrates the relationship between these VTAM and NCP parameters. When data is passed from MVS1 to NCPA, the MAXDATA parameter determines the amount of data that NCPA can receive in one segment of a path information unit (PIU).

Between NCPA, NCPB, and NCPC, if you do not specify the TRANSFR parameter on the LINE statement, VTAM searches the BUILD statement. If you do not define TRANSFR on either statement, the default is taken. See the VTAM Customization manual.

When NCPB passes data to MVS2, the MAXBUFRU times the UNITSZ determines the amount of data that can be passed to MVS2. The data cannot exceed the size in bytes of the VTAM IOBUF buffer pool allocation parameters.



Calculating Minimum Value of NCP TRANSFR

To calculate the minimum value of an NCP TRANSFR, define each NCP-to-NCP connection to accommodate the maximum RU size defined for a IBM Connect:Direct DTF-to-DTF session. This value is controlled by the NCP TRANSFR parameter of each NCP LINE macro defining an NCP-to-NCP link. The following is a summary of the calculations required to determine the NCP TRANSFR value:

Procedure

1. Determine the maximum amount of data to be received on this line definition. For IBM Connect:Direct, this amount is the RU size plus 29 bytes for the request/response headers.
2. Add 24 bytes to this value for required NCP overhead (BFRPAD).
3. Divide the sum by the NCP buffer size (BFRS).
4. Round the result to the next highest integer. This integer is the minimum value that you can specify for TRANSFR that corresponds to the specified RU size.

Changing the TRANSFR Parameter

If you change the TRANSFR parameter in one NCP in a network, all other NCPs in that network require the same change. If you cannot easily change the NCP TRANSFR parameter, use the following reverse process to calculate the maximum RU size for use by IBM Connect:Direct:

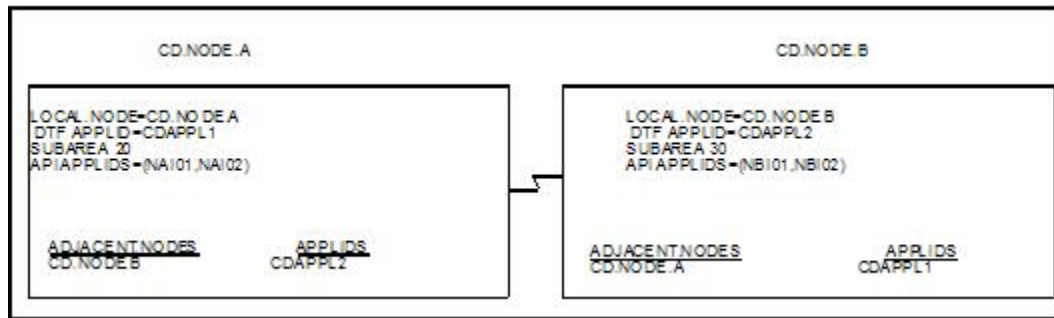
Procedure

1. Multiply the NCP TRANSFR parameter by the NCP BFRS value. The result is the maximum amount of data that NCP can receive.
2. Subtract 24 bytes for required NCP overhead (BFRPAD).
3. Subtract 29 bytes for the request/response header.
4. Find the next lowest RU size value in the table on [Understanding RU Sizing](#). This value is the largest RU that you can specify for use by IBM Connect:Direct.

The amount of data VTAM can receive from the NCP is determined by the product of MAXBUFRU times UNITSZ. This value must be less than or equal to the amount specified on the IOBUF buffer pool allocation parameters in the ATCSTRxx start options list of VTAMLST.

VTAM Definitions for Full Networking IBM Connect:Direct

Following is a sample IBM Connect:Direct network with SNA connections.



The VTAM definitions in this section are based on this LU0 example.

Note: Do not make the node name the same as the VTAM APPLID.

Node A Definitions

Following is the DTF APPL definition for CD.NODE.A:

```
CDAPPL1 APPL ACBNAME=CDAPPL1,           X
        AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VSPACE), X
        EAS=20,                          APPROXIMATE # CONCURRENT SESS X
        MODETAB=CDMTAB,                  MODE TABLE NAME           X
        SONSCIP=NO,                      NO UNBIND IN SCIP EXIT       X
        SRBEXIT=NO,                      NO SRB PROCESSING           X
        VPACING=7,                      RECEIVE PACING OF 7         X
        DLOGMOD=CDLOGM,                  MODE TABLE ENTRY           X
        PARSESS=YES,                    PARALLEL SESSIONS CAN BE USED X
        VTAMFRR=NO
```

The following are the APPL definitions for the API for CD.NODE.A.

```
NAI01    APPL ACBNAME=NAI01,           X
        DLOGMOD=CDLOGM,                X
        MODETAB=CDMTAB
        ENCR=NONE
*
NAI02    APPL ACBNAME=NAI02,           X
        DLOGMOD=CDLOGM,                X
        MODETAB=CDMTAB
        ENCR=NONE
```

The following is the Cross-Domain Manager definition for CD.NODE.A.

```
CDRMA VBUILD TYPE=CDRM
SA30   CDRM   SUBAREA=30,ELEMENT=1,ISTATUS=ACTIVE
SA20   CDRM   SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
```

The following is the Cross-Domain definition for CD.NODE.A.

```
CDRSCA VBUILD TYPE=CDRSC
CDAPPL2 CDRSC CDRM=SA30
```

The following is the network map definition for CD.NODE.A.

LOCAL.NODE=((CD.NODE.A CDAPPL1 , , \$PW)	-
TCQ=(TCX.FILE TCQ.FILE))	
* PNODE=SNODE DEFINITION	*
ADJACENT.NODE=((CD.NODE.A CDAPPL2)	-
PARSESS=(12 2)	-
APPLIDS=(NAI01,NAI02))	
* SNA CONNECTIONS	*
ADJACENT.NODE=((CD.NODE.B CDAPPL)	-
PARSESS=(6 3)	-
APPLIDS=(NAI01,NAI02))	

Node B Definitions

Following is the DTF APPL definition for CD.NODE.B:

CDAPP2 APPL ACBNAME=CDAPP2,	X
AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VSPACE),	X
EAS=20,	X
APPROXIMATE # CONCURRENT SESS	X
MODETAB=CDMTAB,	X
MODE TABLE NAME	X
SONSCIP=NO,	X
NO UNBIND IN SCIP EXIT	X
SRBEXIT=NO,	X
NO SRB PROCESSING	X
VPACING=7,	X
RECEIVE PACING OF 7	X
DLOGMOD=CDLOGM,	X
MODE TABLE ENTRY	X
PARSESS=YES,	X
PARALLEL SESSIONS CAN BE USED	X
VTAMFRR=NO	

The following are the APPL definitions for the API for CD.NODE.B.

NBI01 APPL ACBNAME=NBI01,	X
DLOGMOD=CDLOGM,	X
MODETAB=CDMTAB	
*	
NBI02 APPL ACBNAME=NBI02,	X
DLOGMOD=CDLOGM,	X
MODETAB=CDMTAB	

The following is the Cross-Domain Manager definition for CD.NODE.B.

```
CDRMB VBUILD TYPE=CDRM
SA30 CDRM SUBAREA=30,ELEMENT=1,ISTATUS=ACTIVE
SA20 CDRM SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
```

The following is the Cross-Domain definition for CD.NODE.B.

```
CDRSCA VBUILD TYPE=CDRSC
CDAPPL1 CDRSC CDRM=SA20
```

The following is the network map definition for CD.NODE.B.

LOCAL.NODE=((CD.NODE.B CDAPPL2 , , \$PW)	-
TCQ=(TCX.FILE TCQ.FILE))	
* PNODE=SNODE DEFINITION *	
ADJACENT.NODE=((CD.NODE.B CDAPPL2B)	-
APPLIDS=(NBI01,NBI02))	
* SNA CONNECTIONS *	
ADJACENT.NODE=((CD.NODE.A CDAPPL1)	-
APPLIDS=(NBI01,NBI02))	

Network or Domain Name in Cross-Domain Network

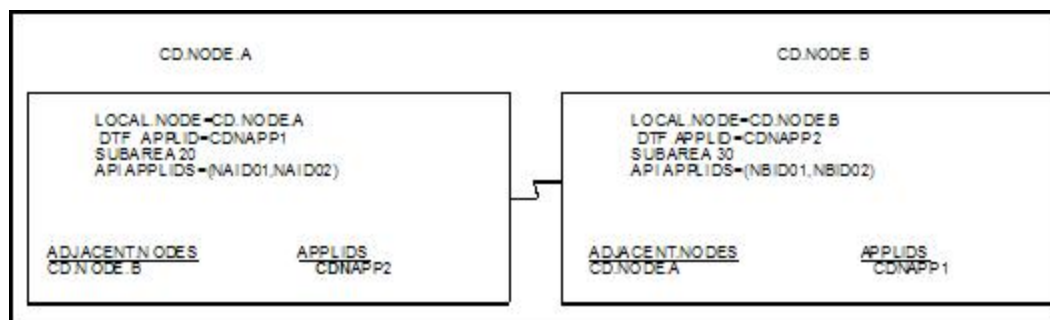
The VTAM APPL definitions for IBM Connect:Direct nodes are in the supplied examples (in \$CD.SDGACNTL) with the name field value (column 1) equal to the ACBNAME value. However, these

names may be different. If they are different, the network name value contained in the name field must be unique within the network, and the value in the ACBNAME field must be unique within the domain.

If the names are different, follow these rules when defining the network map:

- The name specified for the local node's DTF APPLID must be the domain name.
- The name specified for adjacent nodes' DTF APPLIDs must be the network name.
- The names specified in the API APPLIDS keyword of the adjacent node must be the domain names.

The following sample illustrates definitions for two nodes with unique network and domain names:



The VTAM definitions in this section are based on this example.

In the following VTAM definition examples, domain names contain a D as part of their name and network names contain an N.

SNA Node A

Following is the DTF APPL definition for CD.NODE.A:

```

CDNAPP1 APPL ACBNAME=CDDAPP1,
AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VSPACE), X
EAS=20, APPROXIMATE # CONCURRENT SESS X
MODETAB=CDMTAB, MODE TABLE NAME X
SONSCIP=NO, NO UNBIND IN SCIP EXIT X
SRBEXIT=NO, NO SRB PROCESSING X
VPACING=7, RECEIVE PACING OF 7 X
DLOGMOD=CDLOGM, MODE TABLE ENTRY X
PARSESS=YES, PARALLEL SESSIONS CAN BE USED X
VTAMFRR=NO
  
```

The following are the APPL definitions for the API for CD.NODE.A.

```

NAN01 APPL ACBNAME=NAID01, X
DLOGMOD=NDMLOGM, X
MODETAB=NDMTAB
ENCR=NONE
*
NAN02 APPL ACBNAME=NAID02, X
DLOGMOD=CDLOGM, X
MODETAB=CDMTAB
ENCR=NONE
  
```

Following is the Cross-Domain Manager definition for CD.NODE.A:

```

CDRMA VBUILD TYPE=CDRM
SA30 CDRM SUBAREA=30,ELEMENT=1,ISTATUS=ACTIVE
SA20 CDRM SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
  
```

Following is the Cross-Domain definition for CD.NODE.A:

```
CDRSCA VBUILD TYPE=CDRSC
CDNAPP2 CDRSC CDRM=SA30
```

Following is the network map definition for CD.NODE.A:

```
LOCAL.NODE=((CD.NODE.A   CDNAPP1 , , $PW)           -
              TCQ=(CD.NODEA.TCX CD.NODEA.TCQ))
*
*
ADJACENT.NODE=((CD.NODE.A   CDNAPP1)                 -
                APPLIDS=(NAID01,NAID02))
*
*
ADJACENT.NODE=((CD.NODE.B   CDNAPP2)                 -
                APPLIDS=(NBID01,NBID02))
```

SNA Node B

Following is the DTF APPL definition for CD.NODE.B:

```
CDNAPP2 APPL ACBNAME=CDDAPP2,
AUTH=(ACQ,NOCNM,NOPASS,NOPO,NOTSO,VPACE),
EAS=20, APPROXIMATE # CONCURRENT SESS           X
MODETAB=CDMTAB, MODE TABLE NAME                X
SONSCIP=NO, NO UNBIND IN SCIP EXIT              X
SRBEXIT=NO, NO SRB PROCESSING                   X
VPACING=7, RECEIVE PACING OF 7                  X
DLOGMOD=CDLOGM, MODE TABLE ENTRY               X
PARSESS=YES, PARALLEL SESSIONS CAN BE USED      X
VTAMFRR=NO
```

Following are the APPL definitions for the API for CD.NODE.B:

```
NBN01 APPL ACBNAME=NBID01,                        X
DLOGMOD=CDLOGM,                                    X
MODETAB=CDMTAB
NBN02 APPL ACBNAME=NBID02,                        X
DLOGMOD=CDLOGM,                                    X
MODETAB=CDMTAB
```

Following is the Cross-Domain Manager definition for CD.NODE.B:

```
CDRMB VBUILD TYPE=CDRM
SA30 CDRM SUBAREA=30,ELEMENT=1,ISTATUS=ACTIVE
SA20 CDRM SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
```

Following is the Cross-Domain definition for CD.NODE.B:

```
CDRSCA VBUILD TYPE=CDRSC
CDNAPP1 CDRSC CDRM=SA20
```

Following is the network map definition for CD.NODE.B:

```
LOCAL.NODE=((CD.NODE.B   CDNAPP2 , , $PW)           -
              TCQ=(CD.NODEB.TCX CD.NODEB.TCQ))
*
*
ADJACENT.NODE=((CD.NODE.B   CDNAPP2)                 -
                APPLIDS=(NBID01,NBID02))
*
*
ADJACENT.NODE=((CD.NODE.A   CDNAPP1)                 -
                APPLIDS=(NAID01,NAID02))
```

Optional Configuration Tasks

Customizing the Product

After you verify the installation, you may want to customize files, screens, and Processes. The following list guides you to additional instructions for customizing your installation:

1. Fully define all nodes to the network map file. For information on how to update the network map file, see *Maintaining the Network Map*.
2. Set up the IBM Connect:Direct/Plex environment, if applicable.
3. Select the appropriate step to set up security:
 - If you use the IBM Connect:Direct Authorization Facility, add users to the User Authorization file.
 - If you use external security software such as RACF, ACF2, or TOP SECRET, see *Implementing Security* for more information.
4. Add types to the Type file. See *Maintaining the Type File* in *IBM Connect:Direct for z/OS Administration Guide* to add records to the Type file.
5. Customize the Messages file. See *Customizing IBM Connect:Direct* in *IBM Connect:Direct for z/OS Administration Guide*.
6. Customize the SUBMIT screen.
7. Customize the sample Processes. Information on IBM Connect:Direct Processes is available in the *IBM Connect:Direct for z/OS Process Language Reference Guide*.

Installing Connect:Direct File Agent

About this task

Before you install Connect:Direct File Agent, review the requirements in [Planning for Connect:Direct File Agent](#). After you customize Connect:Direct File Agent, see the Connect:Direct File Agent Help for configuration instructions.

Note: You must download and store the Connect:Direct File Agent code in the HFS directory prior to executing the DGA#FINS CLIST. This HFS directory path is required as menu input in the DGA#FINS CLIST.

Access to HFS files is controlled by z/OS UNIX System Services, which enables or denies access based on UNIX permission rules. The installer and users of Connect:Direct File Agent must have the appropriate permissions.

To install Connect:Direct File Agent:

Procedure

1. To download the Java version of the Connect:Direct File Agent component, log on to IBM Passport Advantage Online and navigate to IBM Connect:Direct Product Updates/Downloads. After you have downloaded and saved the .zip file to your PC, decompress it to get the JAR file.
2. Upload the FAInstall.jar to an HFS directory using FTP in binary mode.
3. Execute the installer from a standard OMVS prompt or TELNET session into a UNIX ISHELL environment and enter `java -jar FAInstall.jar`.



CAUTION: The installer requires a region of at least 210 MB. If your region is less than the minimum, make arrangements to define a larger region.

4. Follow the prompts to install Connect:Direct File Agent.
5. After installing Connect:Direct File Agent, execute the DGA#FINS CLIST to build the appropriate JCL to configure and execute Connect:Direct File Agent in a z/OS environment.

Customizing Connect:Direct File Agent

To customize Connect:Direct File Agent:

Procedure

1. From the TSO COMMAND option of ISPF/PDF, execute the DGA#FINS CLIST, where \$CD is the high-level qualifier used to install Connect:Direct for z/OS files, as in the following example.

```
=== > EXEC '$CD.SDGAISPC(DGA#FINS)' '$CD'
```

During the installation process, you will create and name a Connect:Direct File Agent JCL dataset . All installation variables are saved in this JCL. The following table describes these installation variables:

Field	Description
Configuration job	Name of the job that starts the Connect:Direct File Agent Configuration Interface.
Execution job	Name of the job that start Connect:Direct File Agent.
Shutdown job	Name of the job that shuts down Connect:Direct File Agent.
File Agent home	Full path of the directory where Connect:Direct File Agent is installed.
X11 Display variable	<p>X11 display variable used to connect to the X11 GUI server. The Connect:Direct File Agent Configuration Interface displays on the monitor that matches this display variable. You can specify the netowrk ID of the terminal that you want to use for the Connect:Direct File Agent Configuration Interface.</p> <p>Note: If you want to display the Connect:Direct File Agent Configuration Interface on a Windows computer, you can specify the first node of the Full Computer Name. To obtain this information, right-click on My Computer, click Properties., and then click the Network Identification tab. The Full Computer Name is displayed.</p>
Job card information	JCL used for the job card.

You can edit the Connect:Direct File Agent JCL dataset with ISPF to change an installation variable.

The Connect:Direct File Agent Installation Main Menu is displayed after all elements have been unloaded.

```

Connect:Direct File Agent v.rr.mm
Installation Main Menu
More:      +
TIME-hh:mm
DATE-yyyy/mm/dd

CMD ==>

This panel can be scrolled up and down to view the entire
set of fields to be entered

Press ENTER to continue, PF5 to Terminate the JCL Generation

File Agent jcl dataset...
Member name for
Configuration job.... CDFACONF
Execution job..... CDFAEXEC
Shutdown job..... CDFASHUT
Unix path for
File Agent home..... -----
..... -----
..... -----
X11 DISPLAY variable..... -----
Job Card Information..... //CDFA JOB (CDFA),'XX FILE AGENT',
..... // *
..... // *

```

2. To create the DGA#FINS JCL used to install Connect:Direct File Agent in the \$CD.SDGAJCL dataset, specify the fields and press **Enter**.
3. When the Connect:Direct File Agent JCL Generation Main Menu is displayed, press **Enter** to generate the File Agent JCL.

```

Connect:Direct File Agent v.rr.mm
JCL Generation Main Menu
TIME-hh:mm TIME-hh:mm
DATE-yyyy/mm/dd

CMD ==>

Connect:Direct File Agent JCL Generation Complete, RC=0

Name          Description
-----
CDFA          The job to build the File Agent jobs

Review and submit member CDFA to finish the JCL generation.
CDFACONF, CDFAEXEC, and CDFASHUT will be created by job CDFA

Press ENTER to continue

```

After the JCL has been generated, the following panel lists the jobs created.

4. To install Connect:Direct File Agent in the appropriate HFS directory, execute the CDFA JCL member, which generates the CDFACONF, CDFAEXEC, and CDFASHUT JCL members. The CDFACONF, CDFAEXEC, and CDFASHUT job names are based on the JCL member names you entered in the Connect:Direct File Agent Installation Main Menu panel. The CDFA job name is taken from the job card information entered during the installation.

Note: The CDFACONF, CDFAEXEC, and CDFASHUT job names are based on the JCL member names you entered in the Connect:Direct File Agent Installation Main Menu panel. The CDFA job name is taken from the job card information entered during the installation.
5. Open an X11 window, and then run the CDFACONF job to execute the Connect:Direct File Agent GUI. This job copies files from the mainframe library into the UNIX (HFS) directory.

Configuring Connect:Direct File Agent

Configure Connect:Direct File Agent using the configuration interface. The *Sterling Connect:Direct File Agent Configuration Guide* describes how Connect:Direct File Agent works and contains sample configuration scenarios.

Before you begin

Before you implement Connect:Direct File Agent in a production environment, use the Connect:Direct File Agent Help to set up and verify the basic operation. The Connect:Direct File Agent Help contains the same information as the *Sterling Connect:Direct File Agent Configuration Guide*.

Connect:Direct File Agent is designed to run unattended. After you verify Connect:Direct File Agent operation and configuration, start it by running the CDFAEXEC job. Connect:Direct File Agent begins scanning the watch directory containing the directories and partitioned datasets you specify in the configuration file. When a file arrives in a watch directory, Connect:Direct File Agent either submits the default Process to IBM Connect:Direct or performs the actions specified by the rules for the file.

Backing Out the Connect:Direct File Agent Installation

It is not necessary to back out the Connect:Direct File Agent installation—if required, execute the DGA#FINS CLIST again.

Planning for Aspera FASP for Connect:Direct for z/OS

Aspera FASP is a high-speed transport for file transfer that avoids some of the slow-downs experienced with TCP, especially in networks with high latency and/or high packet loss.

IBM Connect:Direct UNIX (Linux and AIX) and IBM Connect:Direct Windows provide native support for utilizing FASP for the COPY step of a PROCESS. In order to support FASP with IBM Connect:Direct, the user must purchase and install a High Speed Adapter Option (HSAO) license.



Figure 1. Install a High Speed Adapter Option (HSAO) license

Support is being added to IBM Sterling Secure Proxy (SSP) that will allow IBM Connect:Direct FASP transfers to pass through SSP. Since HSAO licenses are required at each end of the transfer, no additional HSAO license is needed in SSP.

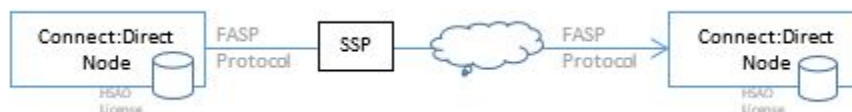


Figure 2. FASP transfers through SSP

SSP is designed in such a way that there is a complete session break between the two IBM Connect:Direct nodes. In this way, SSP offers the capability of having one side of the session using Secure+ while the other side of the session is not. This way, an internal IBM Connect:Direct can eliminate the overhead of encryption/decryption when passing through SSP and have SSP encrypt/decrypt that data this going over the public internet. Since SSP does a complete session break, there should be some way provided for SSP to use FASP protocol for one side of a session and the IBM Connect:Direct TCP protocol for the other side.

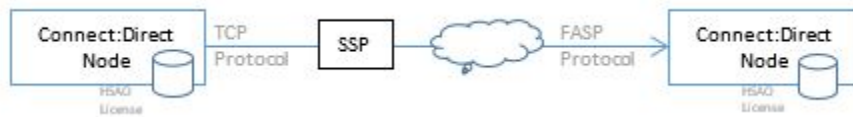


Figure 3. FASP transfers and session break between two IBM Connect Direct nodes

It should also be possible to allow two SSPs to use FASP across the public network without the two IBM Connect:Direct having to use FASP.



Figure 4. SSPs using FASP across public network

The Connect:Direct for z/OS Server requires the following definitions. See, *Connect:Direct for z/OS Administration Guide* for detail description of the FASP Initialization parameters.

Note: A HSAO license must be obtained and stored in the z/OS files system and available to the IBM Connect:Direct Server during initialization.

```
fasp=(yes|no|ssp , yes |no |ssp)
fasp.bandwidth=nnnn|<bandwidth_from_license>
fasp.filesize.threshold=nnn|1G
fasp.policy=fair|high|low|fixed
```

IBM Connect:Direct Server start must include a DD statement for the Aspera license file. For example,

```
//CDASPLIC DD DISP=SHR,DSN=$CD.HSAO.LICENSE
```

Note: To enable FASP via SSP the initialization parameter must be FASP=(SSP, SSP)

Customizing the Spool Transfer Feature

Ignore this procedure if your site does not use the Spool Transfer feature or if you are sending output to the JES reader from IBM Connect:Direct. For outbound transfers where you will be distributing print files from the JES Spool to IBM Connect:Direct, you must customize the Spool Transfer feature.

Note: For information on Spool Transfer, see *IBM Connect:Direct for z/OS Facilities Guide*.

Customizing the Spool Transfer feature consists of the following tasks:

1. Assemble DGASVPSA.
2. Customize VPS/CDI Option.
3. Restart VPS.
4. Customize the sample Processes.

Note: The Spool Transfer feature requires that you install VTAM Printer Support (VPS) and that you include the optional code in VPS that enables it to interface with IBM Connect:Direct.

Assembling DGASVPSA

Edit and modify the member DGAXAVPS in dataset \$CD.SDGASAMP to assemble and link the DGASVPSA module in the \$CD.SDGALINK dataset.

Before you begin

Note: The DGASVPSA module must be reassembled and linked when you upgrade VPS.

Customizing VPS/CDI

Following are the requirements for the VPS to CDI program (VPSSCDI). Add or modify the following library members. Refer to VPS documentation for explanations and requirements.

VPS.CNTL

This VPS library contains the following library members and parameters supported by IBM Connect:Direct:

Member	Description
VPSSTART	Contains the VPS system initialization parameters. The following keyword must be added to activate the VPSSCDI program. KEYCDI activates the VPSSCDI program.
MLISTMEM	This member contains the VPS printer activation member inclusion list. Add IBM Connect:Direct printers to this member.
VPSxxxx	These members contain the VPS printer initialization parameters.

The following table describes the keywords for a IBM Connect:Direct printer definition:

Keyword	Description
DDSNPFX	Required. Specifies the high-level qualifier VPS uses when creating the IBM Connect:Direct staged dataset. The default is VPS.
DEVTYPE=V.CDI	Required. Defines a printer as a IBM Connect:Direct printer.
CDNETMAP	Specifies the IBM Connect:Direct network map dataset name. The default is to use the DMNETMAP DD statement specified in VPS startup.
CDPLIB	Defines the IBM Connect:Direct process library dataset name. The default is to use the DMPUBLIB DD statement specified in VPS startup.
CDPMBR	Specifies the process member name to submit. The default is the printer name with the class appended.
CDSNODE	Defines the SNODE to be passed to the Process. The default is to use the SNODE defined in the Process.
DSPACE	Specifies the amount of DASD space to be allocated to the DASD dataset.
DUNIT	Specifies the unit type of the device on which the DASD is to be allocated.
DVOLUME	Specifies the volume on which the DASD is to be allocated.

VPS/CDI Interface Program

Assemble and link-edit module VPSSCDI into VPS LINKLIB using the sample JCL provided by LRS, and the IBM Connect:Direct macros provided in the SDGASAMP and SDGAMAC libraries.

VPS/CDI Startup Procedure

Add the following IBM Connect:Direct DD statements to the VPS startup procedure:

DD Statement	Description
DMMSGFIL	Required. IBM Connect:Direct message file.
SYSUT1	Required. DD UNIT=VIO, SPACE=(TRK,(2,1)).
DMPUBLIB	Required. IBM Connect:Direct Process library.
STEPLIB	IBM Connect:Direct LINKLIB.

DD Statement	Description
DMNETMAP	IBM Connect:Direct default network map.

Restarting VPS

After making changes to the VPS startup procedure, restart VPS to bring in the changes.

Customizing Sample Processes

Customize the sample Processes according to your environment. Information on IBM Connect:Direct Processes is available at .

Information on IBM Connect:Direct Processes is available at the [Connect:Direct Process Language](#) help.

HP OpenView for SNMP Traps Customization

Network Management applications do not recognize or display the traps without the IBM Connect:Direct trap configuration file and Management Information Block (MIB). Use the information to customize and load these files for HP OpenView Network Management.

Importing the IBM Connect:Direct Trap Configuration File

Perform this step if you are using SNMP Trap agent to communicate with the HP OpenView Network Node Manager.

Procedure

1. Transfer the Connect:Direct for z/OS configuration file, \$CD.SDGAMIB, to the HP UNIX workstation where HP OpenView is installed. This is a text file. Transfer it as CDtrap.conf.
2. Use the HP OpenView process, xnmevents, to load CDtrap.conf into the trapd.conf for HP OpenView (usually in the \$OV_DB directory).

Note: Refer to the HP OpenView documentation for correct syntax for the xnmevents process.

Import the IBM Connect:Direct MIB

Perform this step if you are using SNMP Trap agent to communicate with the HP OpenView Network Node Manager.

Procedure

1. Transfer the Connect:Direct for z/OS MIB, \$CD.SDGAMIB(DGAZMIB), to the workstation where HP OpenView is installed. This is a text file. Transfer it as DGAZMIB.mib.
2. Use the HP OpenView process, xnmloadmib, to load the Connect:Direct for z/OS MIB into the MIB database (usually in the \$OV_DB directory).

Note: Refer to the HP OpenView documentation for correct syntax for the xnmloadmib process.

NetView for SNMP Customization

Network Management applications do not recognize or display the traps without the IBM Connect:Direct trap configuration file and Management Information Block (MIB). Use the information to customize and load these files for the IBM® Tivoli® Netview application.

Customizing Tivoli NetView with the Tivoli Enterprise Console

Perform the following tasks to customize Tivoli NetView using the Tivoli Enterprise Console. Refer to the appropriate Tivoli NetView documentation for the proper command syntax.

Importing the IBM Connect:Direct Trap Configuration File

Transfer and execute the CDTrap_NetView.sh script to add the Connect:Direct for z/OS trap in to the trapd.conf file.

1. Transfer the file, \$CD.SDGATRP(DGAZTRAP), to the appropriate computer. This is a text file. Transfer it as CDTrap_NetView.sh.
2. Run the CDTrap_NetView.sh script to add the traps to the trapd.conf file.

Importing the IBM Connect:Direct MIB

To import the product MIB:

1. Transfer the product MIB, \$CD.SDGAMIB(DGAZMIB), to the appropriate machine. This is a text file. Transfer it as SDGAMIB.mib.
2. Use the UNIX process, xnmloadmib, to load the Connect:Direct for z/OS MIB into the MIB database.

Setting Up Tivoli NetView Rules

Set up a rule using nvrsEdit that forwards the IBM Connect:Direct trap messages to the Tivoli Enterprise Console. A sample rules file is provided in \$CD.SDGATRP(DGAZIVRS) that can be transferred as a text file to CDTrap_Tivoli.rs.

Processing the Boroc File

1. Transfer the \$CD.SDGATRP(DGATOROC) file to the appropriate machine in the appropriate directory (usually TEC_CLASSES). This is a text file. Transfer it as CDTrap_Tivoli.boroc.
2. Compile the rules using the wcomprules process.
3. Load the rulebase using the wloadrb process.
4. Recycle the TEC event server, wstopesvr and wstartesvr.

Customizing Tivoli NetView without the Tivoli Enterprise Console

Use the following information to customize Tivoli Netview for SNMP traps without the Tivoli Enterprise Console.

Importing the IBM Connect:Direct Trap Configuration File

Transfer and execute the CDTrap_NetView.sh script to add the Connect:Direct for z/OS trap in to the trapd.conf file.

1. Transfer the file, \$CD.SDGATRP(DGAZTRAP), to the appropriate computer. This is a text file. Transfer it as CDTrap_NetView.sh.
2. Run the CDTrap_NetView.sh script to add the traps to the trapd.conf file.

Importing the IBM Connect:Direct MIB

To import the product MIB:

1. Transfer the product MIB, \$CD.SDGAMIB(DGAZMIB), to the appropriate machine. This is a text file. Transfer it as SDGAMIB.mib.
2. Use the UNIX process, xnmloadmib, to load the Connect:Direct for z/OS MIB into the MIB database.

Disabling SNMP Traps

Connect:Direct for z/OS provides support for an SNMP agent to send SNMP traps to alert a network manager of certain events. An event is any IBM Connect:Direct message that is written to the console using SCWTO or DMWTO. Each event is triggered by the IBM Connect:Direct message ID and the trap text (short message text of that IBM Connect:Direct message). The IBM Connect:Direct events generated are defined by category and type.

Procedure

1. Edit the member DGAXSNMP in the \$CD.DGASAMP dataset and disable any SNMP trap that you do not want IBM Connect:Direct to trigger.
2. Specify the SNMP.DSN initialization parameter to have the SNMP traps specified in Step 1 disabled at initialization.

Note: Refer to the *IBM Connect:Direct for z/OS Administration Guide* for a description of each SNMP trap.

Configuring IBM Connect:Direct without SNA Support

About this task

To configure IBM Connect:Direct without SNA support:

Procedure

1. Define the following initialization parameters, which allow IBM Connect:Direct to function in a TCP-only environment:
 - SNA=NO
 - TCP=OES
 - TCP.LISTEN=((addr , port) , (addrn , portn))
 - TCP.API.LISTEN=((addr , port) , (addrn , portn))
 - MAXUSER=nnn

See Global Initialization Parameters for detailed information on these parameters.

When SNA=NO, you must specify the TCP parameters, and define a TCP port to accept API signons and commands. The MAXUSER parameter controls the number of concurrent API signons. The following screen shows an example:

```
SNA=NO
TCP=OES
TCP.LISTEN=(199.1.1.2,4199)
TCP.API.LISTEN=(199.1.1.2,4198)
MAXUSERS=10
```

This sample configuration allows IBM Connect:Direct to initialize without SNA, support only TCP connections, and authorize up to 10 concurrent API signons using TCP/IP.

2. Define the network map entries. See Maintaining the Network Map for information.

Define LOCAL node with the second positional parameter as NO-VTAM. The LOCAL node definition is used to define the SNA ACBNAME that IBM Connect:Direct opens in an SNA environment. When you specify NO-VTAM, IBM Connect:Direct does not attempt to open the VTAM ACB. See the example below.

Define the PNODE=SNODE adjacent node entry as TCP and specify a TCPAPI parameter. The PNODE=SNODE adjacent node entry is required to enable API signons. This node definition enables IBM Connect:Direct to run Processes that loop back to this primary node and allows for API signon without the API specifying the transport protocol. Use the LDNS parameter to assign the domain name or define the IP address within the adjacent node entry. The following screen shows an example:

```
LOCAL.NODE=(( CD.OS390 , NO-VTAM      , SUPERUSR) -
              TCQ=( CD.TCX      -
                    CD.TCQ ) )

/*
/* Pnode=Snode Adjacent node entry          */
/* Allows for Pnode=Snode processes and     */
/* allows for API signons thru TCP/IP       */
/*                                          */
ADJACENT.NODE=(( CD.OS390, 4199, 10.20.200.2, TCP) -
                PARSESS=(53 2) -
                TCPAPI=(4198,10.20.200.2) )

Or
ADJACENT.NODE=(( CD.OS390, 4199, , TCP) -
                PARSESS=(53 2) -
                LDNS=long.domain.name      -
                TCPAPI=(4198,) )
```

3. From \$CD.SDGAJCL, execute DGAJLOAD, the network map installation job, to load the network map.

If IBM Connect:Direct uses SNA, you can initialize IBM Connect:Direct even if SNA is not available. This task is accomplished by setting the SNA= initialization parameter to YES.

If the SNA= initialization parameter is YES and you try to start IBM Connect:Direct when SNA is not available, or if SNA becomes unavailable during a session, the system displays the following message:

```
VTAM or Connect:Direct inactive, Type CANCEL, CONTINUE, RETRY or NOVAM
```

The following table describes each option:

Option	Description
Cancel	IBM Connect:Direct does not initialize.
Continue	<p>IBM Connect:Direct continues initializing without SNA and keeps trying to establish the SNA session. IBM Connect:Direct sends a message to the operator every 20 minutes indicating that it is trying to start the SNA session, until the session is established.</p> <p>Any SNA Processes that are queued or submitted during this time period are placed in the Wait queue. These Processes can execute after the SNA session is established.</p>
Retry	IBM Connect:Direct tries to establish the SNA session before continuing initialization. After 20 attempts, it displays the "VTAM or IBM Connect:Direct inactive" message again. You can again choose one of the options in this table.
NoVTAM	IBM Connect:Direct continues initializing without SNA support and does not try to establish the SNA session. Any SNA Processes that are queued or submitted are placed in the Wait queue, where they remain until an SNA session is later established.

Overriding IBM Connect:Direct for z/OS Default Language Environment Run-Time Options

Language Environment run-time options can be overridden in Connect:Direct by adding a CEEOPTS DD to the Connect:Direct startup JCL. This DD should point to a dataset or PDS member. The CEEOPTS dataset or PDS must be RECFM=F or FB. Output will be directed to the CEEOUT DD if it is present. If there is no CEEOUT DD, it will be dynamically allocated to SYSOUT.

For example, to see the Language Environment run-time options report, specify:

```
RPTOPTS(ON)
```

To see that report and the Language Environment storage report, specify:

```
RPTOPTS(ON) ,RPTSTG(ON)
```

These Language Environment reports are generated at Connect:Direct SHUTDOWN time, but only if the Language Environment enclave ends "normally." Thus, you will not see them if you cancel your Connect:Direct job. An example of the use of the Language Environment storage report would be to tune the HEAP and HEAPPOOLS runtime options so that Language Environment will more efficiently allocate storage resources to multiple concurrent COPY tasks.

Note: For more information on Language Environment run-time options, refer to the Language Environment Customization guide for the current level of z/OS.

The following table lists the default settings of the Language Environment run-time options.

OPTION	DEFAULT VALUE
ABPERC	NONE
ABTERMENC	RETCODE
AIXBLD	OFF
ALL31	ON
ANYHEAP	1M, 4M, ANYWHERE, FREE
BELOWHEAP	32K, 32K, FREE
CBLOPTS	OFF
CBLPSHPOP	OFF
CBLQDA	OFF
CHECK	OFF
COUNTRY	US
DEBUG	OFF
DEPTHCONDLMT	10
ENVAR	'_CEE_ENVFILE=DD:ENVIRON'
ERRCOUNT	0
ERRUNIT	6
FILEHIST	OFF
HEAP	16M,16M, ANYWHERE, FREE, 4K, 4K

OPTION	DEFAULT VALUE
HEAPCHK	OFF, 1, 0
HEAPOOLS	ON, 48, 1, 88, 1, 416, 1, 1464, 1, 2200, 1, 9000, 1, 12496, 1, 16384, 1, 32768, 1, 34920, 1, 49256, 1, 64536, 28
INQPCOPN	ON
INTERRUPT	OFF
LIBSTACK	1K, 1K, FREE
MSGFILE	CEEOUT, FBA, 121, 0, ENQ
MSGQ	15
NATLANG	ENU
NOAUTOTASK	null
NOTEST	ALL, *, PROMPT, INSPREF
NOUSRHDLR	"
OCSTATUS	ON
PC	OFF
PLITASKCOUNT	20
POSIX	ON
PROFILE	OFF, "
PRTUNIT	6
PUNUNIT	7
RDRUNIT	5
RECPAD	OFF
RPTOPTS	OFF
RPTSTG	OFF
RTEREUS	OFF
SIMVRD	OFF
STACK	128K, 128K, ANYWHERE, KEEP, 512K, 128K
STORAGE	NONE, NONE, NONE, 8K
TERMTHDACT	UADUMP
THREADHEAP	4K, 4K, ANYWHERE, FREE
THREADSTACK	ON, 512K, 128K, ANYWHERE, KEEP, 512K, 128K
TRACE	OFF, 8192, DUMP, LE=3
TRAP	ON, NOSPIE
UPSI	00000000
VCTRSAVE	OFF
XPLINK	OFF
XUFLOW	AUTO

Note: If IBM Connect:Direct is configured to implement for support of UDT then XPLINK needs to specified as ON.

IBM Connect:Direct Enqueue Resource Management

This topic includes definitions of IBM Connect:Direct resources for a Multi-Image Manager (MIM) or Global Resource Serialization (GRS) system.

Major (Qname)	Minor (Rname)	Scope	Description
NDMGDG	GDG Base DSName	SYSTEMS	Issued during allocation of a GDG dataset when GDGENQ=YES.
NDMNTMAP	APPLID	SYSTEMS	Issued while processing the APPLID record for a signon. When the APPLID is found and opened successfully, the DEQ is issued. This ENQ could be held for awhile depending on the number of signons and the number of APPLIDs in the list.
NDMUPDNM	Netmap DSName	SYSTEMS	Issued during signon while processing the Netmap. Also, issued while processing a Netmap update and held for the entire update. To lock out all API while the Netmap is being updated, NDMUPDNM could be held awhile depending on the parameters in the update file.
NDMTCX	TCQ index DSName (TCX)	SYSTEMS	Issued from several places to lock out all API and other TCQ updates.
TCXESF	TCQ index DSName (TCX)	SYSTEMS	Issued during initialization to lock all ESF updates. Released after initialization of the TCQ/TCX is complete. Also issued during an ESF submit. Serialize processing between ESF submits and initialization.

Note: MIM now handles the cleanup of its QCB control blocks and the use of SCOPE=ALL is no longer valid. Use one of the following parameters appropriate for your system:

- Code SCOPE=RESERVE if single system.
- Code SCOPE=SYSTEMS if using shared DASD (for example, SDF).
- Code SCOPE=SYSTEM if not using shared DASD.
- When in doubt, code SCOPE=SYSTEMS.

Initialization Errors

This topic describes the causes of common errors, such as inadequate storage or errors in the statistics log, that may occur when you initialize Connect:Direct for z/OS.

Note: For all initialization errors related to Strong Password Encryption, see *Troubleshooting Possible SPE Problems* in the *IBM Connect:Direct Secure Plus for z/OS Implementation Guide*.

For initialization warnings related to certificate validation checks, see *Troubleshooting* in the *IBM Connect:Direct Secure Plus for z/OS Implementation Guide*.

In addition to the initialization errors described in this topic, you may see different error messages in traces, which are also related to initialization:

- If you receive a return code of 16, initialization terminates and error messages appear in the JES log. The messages inform you if one or more of the initialization parameters cannot co-exist with any of the

If you receive a return code of 4 when you stop IBM Connect:Direct, be sure to review the NDMLOG for SITA995I messages indicating obsolete parameters. Once you remove the parameters, these messages no longer appear in the NDMLOG trace and IBM Connect:Direct ends with a return code of 0. For more information on traces, see *Isolating Problems* in the *IBM Connect:Direct for z/OS Administration Guide*.

If you receive a return code of 8A in the RPLERRCK trace when you start IBM Connect:Direct and you have multiple TCP/IP stacks defined, the message, No such device or address (STCO999E), appears in the trace. If you have only one TCP/IP stack defined, this is normal and no cause for alarm. If you are in an environment where multiple TCP/IP stacks are defined, you will want to investigate the situation and make sure the correct TCP stack is specified.

Initialization parameters are set up during installation. When an error occurs, it may be necessary to alter these initialization parameters. For example, refresh the TCQ if Processes are hanging and the queue becomes corrupted.

Note: In a IBM Connect:Direct/Plex, only override initialization parameters allowed in the local initialization parameters file. Use the PARM= keyword in the EXEC statement at system startup.

In this example, the startup job is specifying a cold start of the IBM Connect:Direct TCQ (TCQ=COLD), overriding the TCQ= value specified in the initialization parameters file.

Error Messages					
SITA037I	SITA042I	SITA043I	SITA044I	SITA045I	SITA047I
SITA049I	SITA050I	SITA078I	SITA080I	SITA096I	SITA097I
SITA098I	SITA099I	SITA122I			

Use the following table to troubleshoot the storage problem.

Cause	Action	Data to Collect
Inadequate address space is allocated to the IBM Connect:Direct DTF region.	Review the short and long text messages. Check the IBM Connect:Direct startup job stream and the amount of storage allocated to the IBM Connect:Direct region by checking the region parameter on the job card. Use REGION=0M or as advised by Customer Support.	<ul style="list-style-type: none"> IBM Connect:Direct error message Region/partition size

Condition: Server Initialization Error (SXTA101I) with ABEND U1024

The CDPLEX.MAXSERVER value has been exceeded by Server A.

Error	Cause	Action
ABEND U1024	The IBM Connect:Direct/Plex Server A (where A is the name of a particular server in a IBM Connect:Direct/Plex environment) has been rejected by the IBM Connect:Direct Manager. Server A terminated with a User ABEND code of U1024.	Correct the CDPLEX.MAXSERVER global initialization parameter by increasing the value to cover the number of servers in the IBM Connect:Direct/Plex environment, and restart the server.

Condition: ABEND U0044

Use the following table to troubleshoot this abnormal end condition (ABEND).

Error	Cause	Action	Collect
ABEND U0044	GETMAIN failed	<ol style="list-style-type: none"> JCL set REGION=0M Initialization parameters set MEMLIMIT=2M Initialization parameters set TRACE.BUFFER=0 	If there is still an issue after taking the recommended actions, collect the SVCDUMP and job log.

Condition: Error While Allocating or Opening Initialization Parameters File

An error occurs while allocating or opening the initialization parameters file.

Error Messages					
SITA063I	SITA070I	SITA121I	SITA123I	SITA291I	SITA292I
SITA293I	SITA294I	SITA501I	SITA502I	SITA505I	

Cause	Action	Data to Collect
IBM Connect:Direct initialization parameters file does not exist, or the parameter on the EXEC statement is specified incorrectly.	Review the short and long text messages. Look at the IBM Connect:Direct startup JCL for the EXEC statement. Ensure that the specified initialization parameters dataset exists and is correct.	<ul style="list-style-type: none"> IBM Connect:Direct startup JCL DMGSCMAP STARTUP file

Condition: Initialization Parameter Errors

IBM Connect:Direct does not initialize because of initialization parameter errors.

Error Messages					
SITA003I	SITA014I	SITA018I	SITA021I	SITA038I	SITA039I
SITA040I	SITA041I	SITA063I	SITA071I	SITA072I	SITA073I
SITA079I	SITA082I	SITA083I	SITA084I	SITA085I	SITA086I
SITA087I	SITA088I	SITA089I	SITA091I	SITA092I	SITA093I
SITA094I	SITA095I	SITA104I	SITA105I	SITA106I	SITA148I
SITA201I	SITA202I	SITA213I	SITA214I	SITA215I	SITA216I
SITA217I	SITA220I	SITA221I	SITA222I	SITA223I	SITA224I
SITA225I	SITA226I	SITA227I	SITA228I	SITA230I	SITA231I
SITA232I	SITA233I	SITA234I	SITA242I	SITA250I	SITA251I
SITA252I	SITA253I	SITA254I	SITA260I	SITA261I	SITA262I
SITA263I	SITA270I	SITA271I	SITA272I	SITA273I	SITA280I
SITA281I	SITA282I	SITA283I	SITA285I	SITA286I	SITA287I
SITA300I	SITA506I	SITA507I	SITA508I	SITA509I	SITA510I
SITA511I	SITA512I	SITA513I	SITA514I	SITA515I	SITA516I
SITA517I	SITA518I	SITA540I			

Use the following table to troubleshoot the initialization parameters problem.

Cause	Action	Data to Collect
Syntax errors or errors in the values specified exist for one or more initialization parameters.	Review the short and long text messages. Note the messages that you see on the console during initialization to determine the reason for the error. Verify that any comments preceding parameters are followed by “*/”. Refer to the appropriate IBM Connect:Direct installation guide for the valid parameter values. Correct the parameter in the initialization parameters file, and restart IBM Connect:Direct.	<ul style="list-style-type: none"> IBM Connect:Direct error message Initialization parameters file

Condition: IBM Connect:Direct VSAM File Error

IBM Connect:Direct does not initialize because of an error with one of the IBM Connect:Direct VSAM files: the message file, authorization file, type file, network map file, statistics file, checkpoint file, or TCQ file.

Error Messages					
SITA055I	SITA056I	SITA057I	SITA059I	SITA060I	SITA061I
SITA062I	SITA064I	SITA065I	SITA066I	SITA075I	SITA076I
SITA077I	SITA100I	SITA101I	SITA102I	SITA103I	SITA110I
SITA111I	SITA112I	SITA113I	SITA130I	SITA131I	SITA132I
SITA133I	SITA160I				

Use the following table to troubleshoot the Connect:Direct for z/OS VSAM file error.

Cause	Action	Data to Collect
An error occurred while opening the file, or VSAM encountered a physical or logical error. The VSAM file cannot be defined and loaded correctly, the name of the file might be specified incorrectly, or the file might be corrupted.	Review the short and long text messages. Verify that the name of the file is specified correctly in the initialization parameters file. Also, verify that the file is a correctly defined VSAM dataset and loaded. Refer to VSAM Files DASD Requirement and Description and Define the IBM Connect:Direct VSAM Files .	<ul style="list-style-type: none">IBM Connect:Direct messagesAny messages issued by the VSAM AMS Interface Program (IDCAMS) when the IBM Connect:Direct VSAM file was defined and loaded

Condition: VTAM Initialization Errors

IBM Connect:Direct does not initialize because of errors with VTAM initialization.

Error Messages					
SITA004I	SITA005I	SITA048I	SITA051I	SITA052I	SITA053I
SITA054I	SITA090I	user ABEND U0075			

Use the following table to troubleshoot the VTAM initialization error.

Cause	Action	Data to Collect
The IBM Connect:Direct DTF APPLID is defined incorrectly, the APPLID logmode table is incorrect or is not found, the DTF APPLID has a VTAM password associated with it, or the IBM Connect:Direct DTF APPLID is not active.	Review the short and long text messages. Ensure that the DTF APPLID is active. Ensure that the APPLID specified in the LOCAL.NODE entry of the network map is correctly specified and is defined to VTAM. Check the ESTAE output for additional error information. If a VTAM password is associated with the IBM Connect:Direct APPLID, remove it, and do not refer to it in the network map.	<ul style="list-style-type: none">ESTAE outputIBM Connect:Direct network mapAPPLID definitionLogmode table entry

Condition: Invalid Security Environment (SITA997I)

Use the following table to troubleshoot the security environment error.

Cause	Action	Data to Collect
The thread-level security is not valid between IBM Connect:Direct and z/OS UNIX System Services.	If your installation implemented Program Access to Datasets (PADS) functionality within your security system (such as RACF, TSS, ACF2), include the IBM Connect:Direct datasets in your Program Control List (PCL).	<ul style="list-style-type: none"> SYSLOG from IBM Connect:Direct startup Security system profiles for the resources denied in the SYSLOG

Condition: Statistics Log Error

IBM Connect:Direct does not initialize because of an error in the statistics log.

Error Messages					
SSTI004I	SSTI005I	SSTI006I	SSTI007I	SSTI008I	SSTI009I
SSTI010I	SSTI011I	SSTI012I	SSTI013I	SSTI014I	SSTI015I
SSTI016I	SSTI017I	SSTI018I	SSTI019I	SSTI020I	SSTI021I
SSTI022I					

SITA059I and user ABEND U0070 follow one of the previous error messages. The cause and necessary action to take depend on the SSTIxxxI message as explained in the following sections.

SSTI004I

File pair ESDS dataset has invalid CI SIZE.

Cause	Action
The ESDS CI SIZE is less than 2048. The minimum acceptable CI SIZE for this dataset is 2048.	Review short and long text messages. Delete and redefine the file pair ESDS dataset and ensure a CI SIZE that is greater than or equal to 2048.

SSTI005I

Invalid KSDS dataset type in file pair.

Cause	Action
Review both the short text and long text IBM Connect:Direct messages. While verifying the dataset, it was found that the dataset type was not KSDS.	Verify that the statistics file pair KSDS dataset is correctly allocated. If no allocation is correct, delete and redefine the KSDS dataset.

SSTI006I

Invalid KSDS dataset key length.

Cause	Action
The key length of the file pair KSDS dataset is invalid.	Review short and long text messages. Refer to the platform-specific IBM Connect:Direct installation and administration guide for the correct length.

SSTI007I

Invalid file pair KSDS dataset key offset.

Cause	Action
The key offset for the KSDS dataset is not 0.	Review short and long text IBM Connect:Direct messages. Delete and redefine the KSDS with the correct key offset.

SSTI008I

File pair has empty ESDS but non-empty KSDS.

Cause	Action
The ESDS dataset is empty, but the KSDS dataset has data in it. One possible reason is that the ESDS dataset was reset, but the KSDS was not.	Review both short and long text messages. The ESDS and KSDS datasets must both be empty or must both contain data. Either delete and redefine the KSDS dataset, restore the ESDS dataset, or specify STAT.INIT=COLD. IBM Connect:Direct opens both datasets with RESET at initialization time.

SSTI009I

File pair has empty KSDS but non-empty ESDS.

Cause	Action
The KSDS dataset is empty, but the ESDS dataset has data in it. One possible reason is that the KSDS dataset was reset, but the ESDS was not.	Review both short and long text messages. Both KSDS and ESDS datasets must be empty or contain data. Either delete and redefine the ESDS dataset, restore the KSDS dataset, run batch utility DMSTBKEY to rebuild the KSDS dataset, or specify STAT.INIT=COLD. IBM Connect:Direct then opens both datasets with RESET at initialization time.

SSTI010I

Error reading the file pair KSDS control record.

Cause	Action
While attempting to read the KSDS control record, an error code was returned.	Review both short and long text messages. Verify that the KSDS dataset is correctly defined. Register 15 contains the VSAM GET return code. To reuse the dataset, specify STAT.INIT=COLD in the initialization parameters and restart IBM Connect:Direct.

SSTI011I

KSDS control record ESDS name does not match.

Cause	Action
The KSDS control record contains the name of the matching or paired ESDS dataset. When the KSDS and ESDS dataset names are built by IBM Connect:Direct at initialization time based on STAT.DSN.BASE, the ESDS dataset name that the system built did not match the ESDS dataset name in the KSDS control record.	Review both the short text and long text IBM Connect:Direct messages. Do one of the following: <ul style="list-style-type: none"> Run batch utility DMSTBKEY to rebuild the KSDS dataset control record. Delete and redefine both KSDS and ESDS datasets. Specify STAT.INIT=COLD (Connect:Direct opens all file pairs with RESET.).

SSTI012I

KSDS control record ESDS CI SIZE do not match.

Cause	Action
The KSDS control record contains the CI SIZE of the matching or paired ESDS dataset. The CI SIZE from the KSDS control record is not equal to the CI SIZE of the paired ESDS dataset that is opened.	Review the short and long text messages. Verify that the statistics file pair is identified with the STAT.DSN.BASE and STAT.FILE.PAIRS initialization parameters. If the CI SIZE of the ESDS file has changed, you must run the DMSTBKEY batch utility to rebuild the information in the key-sequenced cluster.

SSTI013I

Error reading the ESDS control record.

Cause	Action
While attempting to read the ESDS control record, an error code was returned.	Review both the short and long text messages. Verify that the ESDS dataset is correctly defined. Register 15 contains the VSAM GET return code. To reuse the dataset, specify STAT.INIT=COLD in the initialization parameters and restart IBM Connect:Direct.

SSTI014I

Invalid ESDS control record.

Cause	Action
IBM Connect:Direct read the first record in the ESDS dataset; however, it was not the control record. Either the file is not a IBM Connect:Direct statistics file, or the file was corrupted after being written.	Review both the short text and long text messages. Verify that the statistics file pair is correctly defined using the STAT.DSN.BASE and STAT.FILE.PAIRS initialization parameters. If the file was corrupted and it is the KSDS cluster of the pair, then the information can be rebuilt by running the DMSTBKEY batch utility for the file pair. If the file was corrupted and it is the ESDS cluster of the pair, then the statistics information in the file pair is lost. In this case, both files of the pair must be empty before IBM Connect:Direct initializes successfully. Specify STAT.INIT=COLD in the initialization parameters to initialize successfully.

SSTI015I

Invalid file pair sequence.

Cause	Action
The files that comprise the statistics file pair list are not arranged in chronological order. IBM Connect:Direct requires that all non-empty statistics file pairs be ordered chronologically based on the date and time of the oldest data in each pair. Note:	Review the short and long text IBM Connect:Direct messages. Verify that the statistics file pair list is correctly defined using the STAT.DSN.BASE and STAT.FILE.PAIRS initialization parameters. If the reason for the problem is not apparent, it might be necessary to archive or copy all ESDS file pairs to preserve the current statistics data. Then delete and redefine all statistics file pair clusters.

Note: Although IBM Connect:Direct processes the statistics file pair list in a circular or wrap around method, the first file pair in the list does not always contain the oldest data.

IBM Connect:Direct verifies the order using the following steps:

1. IBM Connect:Direct locates the file pair containing the oldest data.
2. From that point, IBM Connect:Direct examines each non-empty file pair in the list. If the oldest data is not located in the first file pair, when the search reaches the last file pair in the list, it wraps to the beginning of the list. Each successive non-empty file pair must have a later date than the previous pair.

IBM Connect:Direct always maintains statistics records in strict chronological order. If the records are not in chronological order at initialization, IBM Connect:Direct assumes that the list or the files are incorrectly altered since IBM Connect:Direct last wrote them.

SSTI016I

WARNING: KSDS of file pair may be too small.

Cause	Action
The size of the key-sequenced cluster of the file pair named is less than 10% of the size of the associated entry sequenced cluster. This condition can lead to premature filling of the key-sequenced cluster and wasted space in the entry-sequenced cluster.	Review the short and long text IBM Connect:Direct messages. Examine the SSTS005I message issued when the file pair fills to determine the amount of the file IBM Connect:Direct was able to use. Set the key-sequenced cluster to 15% of the size of the entry-sequenced cluster.

SSTI017I

WARNING: KSDS of file pair may be too large.

Cause	Action
The size of the key-sequenced cluster of the file pair named is more than 30% of the size of the associated entry-sequenced cluster. This condition can lead to premature filling of the entry-sequenced cluster and wasted space in the key-sequenced cluster.	Review both the short text and long text IBM Connect:Direct messages. Examine the SSTS005I message issued when the file pair fills to determine the amount of the file IBM Connect:Direct was able to use. Set the key-sequenced cluster to 15% of the size of the entry-sequenced cluster.

SSTI018I

File pair WHERE condition found.

Cause	Action
During statistics initialization, a warning may have been issued specifying the KSDS dataset is either too small or too large. This message contains the FILE PAIR where this situation occurred	None

SSTI019I

No active file pair found.

Cause	Action
Statistics initialization could not determine which of the statistics file pairs was the active pair. The ESDS of each pair contains a status flag that indicates which pair is active. The active flag was not found in any of the ESDS clusters.	Review both the short text and long text IBM Connect:Direct messages. Archive or copy all ESDS file pairs to preserve the current statistics data. Then delete and redefine all statistics file pair clusters.

SSTI020I

More than one file pair active.

Cause	Action
For any given sequence of statistics file pairs, only one set can be active at a time. During IBM Connect:Direct statistics initialization, more than one set was found to be active. This condition occurs when mixing old file pairs with current ones	Review both the short text and long text IBM Connect:Direct messages. Empty the files by deleting and redefining the file pairs. None of the file pairs are active. Another alternative is to set STAT.INIT=COLD forcing IBM Connect:Direct to open each set of file pairs with RESET at initialization time. If you are interested in retaining the data in the files, archive them first, then continue with one of the previous two methods.

SSTI021I

More than one file pair is both not complete and not empty.

Cause	Action
The mixing of old file pair datasets with current file pairs can cause this result. Another cause is incorrect modification of file pairs between executions of IBM Connect:Direct.	Review both the short text and long text IBM Connect:Direct messages. Verify that the statistics file pair list is correctly defined using the STAT.DSN.BASE and STAT.FILE.PAIRS initialization parameters, and restart IBM Connect:Direct. You can also use the STAT.INIT=COLD startup parameter.

SSTI022I

Invalid ESDS dataset type in file pair.

Cause	Action
While verifying the ESDS dataset, it was determined that the dataset type was not ESDS	Review both the short text and long text IBM Connect:Direct messages. Verify that the statistics file pair ESDS dataset is correctly allocated. If the dataset is not correctly allocated, delete and redefine the ESDS dataset.

SSTI028I

Cause	Action
A KSDS is not at the current level. IBM Connect:Direct was started with Stat.INIT=WARM and is using a Stat File Pair that was last reset by a different level of IBM Connect:Direct.	See accompanying message SSTI029I .

SSTI029I

Cause	Action
See accompanying message SSTI028I .	<p>SELECT STATISTICS search by SNODE results are unpredictable until one of the following options is performed:</p> <ol style="list-style-type: none"> 1. Archive all inactive ESDSs reported by SSTI028I containing records you wish to keep. Then Switch Stat File Pairs. Lastly, archive the newly inactive Stat File Pair ESDS. 2. Shut down CD and run DMSTBKEY on all Stat File Pairs reported by SSTI028I. Then start CD with STAT.INIT=WARM. 3. Shut down CD and archive all pairs reported by SSTI028I. Then restart CD with STAT.INIT=COLD.

Condition: TCQ.THRESHOLD Parameter Error (SITA186I)

Use the following table to troubleshoot when the TCQ.THRESHOLD parameter has not been properly defined.

Cause	Action	Data to Collect
If you receive these messages during initialization, the TCQ.THRESHOLD parameter is not specified.	Specify the TCQ.THRESHOLD parameter as YES, NO, or a defined percentage value in the range 0-99.	None

Chapter 5. CICS Administration and User Guide

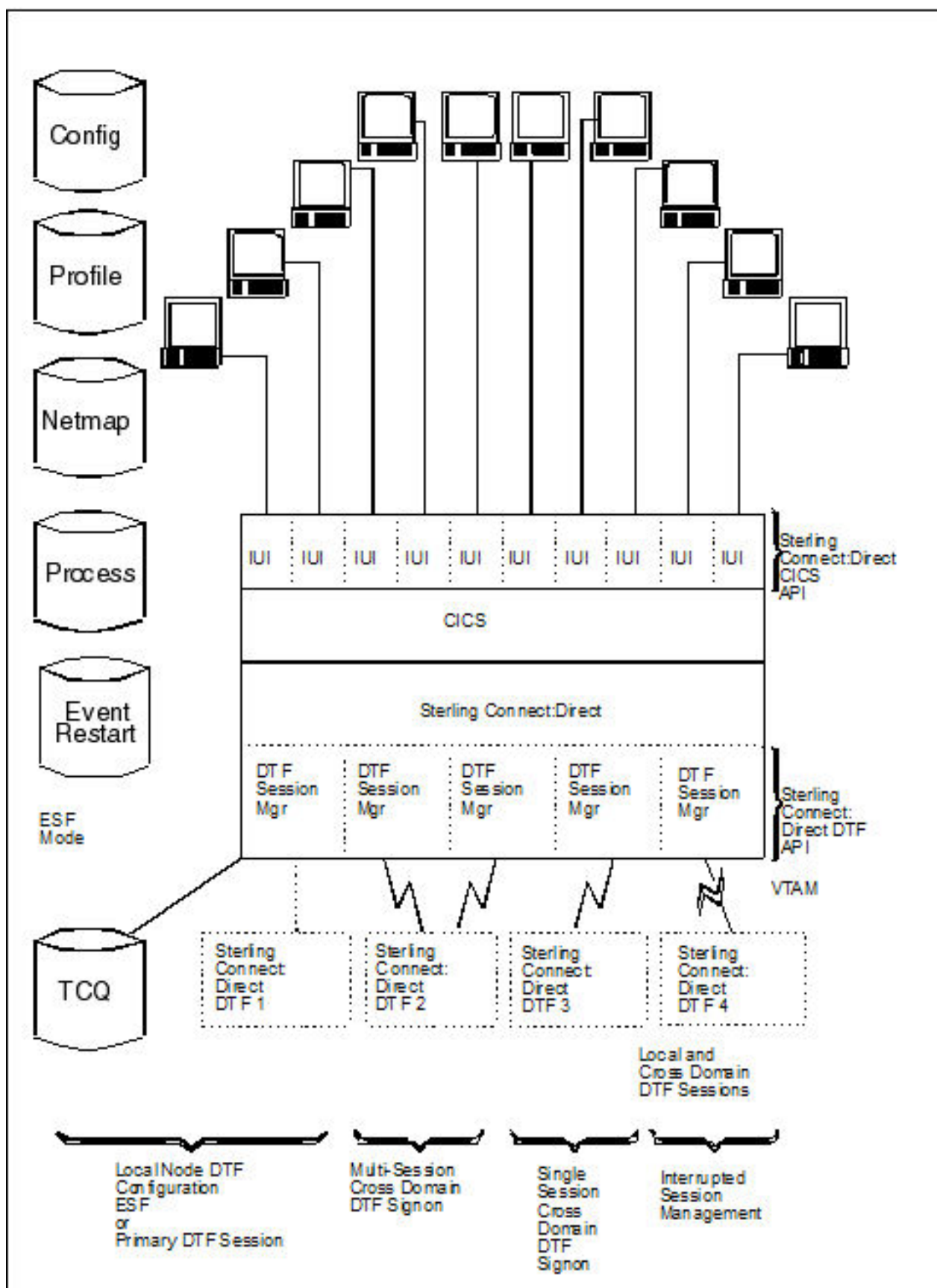
CICS Interface

The Customer Information Control System (CICS®) Interface provides a number of components that enable users and applications to access, control, and transfer data across networks. The major components:

Component	Description
Interactive User Interface (IUI)	Enables users to transfer files, initiate applications, and monitor activity in a user-friendly environment.
Application Program Interface (API)	Enables the IUI and CICS Administration to communicate to the Data Transmission Facility (DTF) through the session manager. The API interprets the commands, but it is the responsibility of the session manager to establish communication sessions and perform standard session management functions.
Data Transmission Facility (DTF)	Controls information distribution to other nodes in the network. In an IBM IBM Connect:Direct/Plex environment, the IBM Connect:Direct/Manager and IBM Connect:Direct/Servers form the DTF.
Extended Submit Feature (ESF) Mode	Enables users to submit data for transmission even if the DTF is not active. IBM Connect:Direct requests sent to an inactive node are routed to the sending Transmission Control Queue (TCQ) until the inactive node is initiated.

Components of the CICS Interface

The IBM Connect:Direct for z/OS installation offers an optional selection to use the CICS interface. The following figure illustrates a CICS implementation.



CICS Screen Conventions

This section describes the conventions for the CICS Interface. The screens are based on the standard IBM 3270 Model 2 display. Use a terminal or terminal emulator that can handle the 3270 Model 2 mode.

The CICS Interface has the following screen conventions to help you input data and navigate within the system:

- All variable data is highlighted.
- Variable fields contain underscore characters to indicate the size of each field.
- Required fields are highlighted and optional fields are displayed at normal intensity.
- Each screen in the User System contains information fields of various widths and attributes for user data and for system-displayed data.

System Fields

As you perform your user activities, the CICS Interface displays several system fields with current information about your data, updates of system information, and responses to your Processes. The system fields include the following:

Field	Description
DATE	The 8-character DATE field contains the system date expressed as month, day, and year (MM/DD/YY) and occurs in the upper right corner of the screen.
TIME	The 8-character TIME field contains the system time expressed as hours, minutes, and seconds (HH:MM:SS) and occurs in the upper right corner of the screen.
ESF MODE	The 8-character ESF MODE field occurs in the upper right corner of the screen, and contains the ESF MODE flag in the event that the DTF is not working, and you are working under ESF. If the field contains the flag ESF MODE, you are working under ESF; if the field is blank, you are working with the DTF. A blank field indicates normal operation.
NODE	The 16-character NODE field, displayed in the upper right corner of the screen, contains the name of your connected node.
MESSAGE	<p>The 75-character MESSAGE field occurs near the bottom of the screen. It contains the system messages associated with the success or failure of your CICS Interface activities. CICS Interface messages that can occur during normal operations are listed in Product Messages.</p> <p>Note: Some messages issued by the CICS Interface are actually Connect:Direct for z/OS messages (inquire with your administrator about the IBM Connect:Direct message file, or select option MD from the PRIMARY MENU to exercise the MESSAGE DISPLAY screen features, or select PF2 to view the LAST MESSAGE screen).</p>

PF Keys

For those screens that have programmable function (PF) keys defined, you can select standard PF keys to assist you in the performance of CICS Interface activities, while you are at the current screen. You cannot redefine PF keys.

The following table describes the function of each PF key.

Keys	Function
PF3	Exits the current screen, and takes you to the previous screen
PF5	Adds an entry to the list on the screen

Keys	Function
PF6	Deletes an entry from the list on the screen
PF7	Scrolls backward through the list of available data on the screen
PF8	Scrolls forward through the list of available data on the screen
PF9	Applies updates to screen information
PF10	Scrolls to the left on the screen
PF11	Scrolls to the right on the screen
ENTER	Refreshes screen data or processes a line command
CLEAR	Resets the data on the screen to default values

About the IBM Connect:Direct for z/OS CICS Documentation

The documentation related to an IBM Connect:Direct for z/OS installation is for programmers and network operations staff who use the CICS interface to maintain Connect:Direct for z/OS. This This documentation assumes knowledge of the IBM z/OS operating system and CICS/TS. If you are not familiar with the IBM z/OS operating system or CICS/TS, refer to the IBM library of manuals.

Configuring the CICS Interface

Activate the CICS Component - Optional

Activation of the CICS Component of Connect:Direct for z/OS requires additional items to be updated. An overview is presented followed by a task checklist.

CICS Customization Overview

To run the CICS Component, IBM Connect:Direct contains code that runs as an exit in CICS/TS to handle the data movement between CICS and IBM Connect:Direct. To perform the install, several transactions must be defined. These transactions are in the RDO source provided. The default transaction codes start with "DGA".

The interface between CICS and IBM Connect:Direct can start up during the CICS start, or it may be manually activated after CICS is running. If you want the interface to start up automatically, you must modify the CICS PLTPI. If you want the interface to shut down automatically, you must customize the CICS PLTSD.

If you want to process Event Data within CICS, you must customize and install DGAQ247. If you need to customize DGAQ249 and DGAQM98, search for Using ESS with the CICS API. If you do not want to use them, you can remove these entries from DGACCSO.

DGAQ249 reads Event Records out of the TDQ. If you need to add logic to DGAQ249 to do more than just read some records, you must customize it. Otherwise, you can remove its entries from DGACCSO.

You must define a few files, including the CONFIG file, which is handled with a REXX exec.

The CICS component has a set of transaction codes that you can accept as they are, or you may set your own. The CICS component uses the following default transaction codes:

Transaction Code	Description
DGA	User Transaction
DGAA	Administrator Transaction
DGAE	Event Services

Transaction Code	Description
DGAI	Initial Startup
DGAM	API Monitor
DGAN	User ESS Process Interface
DGAP	Printer Driver Transaction
DGAT	User Read/Process ESS

CICS Customization Checklist

- Customize the RDO source.

The members to be customized are located in \$CD.SDGACNTL.

Member	Description
DGACCSD	RDO source containing the Transactions, Program, File, Mapset, TDQ, and List definitions
DGACBCKO	RDO source to remove prior release(s)
DGACBACK	RDO source to back out the 5.1 release

- If you plan to migrate to this release from a prior release, review the *IBM Connect:Direct for z/OS Release Notes*® for migration instructions. See [Build the CICS Configuration File through ISPF](#) for the panels that the REXX will display.

Build the CICS Configuration File through ISPF

To install the CICS feature, complete the following CICS panels:

About this task

- Configuration File - Control Parameter Record Creation
- CICS DTF Record Creation
- Configuration File - Network Record Creation

For detailed information on the field descriptions for this series of CICS-related panels, search for the record type, namely, control, DTF node, or network records. You can also use the CICS Administration System later to modify the configuration information entered during the installation process.

You can use the following PF keys on the CICS screens.

PF Key	Function
Enter	Continue with the installation.
PF3	Return to the previous menu.
PF5	Terminate the installation.

Procedure

1. Execute the CONFIG file REXX from a TSO/ISPF session, using option 6 from the ISPF Command Shell. (\$CD is the High Level Qualifier from the SMP/E install.)

```
EX '$CD.SDGAISPC(DGA#CONF)' '$CD'
```

2. Define the appropriate fields in the CICS Feature CONFIGURATION FILE - CONTROL PARAMETER RECORD CREATION panel and press **Enter**.

```

-----      Connect:Direct CICS Feature      -----  DATE-yyyy/mm/dd
CONFIGURATION FILE - CONTROL PARAMETER RECORD CREATION  TIME=hh:mm

CMD ==>
THIS PANEL IS USED TO GENERATE THE CONTROL.PARMS INFORMATION FOR USE
BY Connect:Direct CICS.
AUTO.SIGNON                      Y                      Y OR N
SIGNON.REENTRY                   N                      Y OR N
CONNECT:DIRECT.EQ.CICSID         Y                      Y OR N
SKIP.SIGNON.PANEL                N                      Y OR N
CICS.TRANSACTION.CODE (MONITOR)  DGAM
CICS.TRANSACTION.CODE (STARTUP) DGAI
CICS.TRANSACTION.CODE (PRINT)   DGAP
CICS.TRANSACTION.CODE (ESO)     DGAE
CST.RETRY.INTERVAL               000500                HHMMSS
SESSION.RETRY.INTERVAL           0100                  MMSS
ESF.RETRY.INTERVAL               001500                HHMMSS
WORK.RETRY.INTERVAL              0015                  MMSS
MONITOR.INTERVAL                 30                     SS
INACTIVE.INTERVAL                003000                HHMMSS
MAX.SIGNON                       0100
MAX.TASKS                        02                     01-99
STORAGE.SUBPOOL                  127                   002-127
MENU OPTIONS: CF Y SB Y SS Y SP Y SD Y SN Y MD Y          Y OR N

```

3. Define the appropriate fields in the CICS Feature - CICS DTF RECORD CREATION menu and press **Enter**. You must enter a DTF NODE NAME.

```

-----      Connect:Direct CICS Feature      -----  DATE-yyyy/mm/dd
CICS DTF RECORD CREATION  TIME=hh:mm

CMD ==>

THIS PANEL ALLOWS YOU TO GENERATE A CONNECT:DIRECT CICS IUI.NODE
RECORD TO BE USED TO INITIALLY LOAD THE CONFIGURATION FILE.
THIS RECORD CAN BE UPDATED ONLINE USING THE 'DGAA' TRANSACTION.

PARAMETER                      VALUE
-----
DTF NODE NAME                  CD.DEV-----
NETMAP DDNAME                  NETFINP-----
DUMMY ID FOR DTF SIGNON        CICSUSER
SUPPRESS CONNECTION AT STARTUP Y                      Y OR N
ESF SIGNON ALLOWED              Y                      Y OR N
MAXIMUM WORKER SUBTASKS        04
ENTRIES IN WORK QUEUE           050
OUTPUT RECORD LIMIT             01000
SLOW RESPONSE NOTIFICATION      0030                MMSS

```

4. Define the appropriate fields in the CICS Feature CONFIGURATION FILE - NETWORK RECORD CREATION menu and press Enter. You must enter information for at least one network node record.

Customize and Install the IBM Connect:Direct CICS Resource Definition Source

All program, transaction, file, and associated definitions are provided in \$CD.SDGACNTL(DGACCSD). DGACCSD contains Resource Definition source code, which you may need to customize to meet your site's needs.

About this task



CAUTION: If the CICS option was installed in a release prior to IBM Connect:Direct for z/OS version 5.1, you must back out of the previous Resource Definition CSD source code by using the JCL in \$CD.SDGAJCL(DGAJCSL) prior to running the \$CD.SDGAJCL(DGAJCSL) job referred to in the following procedure.

To customize and install the IBM Connect:Direct CICS resource definition source:

Procedure

1. Review the source (DGACCSD). Use the GROUP and LIST names provided or change them in order to have unique names, if necessary.
2. Update the FILE definitions if any files have different High Level Qualifiers than what were specified in the ISPF panels.

For each IBM Connect:Direct DTF that your CICS system will have an IUI with, you must have a File definition statement that points to the network map used by that IBM Connect:Direct DTF. To tell the IUI what FILE definition CICS is using for the network map, the FILE name (DDName) must match the NETMAP DDName entered in the DTF NODE CONFIGURATION record for that IBM Connect:Direct DTF.

3. (Optional step) If you want to initialize IBM Connect:Direct during CICS startup or add IBM Connect:Direct CICS interface shutdown to the CICS shutdown, review \$CD.SDGASAMP(DGAXPLT). This member contains macro source and instructions for the CICS systems programmer to build or update PLTPI and PLTSD entries.

Note: To make the IUI available immediately upon CICS startup, you must use the LIST name in the SIP GRPLIST (keyword) parameter list.

4. Verify that the JOB and PARM information is correct in \$CD.SDGAJCL(DGAJCSL).
5. Shut down all CICS transaction servers.
6. Run the \$CD.SDGAJCL(DGAJCSL) job.

Modify Your CICS Startup

To modify your CICS startup:

Procedure

1. Review the \$CD.DGASAMP(DGAXCJCL) member for a list of DD statements and their purposes. Copy the applicable DD statements from DGAXCJCL to your CICS startup JCL (PROC or JOB).

Note: Any DD that is normally controlled by RDO should not be replicated in the JCL.

2. Modify any of the copied DD statements to use the correct High Level Qualifiers for dataset names as needed.
3. For any additional customization that may be required, see *Using ESS with the CICS API* in the *IBM Connect:Direct for z/OS Facilities Guide*.
4. Submit your CICS JCL for execution.

Starting the Connect:Direct for z/OS CICS Interface

About this task

Perform this task only if IBM Connect:Direct CICS is not part of the Program Load Table (PLT) initialization. After starting CICS, start the IBM Connect:Direct CICS interface as follows:

Procedure

1. Type the DGAA transaction code, and press Enter.
2. Select option I from the PRIMARY MENU and press Enter to go to the INTERFACE screen.
3. From the INTERFACE screen, select option A, and press Enter. When the message INTERFACE HAS BEEN STARTED is displayed, press Enter to refresh the status information, and wait until ACTIVE is displayed in the INTERFACE STATUS field, and a transaction number is displayed in the MONITOR TASK NUMBER field.
4. Press PF3 to return to the PRIMARY MENU, select option N on the PRIMARY MENU, and press Enter to go to the NODE STATUS screen.
5. On the NODE STATUS screen, for each node you want to activate, select the line command **A** in the field to the far left of each NODE STATUS line. When the date and time are displayed under the SESSION DATE/TIME heading, the node is activated.

Note: The node may already be active if the message SUPPRESS CONNECTION AT STARTUP = N is displayed on the DTF NODE screen during the installation procedures.

6. Press PF3 several times to exit the DGAA transaction, and then sign on to IBM Connect:Direct using the DGA transaction code.

Tune the CICS Interface

Tune IBM Connect:Direct CICS interface to enhance performance. Connect:Direct for z/OS CICS interface exits are not "threadsafe" and may cause performance issues if this application is not isolated to a non-threadsafe AOR.

Actual resource usage varies according to the implemented configuration, including the number of DTF nodes that can be signed on to by the CICS interface, the number of subtasks defined per node, and other considerations, such as the types of commands executed.

You may want to tune your IBM Connect:Direct system, based upon guidelines provided to assist you in estimating the impact of Connect:Direct for z/OS on your CICS online environment for the file I/O buffers, auxiliary or main temporary storage, and transaction priorities and class assignments

File I/O Buffers

You can reduce file I/O buffers to correspondingly lighten the load on your virtual and real storage, but at the expense of increasing disk I/O activity. The configuration and user profile datasets are type VSAM and can participate in Local Shared Resource (LSR) buffer pools, designated for light-to-medium usage datasets.

Auxiliary or Main Temporary Storage

You can choose main or auxiliary temporary storage as a trade off between using more real storage (main) and increasing disk I/O activity (auxiliary). The CICS interface makes heavy use of temporary storage, especially during SELECT PROCESS and SELECT STATISTICS operations.

CICS Dynamic Storage Area (DSA) Usage

The CICS interface uses the CICS DSA as follows:

- Approximately 1 KB is used by the CICS Task Control global exit.

- Approximately 10–20 KB is used for transaction-related storage and temporary storage records by the DGA and DGAA transactions.

In addition, the SELECT PROCESS (SP) and SELECT STATISTICS (SS) functions are heavy users of CICS Temporary Storage. Results returned by these functions are written to Temporary Storage as a series of 300-byte records.

Parameters are provided on the DTF NODE configuration screen of the DGAA transaction (OUTPUT RECORD LIMIT) and on the IUI.NODE statement of the IBM Connect:Direct CICS Configuration Load program (TDLIMIT) to limit the amount of data returned by these commands.

Above-the-Line Storage

Non-CICS storage above the 16-megabyte line is used by the CICS interface as follows:

- Approximately 340 KB is used for IBM Connect:Direct API programs.
- The CICS interface signon table, node table, and subtask tables are allocated above the line, with the total amount required calculated as follows.

$$36 + (18 \times (T+W)) + (304 \times S) + (224 \times N) + (144 \times T)$$

The following table describes the variables in the preceding example:

Variable	Description
T	The sum of the maximum subtask per node values or all nodes eligible to be signed on to by the CICS interface (MAXIMUM WORKER SUBTASKS on the DTF NODE configuration screen or VTAM®.SESSIONS on the IUI.NODE parameter of the CICS interface Configuration Load program).
W	The total number of entries in the WORK QUEUE for all DTF nodes.
S	The value of the MAX.SIGNON parameter of the Configuration Parameters screen or CONTROL.PARMS statement of the Configuration Load program.
N	The number of nodes eligible to be signed on to by the CICS interface (the number of DTF.NODE records defined in the CICS interface configuration file).

Transaction Priorities and Class Assignments

If you find that heavy usage of the CICS interface is causing resource shortages in CICS, you may want to consider imposing transaction class limits on the IBM Connect:Direct transaction. You can impose class limits by first assigning a transaction class to the transaction, and then placing a limit on the number of transactions in that class. You may also want to assign the IBM Connect:Direct transaction to a lower priority than those of other tasks in your system to increase system throughput.

Using the Administration System

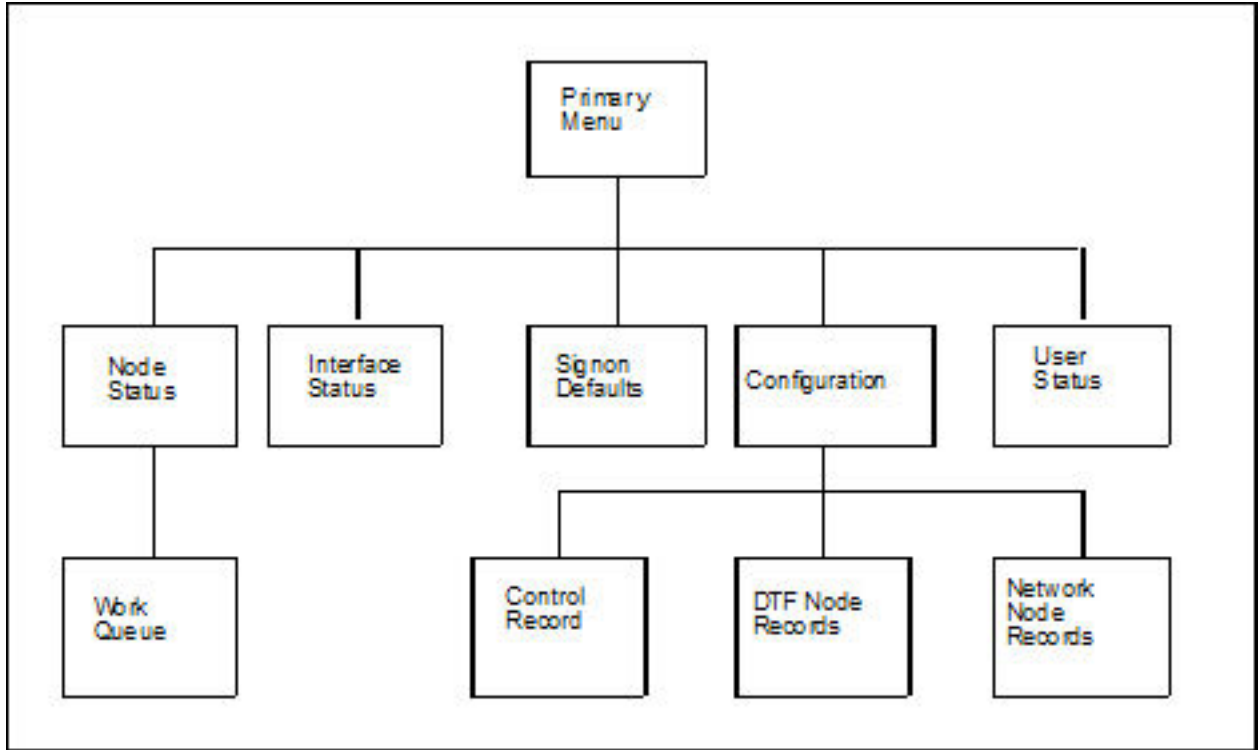
The Administration System helps you configure and control the interface. Tasks you perform while using this system include:

- Configuring Connect:Direct for z/OS to provide functionality to users while maintaining optimum performance
- Defining IBM Connect:Direct DTFs to which Connect:Direct for z/OS submits requests
- Activating and deactivating the IBM Connect:Direct interface
- Activating and deactivating the interface with specific IBM Connect:Direct DTFs
- Adding, modifying, and deleting default user signon characteristics
- Monitoring user activity and intervening if those activities compromise IBM Connect:Direct or the CICS environment

The system guides you through a series of menus, prompts you for input, and performs the requested function. If errors occur during the processing of your request, the system informs you of the cause of the error and, in some cases, suggests remedial action.

Structure of the Administration System

The menu structure of the Administration System follows.



By typing the correct data in the Administration menus and screens, perform the following tasks:

- Check and control user status (USER STATUS)
- Add, update, and delete user signon characteristics (SIGNON DEFAULTS)
- Activate and deactivate the IBM Connect:Direct DTF-to-CICS Interface (INTERFACE)
- Monitor and control node status (NODE STATUS)
- Monitor and control the work queue (WORK QUEUE)
- Modify the global control parameters (CONTROL RECORD)
- Add, delete or modify any DTF node parameters (DTF NODE RECORDS)
- Add, delete or modify any network node configurations (NETWORK NODE RECORDS)

The Administration Primary Menu

The IBM Connect:Direct for z/OS ADMINISTRATION PRIMARY MENU is the root of the administration menu hierarchy and is the access key to all the other features of the IBM Connect:Direct Administration System. Following is an example of the Administration Primary Menu.

```

IBM Connect:Direct for z/OS
ADMINISTRATION PRIMARY MENU      10:17:29

OPTION ==> _

C ... CONFIGURATION
I ... INTERFACE STATUS
N ... NODE STATUS
S ... SIGNON DEFAULTS
U ... USER STATUS

      To restrict U :
      CICS USERID ==> -----
      CICS TERMID ==> -----
Connect:Direct NODE ==> -----

Licensed Materials - property of IBM
IBM Connect:Direct for z/OS
(C) Copyright IBM Corp, 1982, 2011 All Rights Reserved.
IBM is a Trademark of International Business Machines

PF keys:  3 Exit

```

Note: In order to use the Administration System, both CICS and IBM Connect:Direct must be installed and working on your mainframe. Display the Administration Primary Menu by using the DGAA transaction.

The following table describes each option.

Option	Description
C	Select this option to maintain CICS Interface configuration information.
I	Selecting this option takes you to the INTERFACE STATUS menu. If you need to activate or deactivate the IBM Connect:Direct interface, select this option.
N	Selecting this option takes you to the NODE STATUS menu. If you need to view the status of nodes that users of the CICS Interface can sign on to, and information about the sessions and tasks under the node, select this option.
S	Selecting this option takes you to the SIGNON DEFAULTS menu. If you need to maintain user signon default information, select this option.
U	Selecting this option takes you to the USER STATUS menu. If the you need to check on a particular userid and the IBM Connect:Direct activities associated with that userid, or if you need to cancel a user task or signon, select this option.

The following fields are not required under option U, but you can use them to limit the scope of the display:

Field	Description
CICS USERID	This 8-character field contains the IBM Connect:Direct CICS signon ID.
CICS TERMID	This 4-character field contains a valid terminal ID.
Connect:Direct NODE	This 16-character field contains a valid IBM Connect:Direct node name.

Maintaining Configuration Information

The CICS Interface provides a variety of configuration options to allow you to define the resources that can be accessed, limit the scope of functions provided to users, and optimize the performance of the

system. All CICS Interface configuration information is contained in a single file (CONFIG), which is defined and initially loaded at installation.

Configuration Categories

The configuration parameters are divided into the following categories:

- Control Record
- DTF Node Records
- Network Node Records

Control Record

The Connect:Direct for z/OS Control Record contains information defining the operational characteristics of the Connect:Direct for z/OS system. This information includes flags that activate or deactivate system-wide features of the CICS Interface and parameters that govern the performance of the system.

The Control Record is initially loaded during the installation process. The CICS Interface online administration facilities allow you to modify the contents of this record only. Any modifications performed while the CICS Interface is active are immediately reflected in the execution environment.

DTF Node Records

A CICS Interface user has the ability to access multiple IBM Connect:Direct DTFs for the purpose of copying files from that node to other nodes, submitting Processes, and gathering and reporting on statistical information.

The DTF Node Records contain information identifying any node eligible to be signed on to by the CICS Interface and defining the rules governing access to that node (such as ESF access, output limits, and so forth).

Note: One DTF Node Record is required for each IBM Connect:Direct DTF to which the CICS Interface can sign on directly. The first of these records is defined and loaded at product installation.

IBM Connect:Direct DTF node information describes each DTF node available to the CICS Interface. At the startup of the Interface, a subtask called CST (controller subtask) is attached by CICS. The CST in turn attaches other subtasks, called WSTs (worker subtasks).

WSTs establish and manage a VTAM session with a DTF, passing IBM Connect:Direct commands, and receiving returned information. One WST communicates with only one DTF, but multiple WSTs can communicate with one DTF.

Network Node Records

To facilitate the task of copying files from one node to another, the CICS Interface provides a means of predefining frequently used nodes, relieving you of the need to know specific information about nodes to be accessed. Nodes defined in Network Node Records display as a numbered list in the COPY FILE BETWEEN NODES screen. You can select nodes by number and provide only that information relating to the file to be sent or received.

Accessing the Configuration Screen

To access the CONFIGURATION screen, select option C from the PRIMARY MENU (DGAA transaction). The CONFIGURATION screen enables you to display and maintain the Control Record, DTF Node Records, and the Network Node Records. Following is an example of the screen.

CONNECT:DIRECT ADMINISTRATION CONFIGURATION		10:25:28
OPTION ==> _		
C ... CONTROL RECORD D ... DTF NODE RECORDS N ... NETWORK NODE RECORDS		
PF keys: 3 Exit		

The following table describes each configuration option.

Option	Description
C	Selecting this option, and pressing Enter takes you to the CONTROL RECORD screen. If you need to update the CONTROL RECORD with information affecting the operating characteristics of the CICS Interface, select this option.
D	Selecting this option, and pressing Enter takes you to the DTF NODE RECORDS screen. If you need to view the characteristics of a DTF NODE, such as node name, number of worker subtasks, and entries in the work queue, select this option.
N	Selecting this option, and pressing Enter takes you to the NETWORK NODE RECORDS screen. If you need to view the characteristics of a NETWORK NODE, such as node name, node description, and node type, select this option.

Updating the Control Record

To access the CONTROL RECORD UPDATE screen, select option C from the CONFIGURATION screen, and press Enter. Only one CONTROL RECORD exists for the CICS Interface system. This control record contains global configuration parameters. Following is an example of the screen.

CONNECT:DIRECT ADMINISTRATION CONTROL RECORD UPDATE		16:13:15
AUTO.SIGNON	Y	Y OR N
SIGNON.REENTRY	N	Y OR N
CONNECT:DIRECT.EQ.CICSID	Y	Y OR N
SKIP.SIGNON.PANEL	N	Y OR N
CICS.TRANSACTION.CODE (MONITOR)	DGADM	
CICS.TRANSACTION.CODE (STARTUP)	DGAI	
CICS.TRANSACTION.CODE (PRINT)	DGAP	
CICS.TRANSACTION.CODE (ESS)	DGAE	
CST.RETRY.INTERVAL	000500	HHMMSS
SESSION.RETRY.INTERVAL	0100	MMSS
ESF.RETRY.INTERVAL	001500	HHMMSS
WORK.RETRY.INTERVAL	0015	MMSS
MONITOR.INTERVAL	30	SS
INACTIVE.INTERVAL	003000	HHMMSS
MAX.SIGNON	0100	
MAX.TASKS	02	01-99
STORAGE.SUBPOOL	127	002-127
MENU OPTIONS:		
CF Y SB Y SS Y SP Y SD Y SN Y MD Y		Y OR N
PF keys: 3 Exit 9 Update Enter Edit Clear Reset		

The entry fields are:

Field	Description
AUTO.SIGNON	Specifies whether Connect:Direct for z/OS automatically signs you on if a signon defaults record is defined with a CICS userid matching the userid that you specify at CICS signon. The signon defaults record for that CICS user must specify a valid IBM Connect:Direct userid, password and DTF node name. If you specify Y, auto-signon is used if the required information is available; if you specify N, auto-signon is not used. The default is Y.
SIGNON.REENTRY	<p>Specifies whether the CICS Interface remembers that a CICS user previously signed on to the IUI. If this feature is enabled, the user is able to exit the IBM Connect:Direct CICS Interface to perform another CICS function and reenter without signing on again.</p> <p>Signon reentry is not performed for a user who signs off CICS itself and then signs back on again to CICS. Signon reentry is only in effect after the first signon to the IBM Connect:Direct CICS Interface. If you specify Y, signon reentry is performed when appropriate; if you specify N, signon reentry is not performed. The default is N.</p>
CONNECT:DIRECT.EQ.CICSID	Specifies whether the CICS Interface denies a signon attempt if the IBM Connect:Direct userid does not match the CICS userid specified at CICS signon. If you specify Y, a signon is rejected if the IDs do not match; if you specify N, no check takes place. The default is Y.
SKIP.SIGNON.PANEL	Specifies an optional CICS signon interface that does not require you to retype a userid and password. Validity of this approach depends upon a secure environment existing prior to you selecting the CICS Interface; in other words, userid and password validation by a security subsystem (CA-ACF2, RACF®, and so forth) upon original signon to the system. Enable this option by typing Y for this parameter on the Control Record Update screen. The control record can also be set during installation by typing SKIP.SIGNON.PANEL=Y on the CONFIGURATION FILE - CONTROL PARAMETER RECORD screen. The default is N.
CICS.TRANSACTION.CODE (MONITOR)	Specifies the 1-4 character transaction codes to be used for the CICS Interface monitor transaction. The monitor transaction scans for pending requests from users and for completed work by IBM Connect:Direct. If the MONITOR TRANSACTION CODE is not specified during installation, it defaults to DGAM. If you change this parameter, you must also change the supplied transaction definition.
CICS.TRANSACTION.CODE (STARTUP)	Specifies the 1-4 character transaction codes to be used for the CICS Interface start transaction. If you use the startup PLT to activate the CICS Interface at CICS initialization, this transaction is submitted to run immediately following the completion of CICS initialization processing. If you do not specify the STARTUP TRANSACTION CODE parameter during installation, it defaults to DGAI. If you change this parameter, you must also change the supplied transaction definition.

Field	Description
CICS.TRANSACTION.CODE (PRINT)	Specifies the 1-4 character transaction codes to be used for the CICS Interface print transaction. This transaction is attached to the CICS printer in response to CICS print requests. If the PRINTER TRANSACTION CODE parameter is not specified during installation, it defaults to DGAP. If you change this parameter, you must also change the supplied transaction definition. Print requests are handled by writing print lines to the CICS Transient Data Area (TDA). When the data is ready for output, a transaction sends the data to the CICS printer specified in the SIGNON DEFAULTS.
CICS.TRANSACTION.CODE (ESS)	Specifies the 1-4 character transaction code to be used for the Connect:Direct for z/OS Event Services Support. If this parameter is not specified during installation, it defaults to DGAE. If you change this parameter, you must also change the supplied transaction definition.
CST.RETRY.INTERVAL	Specifies the time interval (in hours, minutes and seconds) between attempts to restart an ABEND of CST (controller subtask). The controller subtask is an operating system subtask responsible for monitoring the worker subtasks responsible for interaction with active IBM Connect:Direct DTFs defined in DTF Node Records (that is, DTFs to which CICS users can sign on). The default is 000500 (5 minutes).
SESSION.RETRY.INTERVAL	Specifies the time interval (in minutes and seconds) between attempts to establish a VTAM session with a DTF defined in a DTF Node Record (that is, a DTF to which CICS users can sign on directly). The default is 100 (1 minute).
ESF.RETRY.INTERVAL	Defines the time interval (in hours, minutes and seconds) between attempts to establish a primary session with a IBM Connect:Direct DTF when, during a prior attempt, the DTF is not active and the node is activated in ESF mode. A node is activated in ESF mode only if the DTF it defines supports ESF. The default is 001500 (15 minutes).
WORK.RETRY.INTERVAL	Specifies the time interval (in minutes and seconds) between the time a unit of work is submitted but cannot be placed in the work queue for a particular node and the time that unit of work is cancelled. The size of the work queue for a particular node is governed by the ENTRIES IN WORK QUEUE parameter in the DTF Node Record. The default is 0015 (15 seconds).
MONITOR.INTERVAL	Contains the time interval, expressed as seconds, between scans for work by the monitor transaction. This parameter is a 2-character field. The monitor watches all Processes, queues, tasks, and task lengths, and regulates the flow of tasks in the system to ensure that any particular task does not seize the computer resources. When all Processes and tasks are completed or pending action by the DTF, the monitor waits for the specified interval before rescanning the work queue. The default is 30 (seconds).

Field	Description
INACTIVE.INTERVAL	Specifies the amount of time (in hours, minutes and seconds) that a worker subtask (WST) is allowed to be inactive before it is detached by the controller subtask (CST). The default is 003000 (30 minutes).
MAX.SIGNON	Contains the maximum number of CICS users and reflects the relative size of the signon table file. This parameter is a 4-character field. The default is 100 entries.
MAX.TASKS	Contains the maximum number of simultaneous subtasks that can be attached in the CICS address space. This parameter is a 2-character field. Specify this field as the total of the worker subtask counts for all DTF Node Records defined to the CICS Interface. The default is 2.
STORAGE.SUBPOOL	Contains the number of the operating system storage subpool from which the CICS Interface acquires operating system storage. This parameter is a 3-character field. Numbers 002-127 are user-defined (numbers 000, 001, 128-255 are system-defined) areas of storage related to each other usually by the requirements of your session. DTF keeps track of all storage for a given user Process and task. The default is subpool 127.

Menu Options

The following options contain a Y or N which enables or disables the options on the PRIMARY MENU of the general user. You can use these Y/N flags to determine the level of functionality to be provided by IBM Connect:Direct.

These options affect all users of Connect:Direct for z/OS. If you want to limit the functionality available to specific users, you must use either the authorization functions of IBM Connect:Direct (select only the appropriate options when defining that user in the IBM Connect:Direct Authorization File) or the IBM Connect:Direct Security exit (set an Authorization Bit Mask to allow or restrict the appropriate options).

Option	Description
CF	Contains the toggle to turn off the COPY FILE option on the PRIMARY MENU. The field is one character long; Y permits you to copy files; N denies permission.
SB	Contains the toggle to turn off the SUBMIT PROCESS option on the PRIMARY MENU. The field is one character long; Y permits you to submit Processes; N denies permission.
SS	Contains the toggle to turn off the SELECT STATISTICS option on the PRIMARY MENU. The field is one character long; Y permits you to select statistics; N denies permission.
SP	Toggle to turn off the SELECT PROCESS option on the PRIMARY MENU. The field is one character long; Y permits you to select Processes; N denies permission.
SD	Contains the toggle to turn off the SIGNON DEFAULTS option on the PRIMARY MENU. The field is one character long with a Y or N; Y grants permission to change default signon options; N denies permission.
SN	Contains the toggle to turn off the CHANGE SIGNON option on the PRIMARY MENU. The field is one character long with a Y or N; Y grants permission; N does not.
MD	Contains the toggle to turn off the MESSAGE DISPLAY option on the PRIMARY MENU. The field is one character long with a Y or N; Y grants permission to use this option; N does not.

Updating DTF Node Records

The DTF NODE RECORDS screen is accessed by selecting option C on the PRIMARY MENU, then by selecting option D on the CONFIGURATION screen, and pressing Enter. One DTF node record exists for each DTF node which IBM Connect:Direct users can sign on to directly. Following is an example of the screen.

Note: Changes made to the DTF node record parameters are immediately reflected in the active system.

CONNECT:DIRECT ADMINISTRATION		16:17:53
DTF NODE RECORDS		
DTF NODE NAME	-----	
NETMAP DDNAME	-----	
DUMMY ID FOR DTF SIGNON	-----	
SUPPRESS CONNECTION AT STARTUP	-	Y OR N
ESF SIGNON ALLOWED	-	Y OR N
MAXIMUM WORKER SUBTASKS	--	
ENTRIES IN WORK QUEUE	---	
OUTPUT RECORD LIMIT	----	
SLOW RESPONSE NOTIFICATION	-----	MMSS
PF keys: 3 Exit 5 Add 6 Delete 7 Prev 8 Next 9 Update Enter Read/Edit Clear Reset		

The following table describes the Entry fields:

Field	Description
DTF NODE NAME	Contains the name of a DTF node. This is a 16-character field.
NETMAP DDNAME	<p>Specifies the DDNAME of the IBM Connect:Direct network map file to be used when initiating a signon to this node. The network map must have an adjacent node definition for this node, but need not be exactly the same network map file the node is using. You must code this value; no default is available.</p> <p>The NETMAP DDNAME is defined to CICS using RDO and is the FILENAME used by CICS. You must have one network map file for each node and you must have the DTF node record for each node you want to sign on to.</p>
DUMMY ID FOR DTF SIGNON	Specifies the IBM Connect:Direct userid to be used to initially establish the VTAM session with the DTF. Multiple CICS userids are required if a DTF can have multiple CICS Interface systems signed on concurrently. If you are using the DGASECUR macro to define your IBM Connect:Direct security exit, this parameter must match the CICS parameter coded for that macro. You must code this value; no default is available.
SUPPRESS CONNECTION AT STARTUP	Contains the toggle for startup connections. This is a 1-character field. If the field contains Y, the connection between the specified DTF node and the IBM Connect:Direct system is suppressed when the IBM Connect:Direct software is started. If the field contains N, then the connection is made. The default is Y.

Field	Description
ESF SIGNON ALLOWED	Specifies whether the Extended Submit Facility is to be supported for this node by IBM Connect:Direct. The DTF identified in this record must support ESF in order to sign on in ESF mode using IBM Connect:Direct. The default is N.
MAXIMUM WORKER SUBTASKS	Specifies the maximum number of WSTs (worker subtasks) to be used for this node. Define one WST per parallel session defined for this node. Do not exceed the MAX TASKS value in the Control Record with the total number of worker subtasks defined for all DTF NODE records in the system. The default is 2.
ENTRIES IN WORK QUEUE	Specifies the maximum number of actual requests to be allowed on the pending work queue for this node. An excessive value here could result in an inordinate response time for terminal users. During installation, the default is the MAXIMUM WORKER SUBTASKS specification.
OUTPUT RECORD LIMIT	Specifies the upper limit on the number of lines of output that are accepted in response to a SELECT PROCESS or SELECT STATISTICS command. Output from these commands is stored in CICS temporary storage until viewed or explicitly deleted. The default is a limit of 800 80-byte records.
SLOW RESPONSE NOTIFICATION	Contains the time interval, expressed as minutes and seconds, after which IBM Connect:Direct notifies you of potential problems with slow response. This is a 4-character field. The default is 0200 (2 minutes).

Updating Network Node Records

To access the Connect:Direct ADMINISTRATION NETWORK NODE RECORDS screen, select option C from the PRIMARY MENU, select option N from the CONFIGURATION screen, and press Enter. The screen fields are scrollable, and allow you to view the contents of the network node records. Following is an example of the screen.

```

CONNECT:DIRECT ADMINISTRATION
NETWORK NODE RECORDS
16:02:23

NETWORK NODE NAME
-----

NETWORK NODE DESCRIPTION
-----

NODE TYPE
--  1=OS/390    2=VM      3=VSE      4=VMS
      5=TANDEM  6=WIN95   7=OS/2    8=OS/400
      9=UNIX    10=NETWARE 11=WINDOWS 12=MSP
     13=MVS

```

PF keys: 3 Exit 5 Add 6 Delete 7 Prev 8 Next 9 Update

Enter Read/Edit Clear Reset

Because you can type the node name and environment on the COPY FILE BETWEEN NODES screen, it is not required that you define every node participating in a COPYFILE on the NETWORK NODE RECORDS screen.

Note: Changes made to the Network Node Record parameters are immediately reflected in the active system.

The following table describes the entry fields.

Field	Description
NETWORK NODE NAME	(16-character field) contains the name of the node.
NETWORK NODE DESCRIPTION	(30-character field) contains a description of the node.
NODE TYPE	(1-character field) contains the environment number. Valid environment numbers are as follows: 1 for OS/390® 2 for VM 3 for VSE 4 for VMS 5 for TANDEM 6 for WIN95 7 for OS/2 8 for OS/400® 9 for UNIX 10 for NETWARE 11 for WINDOWS 13 for MVS™

Working with the Administration Interface

To access the IBM Connect:Direct ADMINISTRATION INTERFACE screen, select option I from the PRIMARY MENU and press Enter.

IBM CONNECT:DIRECT ADMINISTRATION INTERFACE		10:30:37
OPTION ==>		CONNECT:DIRECT VER VV REL RR MOD MM
	A ... ACTIVATE INTERFACE	
	M ... START MONITOR	
	I ... SHUTDOWN INTERFACE (IMMEDIATE)	
	S ... SHUTDOWN INTERFACE (NORMAL)	
INTERFACE STATUS	ACTIVE	
PENDING REQUEST	NONE	
MONITOR TASK NUMBER	25	
ACTIVE TASKS	0	
PF keys: 3 Exit ENTER Refresh/Process		

The INTERFACE screen is the key to activating, monitoring, and shutting down the CICS Interface between CICS and the active IBM Connect:Direct DTF nodes. From the INTERFACE screen, you can manually initialize and terminate the operating system subtasks that perform the interaction with IBM Connect:Direct.

The following table describes each option for the INTERFACE screen.

Options	Description
A	Selecting this option, and pressing Enter activates the interface and automatically starts the monitor transaction.
M	Selecting this option, and pressing Enter starts the monitor transaction. Only use this option in the event of a monitor transaction ABEND.
I	Selecting this option, and pressing Enter performs an immediate (hard) shutdown of the interface. All IBM Connect:Direct user sessions are terminated, regardless of status.
S	Selecting this option, and pressing Enter, performs a normal (soft) shutdown of the interface. All IBM Connect:Direct user sessions are allowed to complete execution.

The following table describes the system fields:

Field	Description
VER	(2-character field) contains the version number of the CICS Interface software.
REL	(2-character field) contains the release number of the CICS Interface software.
MOD	(2-character field) contains the modification number of the CICS Interface software.
INTERFACE STATUS	(21-character field) contains ACTIVE or INACTIVE, depending upon the state of the interface. The field contains ACTIVE if the interface is active and changes to INACTIVE if the interface is deactivated.
PENDING REQUEST	(18-character field) contains either NONE, if no activate or shutdown request is pending, or the type of request being processed.
MONITOR TASK NUMBER	(11-character field) contains the number of the monitor transaction or a message such as NOT RUNNING, if the interface is not active.
ACTIVE TASKS	(2-character field) contains the number of the active tasks.

Operating the CICS Interface

In order for the CICS user interface to perform its function, a connection must exist between it and a IBM Connect:Direct DTF. This facility can be local (within the same VTAM domain) or remote (residing in another domain).

This connection is accomplished using operating system subtasks that are attached in the CICS address space at the CICS Interface startup. The subtasks perform two types of functions:

One or more Worker Subtasks (WSTs) are attached for each active link to a IBM Connect:Direct DTF. These subtasks are responsible for establishing a link to the DTF, passing commands to it, and receiving any results.

A single Controller Subtask (CST) is created to monitor the work of all WSTs in the system and perform communications functions with the user interface portion of CICS.

Initialization of the CICS Interface invokes programs that perform the following functions:

- Acquires CICS Interface work areas, such as the signon table and work queue areas
- Enables a task control global exit point and its associated global work area
- Reads the configuration information from the configuration file and places that information in the global work area
- Attaches the CST subtask
- Checks for any DTF nodes that are to be activated at initialization and passes information to the CST task to allow it to create the appropriate WSTs and establish the link with the DTF.

Note: The CICS Interface module is also available for the interface autostart at CICS startup. When the module is in the PLTPI, you can start CICS without starting the interface by including a special DD card in the CICS startup JCL deck: **//NDMINIT DD DUMMY**. If present, NDMINIT will not start the interface as part of CICS initialization. The interface can then be started manually.

Monitor Transaction

In addition to these operating system subtasks, a CICS monitor transaction is created to accept input from the user interface, pass it to the CST, and route the output to the appropriate user. This transaction is also responsible for detecting a loss of the CST subtask due to an error condition and performing a restart. The monitor transaction is activated at the CICS Interface startup and remains in the system for the life of the online region. Monitor activities include the following:

- Restarting the interface in case of an ABEND
- Watching for DTF requests which are taking too long
- Forcing retry of requests still in the queue
- Forcing retry of DTF node session establishment
- Attempting to switch from ESF to primary (DTF connected) mode
- Clearing non-terminal signon table entries, at end of transaction
- Clearing signon table entries in case of ABENDs

Interface Tasks

Although CICS runs a number of subtasks (such as journaling and VSAM handling), CICS can be regarded as a single task. All CICS transactions that can generate IBM Connect:Direct requests run from this single CICS task. The CICS Interface runs as a set of separate tasks in the CICS address space. At the startup of the CICS Interface, a subtask called CST (controller subtask) is attached by CICS.

Interface Startup

This isolates CICS from all the non-CICS work involved in communicating to the IBM Connect:Direct DTF through the IBM Connect:Direct API. A CICS Interface monitor transaction is also invoked as part of interface startup to support and monitor interface operation.

The CST in turn attaches other subtasks, called WSTs (worker subtasks), organized by node. The WSTs establish and manage the DTF sessions, passing IBM Connect:Direct commands, and receiving returned information. The CST controls traffic between all CICS Interface transactions and each WST.

Interface Subtask Management

Each WST handles requests for CICS users through the CST. One WST communicates with only one DTF, but multiple WSTs can communicate with one DTF. The DTF-connected WSTs are not associated with any particular CICS transactions. The various IBM Connect:Direct requests generated by CICS Interface users are handled by any of the multiple WSTs which are for a particular DTF node. The CST oversees all the WSTs, and is responsible for coordinating work generated by CICS transactions.

One WST is attached per node at the CICS Interface startup, as defined in the CONFIGURATION file. Additional WSTs for a node are attached as needed, based on concurrent CICS Interface user demand. The maximum number of WSTs (tasks) attached globally for the interface and per DTF node is defined with CONFIGURATION control and DTF node parameters. The DTF node connection limit is edited to be no larger than the number of CICS interactive applications specified in the Network Map for that node.

When an attached WST becomes inactive (is not used for any requests) for a period of time, it is detached by CST. The inactive interval is defined with a CONFIGURATION control option. You can suppress WST activation by DTF node with a CONFIGURATION DTF node parameter.

Interface Request Management

CICS user requests are placed on a work queue with one queue per node. The one or more WSTs attached per node dispatch work from the queue. Maximum queue length by node is defined with a CONFIGURATION DTF node parameter which defines the number of queue entries.

When WST (session limit) is reached and additional WSTs cannot be attached to dispatch work from the queue, requests remain on the work queue until a WST becomes available. When the number of allowed WSTs for a given node is held to a minimum, yet user activity for the given node is high, increase the number of queue entries.

Work queue size is also affected by the CONFIGURATION control option worker retry interval which defines the time elapsed between attempts to obtain a free WST to dispatch work present on the work queue. IBM Connect:Direct user requests are rejected with a DTF busy message, when a DTF node queue reaches its maximum allowed number of entries.

The CONFIGURATION DTF node parameter defines worry time. A message is written to the log when a request to that node takes longer than the specified amount of time. No action is taken by the CICS Interface to automatically abort any requests which take too long.

The CICS Interface also has one system queue for system commands (for example, SHUTDOWN and QUIESCE) to dispatch system requests. You cannot adjust the system queue size.

Interface VTAM Session

In cases where a WST/DTF VTAM session is active, but becomes inactive, the CICS Interface quiesces the node. Pending requests are allowed to complete, even though they can fail. The number of WSTs for the node is reduced to one.

At this point, if the WST is in session with a local DTF, and ESF mode is allowed, the WST switches to ESF mode. An ESF MODE message is displayed to IUI users, stating that a IBM Connect:Direct session error occurred, but ESF MODE is available. Under ESF operation, only the SUBMIT options are displayed on the PRIMARY MENU.

ESF mode operation is enabled for the entire interface through a CONFIGURATION control parameter; ESF mode operation is enabled for each user through a user profile parameter. When a WST fails to establish or drops a DTF session, and ESF is not allowed, then the WST remains attached, but all user requests for the node are rejected.

The WST periodically tries to establish or re-establish (retry) a session with the associated DTF. The CONFIGURATION control parameter session retry interval defines the time lapsed between retries for session connection. An additional CONFIGURATION control parameter, the ESF session retry interval, defines the time elapsed between retries of a dummy DTF session to check if ESF mode has returned to primary mode.

Viewing Node Status

About this task

The Node Status screen enables you to check the status of all DTF nodes eligible for access by the CICS Interface, and you can selectively activate, deactivate, and view pending work for all DTF nodes. The CICS user interface provides the ability to log on to any IBM Connect:Direct DTF in your network and perform IBM Connect:Direct operations using that node as your Process primary node. DTFs to be accessed must be identified in DTF Node Records in the configuration file along with the configuration parameters to be used when communicating with that node.

To scroll backward and forward, through the list of DTF nodes connected to the Connect:Direct for z/OS session, press PF7 and PF8, respectively.

Procedure

1. To access the CONNECT:DIRECT ADMINISTRATION NODE STATUS screen, select option N from the ADMINISTRATION PRIMARY MENU and press Enter.

Following is an example of the Node Status screen.

CONNECT:DIRECT ADMINISTRATION NODE STATUS								10:30:37	
DTF NODE NAME	CICS STATUS	ADMIN REQUEST	SESS TYPE	SESSION OR	DATE/TIME	MSGID	MAX TASKS	CURR TASKS	CURR WORK
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
_ NODE1	INACT		NONE				3	0	0
_ NODE2	ACTIVE		PRIM	06/19/1998	09:57:18		2	1	0
Line commands: A Activate (start first task) I Shut immediate									
W Work queue display S Shut normal									
PF keys: 3 Exit ENTER Refresh/Line cmd									

The following table describes the fields on the NODE STATUS screen:

Field	Description
DTF NODE NAME	(16-character field) contains the DTF node name as typed in the DTF Node Record.
CICS STATUS	(6-character field) contains the status of the node, such as ACTIVE and INACT. ACTIVE - The node is activated, either by the CICS Interface initialization or manually. INACT - The node is not active.
ADMIN REQUEST	This 8-character field contains the type of administrative request affecting the status of the node, such as ACTIVATE, SHUTIMM, or SHUTNORM. ACTIVATE - The node is activated. SHUTNORM - A normal shutdown of the node is requested, and the node is quiescing. SHUTIMM - An immediate shutdown of the node is requested, and the link with that DTF is being terminated.
SESS TYPE	This 4-character field contains the type of session held with this DTF node, as follows: NONE - No session currently exists with this DTF node. PRIM - The CICS Interface is currently in session with the DTF node. ESF - The DTF is not active, but supports the Extended Submit Facility. The CICS Interface is accepting requests allowed in ESF mode.
SESSION DATE/TIME	This 17-character field contains either the date and time that the connection to this DTF is activated, or the MSGID of the last message issued for this DTF node during activation or deactivation.
MAX TASKS	This 5-character field contains the maximum number of subtasks that are attached to Process requests directed at this node.
CURR TASKS	This 5-character field contains the current number of subtasks attached to Process requests directed at this node.

Field	Description
CURR WORK	This 5-character field contains the number of subtasks currently processing requests directed at this node.

2. Take one of the following actions in the line command column to the left of the DTF node name:

- To activate the node, type A and press Enter.
- To access the WORK QUEUE screen for a node where work is being processed, type W and press Enter.
- To perform an immediate (hard) shutdown of the node to force all Processes to stop, regardless of status, type I and press Enter.
- To perform a normal (soft) shutdown of the node to allow all Processes to complete execution, type S and press Enter.

Viewing the Work Queue Screen

The IBM Connect:Direct Work Queue is defined for each node to dispatch IBM Connect:Direct requests and responses. The DTF node configuration file contains information required by the CICS Interface to manage the DTF VTAM or ESF sessions. Each WST uses the standard IBM Connect:Direct API to manage the DTF or ESF session.

To access the WORK QUEUE screen for a node where work is being processed, type W and press Enter.

The WORK QUEUE screen fields contain the data describing the tasks in the work queue for each CICS user on a IBM Connect:Direct DTF node. Following is an example of the screen.

CONNECT:DIRECT ADMINISTRATION							10:30:37
WORK QUEUE - NODE "nodename"							Page 01 of 01
CICS ID	USERID	CICS TERM	LAST TASK#	CURR CMD	REQUEST DATE/TIME	TD CTR	WORK TASK
-----	-----	----	-----	---	-----	-----	-----
ID1	ID1	M064	00271	SB	06/21/1998 11:47:13	00282	0346
PF keys: 3 Exit 7 Bwd 8 Fwd 12 Node USER STATUS ENTER Refresh							

Scroll backward using PF7 and forward using PF8 to view the entries. Press PF12 to view to the USER STATUS screen, and then press PF3 to get back to the WORK QUEUE screen.

The system fields are as follows:

Field	Description
NODE	(16-character field) contains the node name associated with the work queue data.
Page XX of YY	(13-character field) contains the number of the current page for the work queue list.
CICS ID	(8-character field) contains the CICS userid of the user submitting the work.
USERID	(8-character field) contains the CICS userid of the user submitting the work.
CICS TERM	(4-character field) contains the terminal ID from which the work is submitted.
LAST TASK#	(5-character field) contains the transaction number of the task submitting this request.
CURR CMD	(2-character field) contains the representation of the current command executed by the task (for example, CF, SB, SS).

Field	Description
REQUEST DATE/ TIME	(17-character field) contains the date and time the work is submitted.
TD CTR	(5-character field) contains the number of bytes (counted by the transient data counter) indicating how much data is written by the exit module for a transaction.
WORK TASK	(4-character field) contains the number of the work task.

Signon Defaults

IBM Connect:Direct enables you to set default CICS signon information. This information is not required but it can make IBM Connect:Direct easier to use.

The system automatically uses a CICS userid to read the profile dataset when a transaction is entered. If it finds a profile record, IBM Connect:Direct uses the record's information to control what the user can do.

If the user's signon defaults record includes a userid and password, IBM Connect:Direct automatically signs on the user. If the signon defaults record does not include a userid and password, IBM Connect:Direct prompts the user for a userid and password before allowing access.

After accessing IBM Connect:Direct, you can change your own signon defaults using the SD function.

Accessing the Signon Defaults Screen

To access the Connect:Direct Administration Signon Defaults screen, type S and press Enter on the Administration Primary Menu . The screen is shown in the following figure.

```

                                CONNECT:DIRECT ADMINISTRATION          14:59:46
                                SIGNON DEFAULTS

CICS USERID                      ==>  _____

**Connect:Direct**
USERID ==>  _____
PASSWORD==>  _____

DEFAULT NODE                      ==>  _____
ESF MODE ALLOWED                  ==>  -   Y OR N
UPPER CASE PRINT                  ==>  -   Y OR N
CICS PRINTER                      ==>  _____
PNODE ACCT DATA                  ==>  _____
SNODE ACCT DATA                  ==>  _____

Do you want all commands for this session to be CASE sensitive? ==>  NO_

PF keys:  3 Exit   5 Add   6 Delete  7 Prev   8 Next   9 Update
          Enter Read/Edit  Clear Reset

```

The following table describes the entry fields.

Field	Description
CICS USERID	(8-character field) is the acceptable userid for the CICS signon.
USERID	(64-character field) is the acceptable userid for the IBM Connect:Direct signon.
PASSWORD	(64-character field) is the valid password associated with the IBM Connect:Direct userid.
DEFAULT NODE	(16-character field) is the name of the default IBM Connect:Direct node. The user is automatically signed on to this node if it is active, or is denied access if it is not active.

Field	Description
ESF MODE ALLOWED	(1-character field) is the permission for use of the ESF. If the field contains Y, permission is allowed. If the field contains N, permission is denied.
UPPER CASE PRINT	(1-character field) is the switch for upper case printing. If the field contains Y, all printed output is upper case. If the field contains N, the printed output is upper and lower case.
CICS PRINTER	(4-character field) is the designation for the CICS printer used for print requests from this user.
PNODE ACCT DATA	(50-character field) is the primary node accounting data for allocation of budget to CPU time, I/O, and other computer resources.
SNODE ACCT DATA	(50-character field) is the secondary node accounting data for allocation of budget to users for CPU, I/O, and computer resource time.

The following table describes the system field.

Field	Description
PASSWORD MESSAGE	(44-character field) indicates whether the user's signon defaults record has a password. This field is displayed to the right of the PASSWORD field.

Viewing User Status

About this task

Periodically view the status of a user to determine:

- User access to a resource
- Actions users are performing while using a resource
- Resolve problems encountered by users

IBM Connect:Direct provides the user status function for you to view information about users of the system and, if necessary, intervene to resolve error situations. You can either view all users of the system, a single user (optionally qualified by CICS userid or terminal ID) or a group of users (optionally qualified by IBM Connect:Direct DTF node).

You have the option, while on the PRIMARY MENU screen, to restrict the scope of the status display. Restrict the scope by specifying the CICS userid, terminal ID and IBM Connect:Direct node that you want to display on the USER STATUS screen. To see all signed-on users, do not restrict your selection.

Procedure

1. To access the CONNECT:DIRECT ADMINISTRATION USER STATUS screen, type U and press Enter on the PRIMARY MENU,.

Following is an example of the USER STATUS screen.

CONNECT:DIRECT ADMINISTRATION							10:30:37
USER STATUS							
CICS ID	CICS TERM	SESS TYPE	USERID	DTF NODE NAME	STATUS	LAST TASK #	MSGID
-----	-----	-----	-----	-----	-----	-----	-----
_ MASTER	M064	PRIM	ID1	"dtfnodename"	CICS	45	SAFA000I
Line commands: F Free user C Free user and cancel user's subtask							
T Free user and terminate user's signon							
PF keys: 3 Exit 10 Left 11 Right ENTER Refresh/Process							

2. Take one of the following actions in the underscore field to the left of the CICS userid when a user security violation or resource allocation is abused:

- To free a user from use of the CICS Interface system, type F and press Enter.
- To free a user from use of the CICS Interface system and cancel that user's subtask, type F and press Enter.
- To free a user from use of the CICS Interface system and and terminate that user's signon, type T and press Enter.

The following table describes the systems fields:

Field	Description
CICS ID	(8-character field) contains the CICS userid of all currently signed-on users.
CICS TERM	(4-character field) contains the CICS terminal ID of all currently signed-on users.
SESS TYPE	(4-character field) contains the session type (PRIM or NONE).
USERID	(8-character field) contains the IBM Connect:Direct userid of all currently signed-on users.
DTF NODE NAME	(16-character field) contains the DTF node name the user is signed on to.
STATUS	(7-character field) contains the STATUS of a user.
LAST TASK #	contains the last task number of a user Process.
MSGID	contains the message ID of the last message for a given user.

3. To see the following additional system fields, press PF11 to scroll right on the screen.

Field	Description
TD CTR	(5-character field) contains the number of bytes (counted by the transient data counter) indicating how much data is written by the exit module for a transaction.
LAST SIGNON	contains the time of the last signon.
LAST REQUEST	contains the time of the last request.

Operational Considerations

Signing On to Multiple DTFs from a Single IUI

Use the Connect:Direct for z/OS IUI to sign on to multiple IBM Connect:Direct DTFs on either local or remote processors. To take advantage of this facility, consider the following:

- Provide VTAM access to the DTF you want to sign on to. For local DTFs (within the control of the same VTAM subsystem), you must provide IUI APPLIDs that may be used by ISPF or CICS IUI facilities. If the DTF facilities are located on remote processors, you must define the IUI APPLIDs for those DTFs to the local VTAM subsystem as cross-domain resources.
- For each DTF you want to access through the CICS IUI, you must define a DTF Node Record for that facility in your CICS configuration file. You can define the DTF Node Record by using the configuration update facilities of the **DGAA** transaction.

Note:

All DTF nodes do not have to use the same Network Map file; for instance, if you want to communicate with two DTFs with different Network Map contents, you can specify an alternate Network Map in the CICS JCL (or using the RDO) and in the DTF node record in order to communicate with the second DTF, as long as that DTF is defined as an adjacent node in the Network Map.

- The DTF node record in your configuration file must include the DD name of a Network Map file in order to communicate with the remote IBM Connect:Direct DTF. This Network map must have the remote DTF defined as an adjacent node.
- All Processes to be submitted to a remote DTF facility must reside in the Process library (DD name DMPUBLIB) defined for your CICS system.

Signing On to a Single DTF from Multiple IUI Facilities

Not only can you sign on to multiple DTF facilities from a single CICS IUI, but you can also sign on to a DTF from multiple CICS IUI systems. To sign on to a DTF from multiple CICS IUI systems, note the following:

- If you are using the DGASECUR macro to generate your DTF security exit and provide a value for the CICSID keyword at exit generation, all CICS systems accessing that DTF must specify the same CICSID in their signon requests as specified in the CICSID keyword. The password for the CICS signon to a DTF is always CICSUI.
- If the CICSID keyword is not specified in the DGASECUR macro, no checking of CICSIDs for CICS signon requests are performed in the signon exit; however, the CICSID value with a password of CICSUI are passed to your security facility (if available) or to the IBM Connect:Direct Authorization Facility for validation. The CICSID to be used when signing on to a particular DTF is specified in the CICS DTF NODE configuration record for that DTF.
- The CICSID equals the userid that is specified on the DTF NODE RECORDS SCREEN. If you let CICSID default at installation time, the value is CICSUSER. For example, CICSID=CICSUSER.
- Reassemble the supplied security exit for the value to take effect.
- If you do not want to use the DGASECUR macro to generate your DTF security exit, you can recognize the Connect:Direct for z/OS dummy signon by checking the password, which is always CICSUI. When a dummy signon is received from Connect:Direct for z/OS, your security exit returns an Authorization Bit Mask (ABM) of binary zeros.
- To implement the CICS IUI on a base IBM Connect:Direct that has Stage 2 security turned on, modify the supplied security exit. The exit to be modified depends on which security product is running on the system.

Performing an Immediate or Uncontrolled Shutdown

The CICS IUI facility provides two facilities for orderly termination of the interface, as follows:

- Termination is performed automatically by the CICS monitor transaction upon detection of a normal termination of CICS (through a CEMT PERFORM SHUTDOWN command) when the PLTSD in use specifies the Connect:Direct for CICS shutdown program.
- You can terminate the interface manually by using the DGAA transaction.

If you perform an immediate CICS shutdown (through the CEMT PERFORM SHUTDOWN IMMEDIATE command) or if CICS terminates abnormally, you may receive system A03 ABENDs from the z/OS

interface. The ABENDs are generated as a result of region termination without detaching all the operating system subtasks created by the CICS IUI facility.

In order to avoid the additional ABENDs, you must terminate the CICS IUI facility manually, through the DGAA transaction, before you issue the CEMT PERFORM SHUTDOWN IMMEDIATE command.

Restarting Task ABENDS

Administrative options exist to either quiesce (allowing pending requests to complete) or immediately shut down a specific node or the entire CICS Interface. After the shutdown has completed, another administrative option enables you to restart the CICS Interface.

In case the entire CICS Interface ABENDs, CICS is notified. The abnormal termination is recorded in the CICS CWA. When the CICS Interface monitor transaction detects that the interface failed, tries to automatically restart the interface. Users with requests to the interface when it crashed are freed by the CICS Interface monitor and the users are sent a message explaining the problem.

If a WST ABENDs, then the CST is notified. If any request from a CICS user is pending, CST fills in the return code and message for the user, informing the user that the command might have failed. If the failing WST is the only one running for a node, CST attempts to reattach the WST.

Accessing Accounting and Logging Information

IBM Connect:Direct accounting is accomplished by the DTF. Accounting and statistics are gathered accurately as the DTF enables the userid to be extracted from the UICB for each IBM Connect:Direct command entered.

CICS logging is accomplished in the background of CICS operations, but does not record all CICS events and does not duplicate any other IBM Connect:Direct logs. Some events also display on the system console, where major CICS events and errors are reported, such as the following:

- Interface startup
- CST attach
- WST attach
- Node signon of dummy CICS ID
- WST session failure
- WST detach
- CST detach
- CST termination
- Administrative commands affecting sessions and requests
- Return information for CICS users who issue a request and then abnormally exit CICS without waiting for the response

Using the Extended Submit Facility (ESF)

ESF mode is invoked when an active IBM Connect:Direct DTF session fails or when session establishment fails. In order for Connect:Direct for z/OS to activate ESF, you must install the ESF option on the local DTF and you must enable the option in the configuration file.

The user must also include the parameter ESF=YES in the SIGNON command. In ESF mode, a node is available for use only for SUBMIT commands which the user writes directly to the local DTF TCQ file. ESF SUBMIT requests can be issued only by those users who specify ESF as a profile (signon defaults) option.

Specifying IBM Connect:Direct Signon Parameters

Four parameters in the IBM Connect:Direct SIGNON command explicitly support the CICS IUI as follows:

Parameter	Description
TYPE=CICS	Enables a SIGNON command to be embedded in the middle of an API session. This parameter serves no other function and is invalid for a normal signon.
TDEXIT=modname	Enables specification of the exit to receive control for temporary data set I/O. This parameter is mutually exclusive with the TMPDD, TMPDSN, UNIT, and VOLSER parameters. This exit is called for OPENs, CLOSEs, and WRITEs to the temporary data set.
TDLIMIT=nnnnn	Restricts the number of IBM Connect:Direct statistics. Data is returned from the DTF for the IUI Select Statistics (SS) function. The IBM Connect:Direct API SIGNON (used by CICS WSTs to connect to a IBM Connect:Direct DTF) includes a LIMIT= PARAMETER RECORDS returned by the DTF. The DTF truncates data sent to the IBM Connect:Direct WST API and appends a final record indicating that excessive output is truncated.
NETDD=ddname	Enables a SIGNON command to specify the DDNAME of a IBM Connect:Direct Netmap data set which is already allocated.

About IBM Connect:Direct CICS Data

This section lists files and other data used by IBM Connect:Direct DTF and CICS.

IBM Connect:Direct DTF and CICS Data

The following data, accessed in read-only mode under CICS, is used by IBM Connect:Direct DTF and the CICS Interface:

Data	Description
NETMAP (DDN=NETFINP)	At least one netmap file exists for each CICS region. The file is defined as a CICS file. It is updated in batch and contains all IBM Connect:Direct nodes available to this CICS environment. Some of the information for the network IBM Connect:Direct nodes is also contained in the Configuration File along with additional fields that you can update.
PROCESS (DDN=DMPUBLIB)	One DD name exists for each CICS region. Several files can be concatenated. It is not defined as a CICS file. It is allocated by CICS and used at the subtask level by the IBM Connect:Direct API only. If necessary, you can dynamically allocate and deallocate it using CICS transaction DGAN.
MESSAGE (DDN=NDMMMSG)	One file exists for each system. It is defined as a CICS file and is allocated and opened by CICS. It is updated in batch using the IBM Connect:Direct message load utility.
TCQ	One Transmission Control Queue (TCQ) file exists for each IBM Connect:Direct node. It is not defined as a CICS file. It holds submit (ESF) requests when IBM Connect:Direct DTF is down. It is used at the subtask level by the IBM Connect:Direct API only. It is allocated by CICS and opened by a subtask. If necessary, you can dynamically allocate and deallocate it using the CICS transaction DGAN.
RPLERRCK (DDNAME)	It is not defined as a CICS file nor is it in the RDO. It contains VTAM errors written to a sequential file by the IBM Connect:Direct interface and is defined as SYSOUT.

Data	Description
EVENT RESTART (DDN=NDMEVNT)	One file exists for each CICS region. It is defined as a CICS file, and is allocated and opened by CICS. It is updated by the Event Services Support, and is used for restarting ESS.
Trace Files	<p>Several files within the IBM Connect:Direct API are dedicated to system trace functions, but are not explicitly used by the CICS Interface. The IBM Connect:Direct API design enables trace data to be written when they are available. To capture trace data, add the appropriate DD name statements to the CICS startup JCL. The files must be present at CICS initialization.</p> <p>Note: Do not dynamically allocate trace files after CICS initializes. Each IBM Connect:Direct API (one per WST) writes trace data to these files when present. An example is NDMCMDS, which you can use to view all commands submitted to the API.</p>

IBM Connect:Direct CICS Data Sets

The following data sets, used only by Connect:Direct for z/OS, are for the CICS Interface environment and are updated under CICS:

Data Set	Description
CONFIGURATION	One file exists for each CICS region. It is defined as a CICS file. You can update it online through administrator functions. It is a VSAM KSDS file. It contains all IBM Connect:Direct nodes available to this Connect:Direct for z/OS environment as does NETMAP, but CONFIGURATION contains system parameters that control the IBM Connect:Direct CICS API environment.
USER PROFILE (SIGNON DEFAULTS)	One file exists for each CICS region. It is defined as a CICS file that is a VSAM KSDS file with the CICS userid as key. Update it using the signon defaults function and use it to set up auto-signon to IBM Connect:Direct.

Temporary Data Set

The IBM Connect:Direct API TDEXIT parameter configures the routing of IBM Connect:Direct statistics records to CICS Temporary Storage. CICS Temporary Storage, containing the data, is retained until the user exits the SELECT STATISTICS screen. CICS Temporary Storage containing the data is then deleted.

Signing On and Off CICS

Signing On

About this task

You must install both CICS and IBM Connect:Direct and they must be working on your mainframe. You must also be using an IBM 3270 terminal or equivalent. In order to use the CICS Interface, you must first sign on. The SIGNON screen is the first screen displayed, unless you are authorized for auto-signon or auto-resignon. Sign on as follows:

Procedure

1. Sign on to CICS using the CESN transaction (if needed).

2. Sign on to Connect:Direct for z/OS, using the DGA transaction.
3. Complete the prompted signon information.

The IBM Connect:Direct for z/OS SIGNON screen shows prompts for your userid, password, and node name. Following is an example.

```

                                IBM Connect:Direct
for z/OS                        10:13:12                        SIGNON

USER ID   ==> -----
PASSWORD ==> -----
NODE NAME ==> -----
Do you want all commands for this session to be CASE sensitive? ==>  NO
Licensed Materials - Property of IBM
IBM(R) IBM Connect:Direct(R) for z/OS(R)
(C) Copyright IBM Corp, 1982, 2011 All rights reserved.
IBM is a Trademark of International Business Machines

PF keys:  1  Help      3  Exit

```

4. Type the userid, password, and node name information on this screen. After you have successfully signed on, you see the PRIMARY MENU screen.

The following table describes the entry fields for the IBM Connect:Direct for z/OS SIGNON screen:

Field	Description
USER ID	Following the USERID prompt, type in your userid. The 64-character field contains the acceptable Connect:Direct for z/OS password for the CICS Interface. The CICS Interface administrator has an option to force your IBM Connect:Direct userid to be the same as the CICS userid. This option supports security systems designed to use a single ID per user. When this option is set, the CICS Interface fills the signon screen with the CICS userid. The signon USERID field is then protected from update.
PASSWORD	Following the PASSWORD prompt, type in your password. The 64-character field contains the acceptable Connect:Direct for z/OS password for the CICS interface. The field is darkened to maintain security.
NODE NAME	Following the NODE NAME prompt, type in the name of the IBM Connect:Direct node to which you want to sign on. Include the name of the 16-character field of an active node.
Do you want all commands for this session to be CASE Sensitive?	Following this question, indicate whether you want to allow mixed case input. This option is available as a session default, and you can specify the option during signon. You can override the specified default on commands that apply to userid, password, and data set name. When you submit commands and specify YES, IBM Connect:Direct includes the CASE=YES parameter with your command. Note: Terminals used for the IBM Connect:Direct CICS interface must be defined with UCTRAN(NO) if mixed case data is required.

Auto-Signon

If you are authorized for auto-signon, the SIGNON screen is displayed with an IN PROGRESS message. IBM Connect:Direct uses information stored in your user profile to complete the signon process. If you are not authorized for auto-signon or your user profile does not contain signon information, you must sign on to IBM Connect:Direct manually.

Note: IBM Connect:Direct uses the CICS userid key to access profile information even when the CICS userid does not equal the IBM Connect:Direct userid.

Auto-ReSignon

If configured for resignton, the CICS Interface automatically resigns on those returning to CICS, after exiting CICS to use other CICS transactions. Resignton is canceled when you sign off CICS.

Note: When you attempt to reenter the CICS IUI, your userid and password are reverified by the DTF.

Status Alert Screen

If you attempt to use the CICS Interface without first signing on to CICS, you see the Connect:Direct STATUS ALERT screen.

The STATUS ALERT MESSAGE displayed near the center of the screen indicates a failure of your SIGNON attempt. If you encounter this screen, press PF3 to go to a blank screen with the message DGA TRANSACTION ENDED in the upper left corner. At this point, sign on to CICS and then type in the DGA transaction, followed by Enter to go to the PRIMARY MENU.

If the signon transaction at your site is defined with CICS security or with RACF, CA-ACF2, or CA-TOP SECRET security, messages are displayed from the appropriate security facilities instead of the STATUS ALERT screen messages.

If the CICS Interface is not active, the STATUS ALERT screen is displayed with a message indicating that the interface is not active. In this case, you must activate the CICS Interface using the DGAA transaction. In addition, if your node is not active to the CICS Interface, either through the interface or through the node, you cannot sign on to that node. Ask your administrator to activate your session, your interface, or your node. See the IBM Connect:Direct for z/OS Administration Guide.

Using the Primary Menu

The IBM Connect:Direct for z/OS PRIMARY MENU is the root menu of the menu hierarchy and is the access key to all other features of the CICS Interface.

The PRIMARY MENU contains a list of authorized IUI functions based on your IBM Connect:Direct function authorization. Your administrator can globally restrict these IUI functions in the system Configuration file, so some options may not be available to you. In addition, under ESF operation, the options are restricted to submit and utility options only. Following is an example of the PRIMARY MENU.

```

                                IBM Connect:Direct for z/OS
                                PRIMARY MENU                                15:52:53

node.name
OPTION ==>  __

                                CF  COPY FILE
                                SB  SUBMIT PROCESS

                                SP  SELECT PROCESS
                                SS  SELECT STATISTICS

                                MD  MESSAGE DISPLAY
                                SD  SIGNON DEFAULTS
                                SN  CHANGE SIGNON

Licensed Materials - property of IBM
IBM Connect:Direct for z/OS
(C) Copyright IBM Corp, 1982, 2011 All Rights Reserved.
IBM is a Trademark of International Business Machines

SAFA000I - Connect:Direct signon process completed.
PF keys:  1 Help   2 Msg   3 Exit   6 Id
```

The 2-character OPTION field contains your option selection as follows.

Option	Screen Accessed
CF	COPY FILE BETWEEN NODES

Option	Screen Accessed
SB	SUBMIT PROCESS
SP	SELECT PROCESS
SS	SELECT STATISTICS
MD	MESSAGE DISPLAY
SD	SIGNON DEFAULTS
SN	SIGNON

Press Enter after making your selection. Options CF, SB, SP, SS, and SD are described in other sections that follow. Options SD and SN are described in this section.

Using the SN Option for Multiple Terminal Signon

The signon SN option also displays the signon screen. You can sign on to a IBM Connect:Direct node using signon information other than that specified in your user profile. You can sign on as you want, without updating your user profile.

By using the SN signon option, you can sign on to multiple IBM Connect:Direct nodes, by signing on to multiple CICS terminals. This is a convenience, not a multiple signon.

If you sign on to multiple terminals, only the latest DTF signon authorization rules apply. The latest authorization rules are used for all terminals where you previously signed on.

Updating Your Signon Defaults

About this task

You can use the CICS IUI Signon Defaults (SD) menu option to type your signon defaults, which are stored as part of your user profile and used for subsequent CICS IUI access. The administrator can also assign your user profile information. If you are authorized for auto-signon, you can bypass the IBM Connect:Direct for z/OS SIGNON screen by providing signon defaults.

Note: User profile information is keyed by CICS userid.

Use the SIGNON DEFAULTS screen to update your IBM Connect:Direct password, default node, ESF mode allowance, uppercase printing, CICS session printer, and sending and receiving node accounting information.

Procedure

1. To access the SIGNON DEFAULTS screen, select option SD on the PRIMARY MENU and press Enter. Following is an example.

SIGNON DEFAULTS		11:51:48 node.name
CICS USERID	==>	-----
CONNECT:DIRECT		
USERID	==>	XXXXX1
PASSWORD	==>	
DEFAULT NODE	==>	NODE.NAME
ESF MODE ALLOWED	==>	Y Y OR N
UPPER CASE PRINT	==>	Y Y OR N
CICS PRINTER	==>	LPT1
PNODE ACCT DATA	==>	-----
SNODE ACCT DATA	==>	-----
Do you want all commands for this session to be CASE sensitive? ==> NO		
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id 9 Update Enter Edit Clear Reset		

2. To display your current information, type your IBM Connect:Direct userid and password and press Enter.
3. To change information, use the ARROW, TAB, and RETURN keys to move the cursor to the field you want to change. Press PF9 to update the file.

The following table describes the fields on this screen:

Field	Description
CICS USERID	This 8-character system field contains your CICS userid. If you have a CICS userid that is shared, then your update to the signon defaults affect all users of that userid.
CONNECT:DIRECT USERID	This 64-character field contains your IBM Connect:Direct userid. This field is required.
CONNECT:DIRECT PASSWORD	This 64-character field contains your IBM Connect:Direct password for the IBM Connect:Direct session. The field is darkened to maintain security.
DEFAULT NODE	This 16-character field contains the name of your default node.
UPPER CASE PRINT	This 1-character field contains the Y or N toggle to force uppercase printing on the CICS printer. Y means that all of the printed reports and summary table printouts from this user are in uppercase.
CICS PRINTER	This 4-character field contains the designation for the printer connected to your environment.
PNODE ACCT DATA	This 50-character field contains your accounting data for the primary node. You can use it for your cost billing for CPU, I/O, and system resource time, materials, and personnel.
SNODE ACCT DATA	This 50-character field contains your accounting data for the secondary node. You can use it for your cost billing for CPU, I/O, and system resource time, materials, and personnel.

Using the ESF Session Mode Option

ESF MODE enables you to continue to submit processes when an active Connect:Direct for z/OS DTF session fails or when session establishment fails. The ESF MODE interface differs from the primary node in that the PRIMARY MENU only displays the SUBMIT options. You must install the ESF option on the local DTF and you must enable the ESF option in the Configuration file.

Note: In ESF MODE, the local node is available for use only through SUBMIT commands.

ESF SUBMIT requests can only be issued by those who have the ESF specified as a profile (SIGNON DEFAULTS) option.

Note: This option is restricted to the local node. The administrator can globally disallow it.

ESF MODE operation is toggled on for the entire Connect:Direct for z/OS interface through an administrator parameter. ESF MODE operation is also toggled on for each user through a user profile parameter.

For the non-terminal user of the interface, the switch to primary mode (DTF MODE) is attempted until that task completes. This limitation prevents having to abort the non-terminal processing (which could not continue in primary mode without another signon).

ESF Session Signon and Notification

After you sign on to ESF mode, Connect:Direct for z/OS notifies users that the session is in ESF mode in the following ways:

- A message is displayed on all user screens that the session is in ESF mode
- When a node session switches from ESF to primary mode, or from primary mode to ESF mode, all terminal users of that node are notified on their next IUI access

DTF Notification

Connect:Direct for z/OS periodically attempts to establish or reestablish a failed DTF session. When ESF mode switches back to primary mode, Connect:Direct for z/OS prompts you for primary mode signon.

Connect:Direct for z/OS displays a special screen informing IUI users that the DTF has become active or inactive. This screen gives you the option either to exit Connect:Direct for z/OS or to resignon in ESF mode.

Signing Off

About this task

The signoff sequence is as follows:

Procedure

1. Press PF3 repeatedly until you reach the PRIMARY MENU or press PF4 once from your current screen to go to the PRIMARY MENU.
2. Press PF3 again, and you see a blank screen with the message -- DGA TRANSACTION ENDED -- in the upper left corner.
3. Type CESF LOGOFF and press Enter.

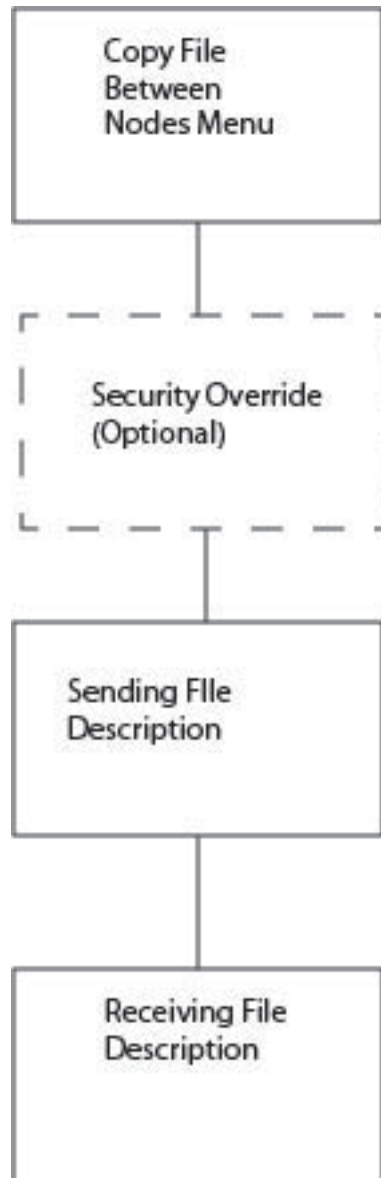
About Copying Files

The Copy File menus are a series of panels that collect information used for copying files between nodes. You can use the following four screens to build a COPY Process:

Screen Name	Description
COPY FILE BETWEEN NODES Menu	Specifies the sending and receiving nodes of the COPY and the submission parameters for your COPY Process.
Security Override Menu	Collects information about userid, password, and accounting data. This optional panel is displayed only if you request it on the first panel.

Screen Name	Description
SENDING FILE Menu	Collects information for the Process variables for the sending file.
RECEIVING FILE Menu	Collects the information for the Process variables for the receiving file.

The relationship of these four panels is illustrated in the following diagram.



Requirements

To copy a file between nodes the following must be true:

- You must be authorized to perform COPY Processes on both nodes
- You must have access to the appropriate files on both nodes
- Both nodes must be active
- All interfaces must be started

Generating a Copy File from Connect:Direct for CICS

About this task

In the following procedure, you can skip Step 4 which involves the Security Override screen, if you do not enter Y in the OVERRIDE SECURITY field on the second Copy File Between Nodes screen.

To obtain Help on the field content of any screen, press the PF1 key.

Procedure

1. To access the COPY FILE BETWEEN NODES first screen, select option **CF** on the PRIMARY MENU and press ENTER.

COPY FILE BETWEEN NODES				15:34:14
			node.name	
	NODE NAME	DESCRIPTION		
01	MSDOS.NODE	- MS-DOS NODE		
02	MVS.NODE	- MVS NODE		
03	NETWARE.NODE	- NETWARE NODE		
04	OS2.NODE	- OS/2 NODE		
05	OS400.NODE	- OS/400 NODE		
06	VSE.NODE	- VSE DTF		
07	TANDEM.NODE	- TANDEM NODE		
08	UNIX.NODE	- UNIX NODE		
09	VM.NODE	- VM NODE		
10	VMS.NODE	- VMS NODE		
11	OS390.NODE1	- Z/OS NODE		
12	OS390.NODE2	- Z/OS NODE		

SENDING NODE NUMBER => --	or NODE NAME=> -----	ENV=> -----
RECEIVING NODE NUMBER=> --	or NODE NAME=> -----	ENV=> -----
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id 7 Bwd 8 Fwd		

The top half of the COPY FILE screen presents a scrollable list of all the IBM Connect:Direct nodes that can participate in a COPY FILE Process (as defined in the CONFIGURATION file by the administrator). The list consists of a symbolic node name, a descriptive node name, and a node selection number.

2. Specify the copy file sending and receiving nodes by selecting the corresponding node selection numbers. If you do not see the node that you need, press PF8 to scroll the data forward in the list and press PF7 to scroll the data backward.

When specifying a sending or a receiving node, you have two options:

- You can use the number from the scrollable list
- You can use the node name and the environment

Note: If you do not provide this information, the default is the node that you are currently signed on to.

The ENV fields are updated automatically when you type the node number or node name from the scrollable list, but you must fill in the ENV field manually if the node name is not in the list.

The current node (as determined from the Network Map) is highlighted in the list and you must select it as the sending or receiving default node. You can save the node choice numbers and display them again the next time you enter this screen.

If the COPYFILE node is not listed, you must type both the copy node names and environment types.

The nodes in the scrollable list must be added using the DGAA transaction.

The following table defines each entry field on the Copy File Between Nodes panel.

Entry Field	Description
SENDING NODE NUMBER	This 2-character field contains the sending node number. In the scrollable field to the left of the node name field are listed the associated node numbers. If you specify neither, the sending node defaults to the node you are signed on to. When you identify the sending node and press ENTER, the environment field is filled with the appropriate sending node.
NODE NAME	This 16-character field contains the sending node name.
ENV	This 8-character field contains the sending environment. If you do not select from the list, you must specify the environment. Valid sending environments for Connect:Direct for CICS are OS/390, VSE, VM, VMS, Tandem, OS/400, UNIX, and Windows (for Windows NT).
RECEIVING NODE NUMBER	This 2-character field contains the receiving node number. In the scrollable table to the left of the node name field the associated node numbers are listed. If you specify neither, the receiving node defaults to the node that you are signed on to. When you identify the receiving node and press ENTER, the environment field is filled with the appropriate receiving node.
NODE NAME	This 16-character field contains the receiving node name.
ENV	This 8-character field names the receiving environment. Valid receiving environments for Connect:Direct for CICS are OS/390, VSE, VM, VMS, Tandem, OS/400, UNIX, and Windows (for Windows NT).

After the successful submission of the COPY FILE Process, this first screen in the panel series is redisplayed and a PROCESS NUMBER message is displayed. The PROCESS NUMBER message contains the number assigned to your Process by IBM Connect:Direct.

- After you provide the sending and receiving node names, press ENTER.

```

                                COPY FILE BETWEEN NODES                                15:36:47
                                                                node.name

SENDING NODE NAME => OS390.NODE1      ENV=> OS390
RECEIVING NODE NAME=> OS390.NODE2    ENV=> OS390

NEW PROCESS NAME=> COPYCF__
CLASS              => --- (NUMERIC)
HOLD               => N   (Y, N, or C-Call)
PRIORITY           => --- (RANGE: 0 to 15)
REQUEUE           => N   (Y or N)
RETAIN on TCQ      => N   (Y, N, OR I-Initial)
START DATE         => ----- START TIME=> ----- (HH:MM:SSXM)
CHECKPOINT         => ----- (BYTE INTERVAL - nK|nM)
PLEXCLASS          => ----- (PNODE SNODE)
COMPRESS           => N---- (Y, N, E-Extended, X'xx', or C'c')
OVERRIDE SECURITY=> N---- (Y or N)

Do you want values for this copy to be CASE sensitive? ==> NO

PF keys:  1 Help   2 Msg   3 Exit  4 Menu  6 Id

```

On the second COPY FILE BETWEEN NODES screen you can specify the parameters of the copy. The following table describes the system fields for this screen which contain information specified on the first COPY FILE BETWEEN NODES screen.

System Fields	Description
SENDING NODE NAME	This dual field contains the sending node name and environment that you provided in the previous screen.

System Fields	Description
RECEIVING NODE NAME	This dual field contains the receiving node name and environment that you provided in the previous screen.

The following table describes each of the parameters you can specify for the copy.

Parameter	Description
NEW PROCESS NAME	This 8-character field contains your selection for a new unique Process name to be assigned to the COPY operation.
HOLD	This 1-character field contains the Y, N, or C toggle for a Process to be kept in the HOLD queue. This field is optional. Y (yes) holds the Process until it is deleted or released N (no) does not hold the Process (default) C (call) holds the Process in the Wait queue until the secondary node requests work. If the Process consumes computer resources during periods of heavy system use, you can place the Process temporarily in the Hold queue, and release it during a time of infrequent use.
PRIORITY	This 2-character field contains the priority number for Process execution. This field is optional. Valid priority numbers range from 0-15 (highest).
REQUEUE	This optional field specifies whether a copy step requeues if the Process terminates abnormally.
RETAIN on TCQ	This 1-character field contains the Y, N, or I toggle to keep a copy of a Process in a queue after execution. Filling this field is optional. Y (yes) keeps the Process in the queue after execution N (no) deletes the Process after execution I (initialize) schedules the Process for execution every time IBM Connect:Direct is initialized
START DATE	This field contains the starting date for the copy operation expressed as month, day, and year. If you leave this field blank, the Process takes place on the current date. Fill in this field if you want to start the Process at some future time.
START TIME	<p>This 10-character field contains the starting time for the copy operation expressed as hours, minutes, and seconds, AM or PM (HH:MM:SSXM). The default is a 24-hour clock. If a 12-hour clock is used, the AM or PM is required.</p> <p>The starting time works in conjunction with the starting date, but the field is optional. If you do not specify a time and the date is the current day, the Process begins immediately.</p> <p>If the date is set for a future date and the START TIME is not specified, the Process begins at 12:00 AM. The START TIME field is optional. Use this field if you want to start the Process at some future time.</p>
CHECKPOINT	The CHECKPOINT field specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid transmission point. This feature avoids the need to restart transmission from the beginning. K denotes thousands; M denotes millions. A checkpoint value of zero stops automatic checkpointing.

Parameter	Description
PLEXCLASS	The PLEXCLASS field specifies the class that directs the Process to only certain servers in a IBM Connect:Direct/Plex. Only use this parameter in a IBM Connect:Direct/Plex. Each server in a IBM Connect:Direct/Plex can be designated to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. Processes can then be limited to only those servers by specifying the PLEXCLASS in the Process definition. The PNODE class controls which IBM Connect:Direct/Server runs the Process. The SNODE class controls what other node is used with the Process. The PNODE class and SNODE class are each 1–8 characters long. Use an asterisk (*) to indicate that the Process can run on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If no PLEXCLASS is specified, the Process runs on any IBM Connect:Direct/Server that supports PLEXCLASS. If a Process must run on a specific IBM Connect:Direct/Server, specify the IBM Connect:Direct/Server name in this field. The Process runs only on that server.
COMPRESS	The COMPRESS field specifies that IBM Connect:Direct is to compress the data. This feature reduces the amount of data transmitted as the file copies from one node to another. IBM Connect:Direct automatically decompresses the file at its destination. The default subparameter for the COMPRESS parameter is PRIMEchar=X'40'.
OVERRIDE SECURITY	If you want to type security information such as userid and password, type a Y at the OVERRIDE SECURITY prompt. IBM Connect:Direct displays the SECURITY OVERRIDE screen. (See the next step.)
CASE Sensitive	Following this question, indicate whether you want to allow mixed case input. This option is available as a session default, and you can specify the option during signon. You can override the specified default on commands that apply to userid, password, and data set name. When you submit commands with YES specified, IBM Connect:Direct includes the CASE=YES parameter with your command. CICS only interprets mixed case data if your terminal is defined to accept it. The CICS RDO or CEDA definition must be defined with UCTRAN=NO for mixed case data to be input to IBM Connect:Direct from a CICS terminal.

4. If you typed Y in the Override Security prompt on the COPY FILE BETWEEN NODES screen, IBM Connect:Direct the optional SECURITY OVERRIDE screen is displayed.

COPY FILE BETWEEN NODES		17:28:36
		node.name
SECURITY OVERRIDE		
SENDING NODE NAME	=> OS390.NODE1	ENV => OS390
SECURITY ID:	-----	
PASSWORD :	-----	
NEW PASSWORD:	-----	
ACCT DATA :	-----	
RECEIVING NODE NAME	=> OS390.NODE2	ENV => OS390
SECURITY ID:	-----	
PASSWORD :	-----	
NEW PASSWORD:	-----	
ACCT DATA :	-----	
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id		

The screen contains the same entry fields for both the sending and receiving nodes. The following table describes each entry field.

Entry Field	Description
SECURITY USERID	This optional field specifies the 1-64 character security ID that passes to a security exit.
PASSWORD	This optional field specifies the 1-64 character current security password to pass to a security exit.
NEW PASSWORD	This optional field specifies the new 1-64 character security password to be passed to a security exit. The exit can change the current password to this value.
ACCT DATA	This optional field specifies accounting data to be passed to the security exit.

5. The values selected in the Sending and Receiving Node Name and Environment fields on the first screen determine which Sending and Receiving screens display to complete your COPY statement. The following example shows the Sending File screen for z/OS. (The valid Environment values for sending COPY files are: OS/390, OS/400, WIN95, VM, VMS, VOS, VSE, TANDEM, and UNIX.)

```

                                COPYFILE - SENDING FILE (OS/390)                                17:28:36
                                                                node.name

SENDING NODE:    ...=>  OS390.NODE1

SENDING DSNAME   =>  _____
UNIT            =>  ( _____ )
VOLUME          =>  ( _____ )

SYSOPTS         =>  " _____
                    _____
                    _____
                    _____"

PDS ONLY:
SELECTION CRITERIA =>  _____
                    =>  _____
                    =>  _____
                    =>  _____

REPLACE         =>  N (Y OR N)

PF keys:  1 Help   2 Msg   3 Exit   4 Menu   6 Id

```

Note: For a complete description of the valid parameters of a COPY Statement and examples, see *IBM Connect:Direct for z/OS Process Language Reference Guide*.

- Fill in the appropriate values for the SENDING FILE screen.
 - When you finish typing the values on the SENDING FILE screen, press PF3 to proceed to the RECEIVING FILE screen.
6. The values selected for the Receiving Node Name and Environment fields on the first screen determine which Receiving screen is displayed to complete your COPY statement. The following example shows the Receiving File screen for z/OS.

COPYFILE - RECEIVING FILE (0S/390)		17:28:36
		node.name
RECEIVING NODE	=> 0S390.NODE2	
RECEIVING DSNAME	=> _____	
DISP	=> (NEW , CATLG)	
UNIT	=> (_____)	
VOLUME	=> (_____)	
DCB	=> (_____)	
LABEL	=> (_____)	
SPACE	=> (_____)	
TYPEKEY	=> _____	
SYSOPTS	=> " _____ _____ _____ _____ "	
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id		

- Fill in the appropriate values for the RECEIVING FILE screen.
- When you finish typing the values on the RECEIVING FILE screen, press PF3 to process the copyfile request and exit.

Copying a File from Your Node to Your Node

Use Connect:Direct for z/OS CICS to copy a file from your node back to your node, by performing the following procedure:

Procedure

- Select option CF from the PRIMARY MENU and press ENTER to display the COPY FILE BETWEEN NODES screen.
- Note:** You are signed on to a z/OS node for this example. Ensure that your node name is in the node list at the top of the screen. If it does not display, ask the administrator to start your node on the network. Alternatively, you can type in your node number, node name, and environment directly.
- Type in your node number in the SENDING NODE NUMBER field.
- Type in your node number in the RECEIVING NODE NUMBER field and press ENTER.
- Ensure that all your entries are correct and press ENTER.
- Assuming the sending file is cataloged, on the COPYFILE - SENDING FILE (z/OS) screen, type in the SENDING DSNAME and press ENTER.
- On the COPYFILE - RECEIVING FILE (z/OS) screen, type in the RECEIVING DSNAME and press ENTER.
- When the Process is submitted to IBM Connect:Direct, the COPY FILE BETWEEN NODES screen with the PROCESS NUMBER message is displayed. This number indicates the number assigned to your Process.
- Press PF4 to go to the PRIMARY MENU.
- Select option SP to go to the SELECT PROCESS screen to check on the status of your Process.
- On the SELECT PROCESS screen select option O for OPERATOR TABLE, select A for all queues, type in the PROCESS NUMBER you observed from the COPY FILE BETWEEN NODES screen, and press ENTER to display the status.

Building, Modifying, and Submitting Processes through CICS

Submit Process Screen

Use the SUBMIT PROCESS screen to execute a Process by identifying the Process name, secondary node, times and dates, priority, requeue, class, hold and retain status, and symbolic parameters. The Process is located in the local node IBM Connect:Direct Process library.

You can specify symbolic parameters before you submit the Process to the IBM Connect:Direct DTF for execution. The IUI then submits the IBM Connect:Direct COPY Process to the connected IBM Connect:Direct DTF (the DTF specified on the Connect:Direct for z/OS CICS signon). A PROCESS SUBMITTED message is returned after the Process is successfully submitted to the DTF.

To access the SUBMIT PROCESS screen, select option SB on the PRIMARY MENU screen and press Enter. Following is an example.

```

                                SUBMIT PROCESS                                11:24:50
                                                                node.name

PROCESS NAME      ==>  -----
SECONDARY NODE    ==>  -----
HOLD PROCESS      ==>  ----- (Y=YES, N=NO, C=CALL)
REQUEUE PROCESS   ==>  ----- (Y=YES, N=NO)
RETAIN PROCESS    ==>  ----- (Y=YES, N=NO, I=INIT)
PRIORITY          ==>  --
CLASS             ==>  ---
START DATE        ==>  ----- TIME ==>  -----
NEW PROCESS NAME  ==>  -----
PLEXCLASS         ==>  ----- (PNODE SNODE)
SYMBOLIC PARAMETERS ==>  -----
-----
OVERRIDE SECURITY=>  N          (Y or N)

Do you want values for this process to be CASE sensitive? ==>  NO

PF keys:  1 Help   2 Msg   3 Exit  4 Menu  6 Id
```

The following table describes the Entry fields:

Field	Description
PROCESS NAME	8-character field contains the name of the Process to be submitted. Field is required.
SECONDARY NODE	This 16-character field contains the name of the secondary node (destination node) to which the Process is to be submitted. This field is optional. The value defaults to the SNODE specified in the Process.
HOLD PROCESS	This 1-character field contains the Y, N, or C toggle for a Process to be kept in the HOLD queue. This field is optional. <ul style="list-style-type: none">• Y (yes) holds the Process until it is deleted or released.• N (no) does not hold the Process. This value is the default.• C (call) holds the Process in the Wait queue until the secondary node requests work. If the Process consumes computer resources during periods of heavy system use, you can place the Process temporarily in the Hold queue, and release it during a time of infrequent use.
REQUEUE PROCESS	This optional field specifies whether a copy step requeues if the Process terminates abnormally.

Field	Description
RETAIN PROCESS	<p>This 1-character field contains the Y, N, or I toggle to keep a copy of a Process in a queue after execution. This field is optional.</p> <ul style="list-style-type: none"> • Y (yes) keeps the Process in the queue after execution. • N (no) deletes the Process after execution. • I (initialize) schedules the Process for execution every time IBM Connect:Direct is initialized.
PRIORITY	<p>This 2-character field contains the priority number for Process execution. This field is optional. Valid priority numbers range from 0–15 (highest).</p>
CLASS	<p>This 3-character field contains the Process class assignment for a submitted Process. Filling the field is optional. Acceptable values range from 1–255.</p>
TIME	<p>This 10-character field contains the start time expressed as hours, minutes, and seconds AM or PM (HH:MM:SSXM). This field is optional. Enter the start time on the basis of a 12- or 24-hour clock. If AM or PM is not used, the default is a 24-hour clock (2:00 PM or 14:00).</p>
NEW PROCESS NAME	<p>8-character field contains a new name for a submitted Process. This field is optional.</p>
PLEXCLASS = (pnode class, snode class)	<p>Specifies the class that directs the Process to only certain servers in a IBM Connect:Direct/Plex. This parameter is only used in a IBM Connect:Direct/Plex.</p> <p>Each server in a IBM Connect:Direct/Plex can be designated to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. Processes can then be limited to only those servers by specifying the PLEXCLASS in the Process definition.</p> <p>The PNODE class controls which IBM Connect:Direct/Server runs the Process. The SNODE class controls what other node is used with the Process.</p> <p>The PNODE class and SNODE class are each 1-8 characters long. Use an asterisk (*) to indicate that the Process can run on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If no PLEXCLASS is specified, the Process runs on any IBM Connect:Direct/Server that supports PLEXCLASS.</p> <p>If a Process must run on a specific IBM Connect:Direct/Server, specify the IBM Connect:Direct/Server name in this field. The Process runs only on that server.</p>
SYMBOLIC PARAMETERS	<p>This 2-line field contains the symbolic parameters that you want substituted for the items in the SUBMIT PROCESS operation. This field is optional. Enter the symbolics as follows: &PARAMETER=SUBSTITUTION, and so forth. You can specify as many symbolic parameters as fit on the lines provided. Each must be separated by at least one space.</p>
SECURITY OVERRIDE	<p>This optional field is for PNODE and SNODE security checking. Type a Y or N. You can check or modify security information on PNODE or SNODE on the Submit Process screen. Refer to Security Override Screen.</p>

Field	Description
CASE Sensitive	Following this question, indicate whether you want to allow mixed case input. This option is available as a session default, and you can specify the option during signon. You can override the specified default on commands that apply to userid, password, and data set name. When you submit commands with YES specified, IBM Connect:Direct includes the CASE=YES parameter with your command. Note: CICS only interprets mixed case data if your terminal is defined to accept it. The CICS TCT MACRO or RDO TYPETERM definition must be defined with UCTRAN=NO for mixed case data to be input to IBM Connect:Direct from a CICS terminal.

The following table describes the one system field which is displayed after you successfully submit a Process.

Field	Description
PROCESS NUMBER number	This 21-character field contains the PROCESS NUMBER number message after the successful completion of a SUBMIT PROCESS task, where number is the system-generated Process number.

Security Override Screen

The security override panel provides the opportunity to check or modify PNODE and SNODE security. This panel is displayed when you specify yes for the OVERRIDE SECURITY prompt of the SUBMIT PROCESS screen.

SUBMIT PROCESS		11:38:25
		node.name
SECURITY OVERRIDE		
<<PNODE>>		
SECURITY ID:	-----	
PASSWORD :	-----	
NEW PASSWRD:	-----	
ACCT DATA :	-----	
<<SNODE>>		
SECURITY ID:	-----	
PASSWORD :	-----	
NEW PASSWRD:	-----	
ACCT DATA :	-----	
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id		

The SUBMIT PROCESS SECURITY OVERRIDE screen contains the following entry fields, one set for the PNODE and one for the SNODE:

Field	Description
SECURITY USERID	This optional field specifies the 1–64 character security ID that passes to a security exit.
PASSWORD	This optional field specifies the 1–64 character current security password to pass to a security exit.
NEW PASSWORD	This optional field specifies the new 1–64 character security password to be passed to a security exit. The exit changes the current password to this value.
ACCT DATA	This optional field specifies accounting data to be passed to the security exit.

Selecting a Process through CICS

Determining the Status of a Submitted Process

The Select Process and Select Process - Operator Table screens allow you to view information about the Processes you have submitted.

Procedure

1. To access the SELECT PROCESS screen to view information about your Processes, type **SP** on the PRIMARY MENU and press Enter.

Following is an example of the SELECT PROCESS screen.

```

                                SELECT PROCESS                                11:39:56
                                                                node.name

OPTION ==> 0          (0 - OPERATOR TABLE   P - PRINT REPORT)

QUEUE  ==> _  (A-ALL,W-WAIT,E-EXECUTE,H-HOLD,T-TIMER)

PROCESS NUMBERS:  ==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:   ==> _____ ==> _____ ==> _____ ==> _____
SERVER NAMES:    ==> _____ ==> _____ ==> _____ ==> _____

STATUS:          (HO,HR,HI,HE,HC,HP,HS,RH,RA,WC,H,R,W)
==> __          ==> __          ==> __          ==> __

DESTINATION NODES:
==> _____ ==> _____
==> _____ ==> _____

USER ID:         NODE ID:
==>.node1       ==> node.name

==> _____
==> _____

Do you want values for this command to be CASE sensitive? ==> NO

PF keys:  1 Help   2 Msg   3 Exit  4 Menu  6 Id
```

The following table describes the entry fields on this screen:

Field	Description
QUEUE	This 1-character field contains your selection of the Process queues for display: A (to select all queues), W (for the Wait queue), E (for the execution queue), H (for the Hold queue), or T (for the Timer queue). This field is optional.
PROCESS NUMBERS	These four fields of six characters each contain the numbers of Processes to be selected. The field is optional. If you select nothing, all Processes in the selected queue are displayed.
PROCESS NAMES	These four fields of eight characters each contain the names of Processes to be selected. The field is optional. If you select nothing, all Processes in the selected queue are displayed.
SERVER NAMES	These four fields contain the IBM Connect:Direct/Server names to select Processes from. The server name is a 1–8 character name assigned to each server in a IBM Connect:Direct/Plex through the CDPLEX.SERVER initialization parameter. This field only applies to a IBM Connect:Direct/Plex.

Field	Description
STATUS	<p>This optional field indicates specific queue status selection by use of the following characters:</p> <p>HO, held by operator. The Process was submitted without hold specified and later was changed with the CHANGE PROCESS command.</p> <p>HR, held retain. The Process was submitted with RETAIN=YES specified.</p> <p>HI, held initially. The Process was submitted with HOLD=YES specified.</p> <p>HE, held in error. The Process was submitted, but the submitter is not defined on the SNODE.</p> <p>HC, held for call. The Process was submitted with HOLD=CALL specified. A session started from the other node causes this Process to be put on the wait queue in WC status.</p> <p>HP, held Process error. An error occurred during initiation of Process execution. This condition can occur if the session is lost before any Process Statements begin to execute.</p> <p>HS, held for suspension. The operator issued a SUSPEND PROCESS command. The Process can be released later.</p> <p>RH, restart Held. A checkpointed Process was executing when an error such as a lost session or an I/O error occurred. This enables the copy to be restarted when the session is lost and reestablished.</p> <p>RA, held for restart allocation error. During Process execution, an allocation error occurred matching those specified in the initialization parameters. This status enables the Process to be restarted after the allocation problem is resolved.</p>
STATUS	<p>WC, wait for connection. Session establishment attempted, including retries if specified, and failed. This Process is put on the wait queue and processed if a session with that node is established later. It can also be released by the operator.</p> <p>WT, wait for transport. The transport protocol is not available. The Process runs as soon as the transport protocol is available.</p> <p>WX, wait for server. The Process is waiting for an eligible IBM Connect:Direct/Server to become available. The Process runs as soon as an eligible IBM Connect:Direct/Server is available.</p> <p>An eligible IBM Connect:Direct/Server is an active server that supports the Process PLEXCLASS and the transport protocol (SNA, TCP, UDT, or CTCA). The transport protocol must also be available on the server for it to be eligible.</p> <p>H, all Held Processes. This selection enables you to view a list of all held Processes from the Select Process screen.</p> <p>R, all restarted Processes. This selection enables you to view a list of all restarted Processes from the Select Process screen.</p> <p>W, all waiting Processes. This selection enables you to view a list of all waiting Processes from the Select Process screen.</p>
DESTINATION NODES	This optional field indicates the destination site identifier (NODE NAME).
USER ID	This optional field is an alphanumeric userid corresponding to a selected Process.

Field	Description
NODE ID	This optional field identifies the submitter node corresponding to a selected Process. NODE ID is required if USER ID is specified.
Do you want values for this command to be CASE Sensitive?	<p>Following this question, indicate whether you want to allow mixed case input. This option is available as a session default, and you can specify the option during signon. You can override the specified default on commands that apply to userid, password, and data set name. When you submit commands and specify YES, IBM Connect:Direct includes the CASE=YES parameter with your command.</p> <p>Note: CICS only interprets mixed case data if your terminal is defined to accept it. The CICS RDO TYPETERM definition must be defined with UCTRAN(NO) for mixed case data to be input to IBM Connect:Direct from a CICS terminal.</p>

2. Take one of the following actions in the OPTION field:

- To print a report on your default printer (defined in your signon defaults), type P and press Enter.
- To access the SELECT PROCESS - OPERATOR TABLE screen and create a 1-line summary of each selected Process, specify the entry fields listed above, then type O (alphabetic, not zero) and press ENTER.

Viewing Summary Information

About this task

You can use the SELECT PROCESS - OPERATOR TABLE screen to view summary information about Processes pending or executing, including name, number, sending and receiving nodes, queue type and status, and last message received.

Procedure

1. To access the SELECT PROCESS - OPERATOR TABLE screen, access the SELECT PROCESS screen, fill in the entry fields to specify the Processes, and type O (alphabetic, not zero) and press Enter.

Following is an example of the SELECT PROCESS - OPERATOR TABLE screen.

SELECT PROCESS - OPERATOR TABLE							11:02:05
							node.name
Cmd	ProcName	ProcNum	Submitter Node	Secondary Node	QType	QStat	Last Msgid
—	COPYNC	202	OS390.NODE	MVS.NODE	TIMER	WC	
—	COPYJC	203	OS390.NODE	MVS.NODE	TIMER	WC	
Line commands: M Last Msgid S Select process detail							
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id Enter Refresh							

The following table describes the fields on the SELECT PROCESS - OPERATOR TABLE screen:

Field	Description
PAGE XXXX OF YYYY	This 17-character field contains the current page number XXXX out of YYYY total pages of SELECT PROCESS -- OPERATOR TABLE data.
ProcName	This 9-character field contains the Process name.
ProcNum	This 6-character field contains the Process number.
Submitter Node	This 16-character field contains the node name of the primary node.

Field	Description
Secondary Node	This 16-character field contains the node name of the secondary node.
QType	This 5-character field contains the type of queue: <ul style="list-style-type: none"> • EXEC - Process is currently executing • HOLD - Process is held • WAIT - Process is waiting for execution • TIMER - Process is to execute at a given time
QStat	This 2-character field contains the current status of the queue: <ul style="list-style-type: none"> • EX - currently executing • HC - is held for call • HE - held in error • HI - held initially • HO - held by the operator • HP - an error occurred during initiation of Process execution • HR - submitted with RETAIN=YES • HS - is held for suspension • RA - is held for restart allocation error • RH - is held for restart • RS - is being restarted • WA - is awaiting acknowledgement • WC - is awaiting connection • WR - is awaiting restart • WS - is awaiting the designated start time • WT - is awaiting transport protocol • WX - is awaiting an eligible IBM Connect:Direct/Server
Last Msgid	This 8-character field contains the designation of the last message received for Connect:Direct for z/OS operations.

2. Take one of the following actions in the Cmd column next to the ProcName:

- To display the text of the last message received for this Process, type M and press Enter.
- To display the SELECT PROCESS -- PROCESS DETAIL screen, type S and press Enter.

Viewing Process Detail

You can use the SELECT PROCESS - PROCESS DETAIL screen to view detailed information about pending or executing Processes selectively, by choosing any one of the Process details, such as name, number, queueing priority, schedule time, retain status, I/O bytes, and so forth.

Procedure

1. To access the SELECT PROCESS - PROCESS DETAIL screen, type S and press Enter on the SELECT PROCESS - OPERATOR TABLE screen.

Following is an example.

```

                                SELECT PROCESS - PROCESS DETAIL                                11:21:16
OPTION ==> S (M or S)                                node.name

Options: M - Last Msgid; S - Select process detail                                RECEIVING SIDE
Process Name => SEQ001      Number => 3                                Step => STEP1
Other Node  => SC.SC1.SW00D2                                Status => EX
Commid      => 26221;10.20.129.168                                Queue => EXEC
Function    => COPY        Sub State =>                                State => FILE I/O
Server Name => SDWSERV1    PLEXCLASS => ( *                                )
Submitter   => SC.SC1.SW00D1
Userid      => SYS002
Scheduled Time =>          Date      =>          Day      =>
Queueing Prty => 10        Class      => 1        Retain => NO
Submitted Class=> NONE      Max Class => NONE      Sess.Id=> PNOD
Session restrt => 0        Dyn restrt=> 0        RouteID=>
Last Msgid   =>          Last RC    => 00000000      RetProc=>

Sending File  => CSDQA1.TESTFILE.BENCH.M10
Receiving File => SW00D1.TEMPM10
Volume seq no. => 1        Volser    => USER28      TTRN    => 000B0100
Blks          => 23        Recs       => 0          RUs     =>
I/O bytes     => 642,160
VTAM bytes    => 436,224                                Compression Factor =>

PF keys:  1 Help  2 Msg  3 Exit  4 Menu  6 Id  Enter Refresh

```

The following table describes the fields on the SELECT PROCESS - PROCESS DETAIL screen:

Field	Description
SIDE	This 14-character field contains the message SENDING SIDE if the PROCESS DETAIL parameters are from the sender and the message RECEIVING SIDE if the PROCESS DETAIL parameters are from the receiver. The field occurs in the upper right corner below the NODE field.
Process Name	This 8-character field contains the name of the selected Process.
Number	This 6-character field contains the Process number.
Step	This 8-character field contains the name of the currently executing step of the selected Process.
Other Node	This 16-character field contains the name of the node identified as the destination of the Process.

Field	Description
Status	<p>This 2-character field contains the current status of the queue:</p> <ul style="list-style-type: none"> • EX - currently executing • HC - is held for call • HE - held in error • HI - held initially • HO - held by the operator • HP - an error occurred during initiation of Process execution • HR - submitted with RETAIN=YES • HS - is held for suspension • RA - is held for restart allocation error • RH - is held for restart • RS - is being restarted • WA - is awaiting acknowledgement • WC - is awaiting connection • WR - is awaiting restart • WS - is awaiting the designated start time • WT - is awaiting transport protocol • WX - is awaiting an eligible IBM Connect:Direct/Server
Commid	<p>This 46-character field contains the SNA VTAM APPLID (application identification) of the destination IBM Connect:Direct node. If the destination node is TCP/IP, this field contains the port and IP address of the remote node in the format port_number;IP_address.</p>
Queue	<p>This 5-character field contains the name of the queue where the selected Process resides: WAIT, TIMER, EXEC, or HOLD.</p>
Function	<p>This 8-character field contains the name of the function being performed by the Process.</p>
Sub State	<p>This 12-character field contains the name of the current session macro in execution.</p>
State	<p>This 8-character field contains the current state of the Process.</p>
Server Name	<p>The name of the IBM Connect:Direct/Server where the Process runs. This field only applies to a IBM Connect:Direct/Plex.</p>
PLEXCLASS	<p>The class that directs the Process to only certain servers in a IBM Connect:Direct/Plex. This field only applies to a IBM Connect:Direct/Plex.</p>
Submitter	<p>This 16-character field contains the name of the submitting node.</p>
Userid	<p>This 8-character field contains the userid that submitted the Process.</p>
Scheduled Time	<p>This 8-character field contains the time when the Process is scheduled to execute.</p>
Date	<p>This 8-character field contains the date when the Process is scheduled to execute.</p>
Day	<p>This 10-character field contains the name of the day when the Process is scheduled to execute.</p>

Field	Description
Queueing Prty	This 3-character field contains the queueing priority of the selected Process. The range of priorities is 0–15 (highest).
Class	This 4-character field contains the class or parallel session used.
Retain	This 4-character field contains YES or NO to indicate the retain status.
Submitted Class	This 4-character field contains the submitted class.
Max Class	This 4-character field contains the maximum number of parallel sessions allowed.
Sess. Id	This 4-character field contains the session ID (P for SNA primary and S for SNA secondary).
Session restrt	This 3-character field contains the number of consecutive session attempt failures.
Dyn restrt	This 3-character field contains the number of failed attempts to perform a copy to an unavailable file.
RouteID	This 8-character field contains the userid to notify upon completion of a step or Process.
Last Msgid	This 8-character field contains the last message number received by this Process. Selecting option M displays the text of the message.
Last RC	This 8-character field contains the last step return code, if the Process went into execution.
RetProc	The retained Process number.
Sending File	This 44-character field contains the name of the sending file.
Receiving File	This 44-character field contains the name of the receiving file.
Volume seq no.	This 3-character field contains the volume sequence number.
Volser	This 6-character field contains the volume serial number.
TTRN	This 8-character field contains the TTR address (disk) or relative block (tape) currently accessed.
Blks	This 9-character field contains the number of blocks sent or received.
Recs	This 14-character field contains the number of records sent or received as a result of the Process.
RUs	This 9-character field contains the number of request units sent or received.
I/O bytes	This 22-character field contains the number of bytes read or written from external storage.
Member	This 8-character field contains the name of the copied member during a PDS copy operation.
VTAM bytes	This 22-character field contains the number of bytes transmitted.
Compression Factor	This 6-character field contains the percentage of savings from the use of data compression from a copy Process.

2. Take one of the following actions in the OPTION field:

- To display the text of the last message received for this Process, type M and press Enter.
- To refresh the SELECT PROCESS -- PROCESS DETAIL parameters, type S and press Enter.

Selecting Statistics through CICS

Viewing Process Statistics

About this task

Use the SELECT STATISTICS screen to view Process statistics, by choosing any combination of Process names, numbers, start and stop times and dates, and condition codes. Process statistics summarize the IBM Connect:Direct Process execution event log information for a DTF.

Procedure

1. To access the SELECT STATISTICS screen, type **SS** and press ENTER on the PRIMARY MENU. Following is an example.

```

                                SELECT STATISTICS                                11:34:47
                                                                 node.name

OPTION ==> _

PROCESS NUMBERS:
==> -----
PROCESS NAMES:
==> -----

START DATE      ==> -----
START TIME      ==> ----- (HH:MM:SSXM)
STOP DATE       ==> -----
STOP TIME       ==> ----- (HH:MM:SSXM)
CONDITION CODE  ==> -----
EXCLUDE MEMBER RECS ==> N (Y OR N)
EXCLUDE WTO RECS  ==> Y (Y OR N)

OPTIONS: S ... SUMMARY TABLE
         P ... PRINT REPORT

PF keys:  1 Help   2 Msg   3 Exit   4 Menu   6 Id
```

2. Take one of the following actions:
- To generate a printed report of the Process statistics on the printer (defined in your signon defaults), fill in the criteria using the entry fields, and in the OPTION field, type P and press ENTER.
 - To access the SELECT STATISTICS - SUMMARY TABLE screen where a summary of Process statistics are displayed, fill in the criteria using the entry fields, and in the OPTION field, type S and press ENTER.

The following table describes the entry fields for the SELECT STATISTICS screen:

Field	Description
OPTION	This 1-character field contains the S or P option as follows: <ul style="list-style-type: none">• S to access the SELECT STATISTICS - SUMMARY TABLE screen where a summary of Process statistics are displayed.• P to generate a printed report of the Process statistics on the printer (defined in your signon defaults).
PROCESS NUMBERS	These four fields of eight characters each contain the numbers of Processes selected for statistical information. You can type one to four Process numbers for statistics selection.
PROCESS NAMES	These four fields of eight characters each contain the names of Processes selected for statistical information. You can type one to four Process names for statistics selection.

Field	Description
START DATE	<p>This 8-character field specifies the date from which the statistics records are selected. You can specify the day (D), month (M), and year (Y).</p> <p>To specify the order of a Gregorian day, month, and year, you must define the DATEFORM initialization parameter. If you do not specify the DATEFORM parameter, Connect:Direct for z/OS defaults to MDY date format.</p> <p>After you have specified the order in the DATEFORM initialization parameters, you can use the following formats according to the order you selected:</p> <ul style="list-style-type: none"> • DATEFORM=MDY specifies the use of the following formats: <ul style="list-style-type: none"> – mmddyy or mmddyyyy – mm/dd/yy or mm/dd/yyyy – mm.dd.yy or mm.dd.yyyy • DATEFORM=DMY specifies the use of the following formats: <ul style="list-style-type: none"> – ddmmyy or ddmmyyyy – dd/mm/yy or dd/mm/yyyy – dd.mm.yy or dd.mm.yyyy • DATEFORM=YMD specifies the use of the following formats: <ul style="list-style-type: none"> – yymmdd or yyyyymmdd – yy/mm/dd or yyyy/mm/dd – yy.mm.dd or yyyy.mm.dd • DATEFORM=YDM specifies the use of the following formats: <ul style="list-style-type: none"> – yyddmm or yyyyddmm – yy/dd/mm or yyyy/dd/mm – yy.dd.mm or yyyy.dd.mm <p>IBM Connect:Direct processes Julian dates the same as previous releases. The following formats are valid:</p> <ul style="list-style-type: none"> • yyddd or yyyyddd • yy/ddd or yyyy/ddd • yy.ddd or yyyy.ddd
START TIME	<p>This 10-character field contains the start date expressed as hours, minutes, and seconds, and AM or PM (HH:MM:SSXM). You can specify either a 12-hour or 24-hour time or the words NOON or MIDNIGHT. If you use 12-hour time, designate AM or PM; if you do not, Connect:Direct for z/OS CICS assumes a 24-hour time format. Processes that started after the time specified are selected for reporting.</p>
STOP DATE	<p>This 8-character field specifies the date from which the statistics records are selected. Specify the day (D), month (M), and year (Y).</p> <p>To specify the order of a Gregorian day, month, and year, you must define the DATEFORM initialization parameter. If you do not specify the DATEFORM parameter, Connect:Direct for z/OS defaults to MDY date format.</p> <p>After you have specified the order, you can specify formats. See the START DATE field for a description of the formats that you can use.</p>

Field	Description
STOP TIME	This 10-character field contains the stop date expressed as hours, minutes, and seconds AM or PM (HH:MM:SSXM). You can specify either a 12-hour or 24-hour time or the words NOON or MIDNIGHT. If you use 12-hour time, designate AM or PM; if you do not, Connect:Direct for z/OS CICS assumes a 24-hour time format. Processes that started before the time specified are selected for reporting.
CONDITION CODE	This dual field contains the condition code for the selected Process; the first 2-character field contains a logical operator (LT, LE, GT, GE, NE, or EQ); and the second 8-character field contains the condition code to be checked. If a condition code is specified, all Processes terminating with the specified code are selected for reporting.
EXCLUDE MEMBER RECS	This 1-character field contains the Y or N toggle to include or omit member records from PDS copy statistical information.
EXCLUDE WTO RECS	This 1-character field contains the Y or N to include or omit Write To Operator records in the displayed statistical information. If you do not want WTO records to be displayed, select Y; if you do, select N.

Viewing Statistics Summary Information

About this task

Use the SELECT STATISTICS - SUMMARY TABLE screen to view a list of Process statistics, including function type, Process name and number, sending and receiving nodes, last message received, end time and date, and the return code.

Procedure

1. To access the SELECT STATISTICS - SUMMARY TABLE screen, select option **S** on the SELECT STATISTICS screen, and press ENTER.

Following is an example.

SELECT STATISTICS - SUMMARY TABLE							17:36:45
Cmd (M)	Function	ProcName	ProcNum	Submitter	Node	Secondary Node	node.name Last Msgid RC
Submitter	P/Snode	End date	End time				
-	SUB-CMD	COPYNC	0	OS390.NODE	OS390.NODE	SPQL001I	0C
-	OPER	PNODE	05/05/1998	12:37:46			
-	SUB-CMD	FROMMVS	0	OS390.NOD	OS390.NODEEE	SPQL001I	0C
-	OPER	PNODE	05/05/1998	12:40:06			
-	SUB-CMD	FROMMVS	0	OS390.NOD	OS390.NODEEE	SPQL001I	0C
-	SCIVSE5	PNODE	05/05/1998	12:44:19			
-	SUB-CMD	FROMMVS	1	OS390.NOD	OS390.NODEEE	SSPA001I	00
-	SCIVSE5	PNODE	05/05/1998	12:59:05			
-	PROC-END	FROMMVS	1	OS390.NOD	OS390.NODEEE	ACF01012	0C
-	SCIVSE5	PNODE	05/05/1998	12:59:20			
-	SUB-CMD	FROMMVS	2	OS390.NOD	OS390.NODEEE	SSPA001I	00
-	SCIVSE5	PNODE	05/05/1998	13:03:55			
-	COPY-END	FROMMVS	2	OS390.NOD	OS390.NODEEE	SCPA000I	00
-	SCIVSE5	PNODE	05/05/1998	13:04:35			
-	PROC-END	FROMMVS	2	OS390.NOD	OS390.NODEEE	SVTM100I	00
-	SCIVSE5	PNODE	05/05/1998	13:04:35			
SOPS000I - Select Statistics command successfully completed.							Page 1 of 42
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id 8 Fwd							

2. To go to the LAST MESSAGE screen and display the text of the last message received by a particular Process, in the Cmd column next to the the line containing the Process, type M and press ENTER.

The following table describes the system fields for the SELECT STATISTICS - SUMMARY TABLE:

Field	Description
PAGE	17-character field contains the current page number and total pages of SELECT STATISTICS - SUMMARY TABLE data.
FUNCTION	9-character field contains the function designation for Connect:Direct for z/OS Process statistics.
PROCNAME	This 8-character field contains the Process name.
PROCNUM	This 6-character field contains the Process number.
SUBMITTER NODE	This 16-character field contains the node name of the primary node.
SECONDARY NODE	This 16-character field contains the node name of the secondary node.
LAST MSGID	This 8-character field contains the designation of the last message received by the Process.
RC	This 2-character field contains a system-generated return code as a result of the success or failure of your Process, as follows: <ul style="list-style-type: none"> • 00 - Processes that completed successfully • 04 - Processes that completed successfully and have one minor error such as an incorrect file disposition • 08 - Processes that did not complete successfully but that have several errors • 12 - Processes with major errors
SUBMITTER	This 8-character field contains the userid of the submitted Process.
P/SNODE	This 5-character field contains the primary or secondary node name designation, either PNODE or SNODE.
END DATE	This 8-character field contains the ending date of the displayed Process, expressed as month, day, and year (MM/DD/YY).
END TIME	This 8-character field contains the ending time of the displayed Process, expressed as hours, minutes, and seconds (HH:MM:SS).

Displaying CICS Messages

Displaying Product Messages

About this task

Use the MESSAGE DISPLAY screen to view the contents of a message, the software module producing the message, some details of system action, and suggested response on your part.

Note: The CICS Interface messages use the prefix SCCS to distinguish them from non-CICS IBM Connect:Direct messages.

Procedure

1. To access the MESSAGE DISPLAY screen, type **MD** and press ENTER on the PRIMARY MENU screen. (The MD option is completed as part of CICS transaction processing and does not require access to the IBM Connect:Direct DTF API.)

Following is an example.

MESSAGE DISPLAY		06:15:43
MESSAGE ID ==>	-----	node.name
MODULE ID ==>		
SHORT TEXT==>		
LONG TEXT==>		
==>		
==>		
==>		
==>		
==>		
==>		
==>		
==>		
==>		
PF keys: 1 Help 2 Msg 3 Exit 4 Menu 6 Id		

2. To display a message type its 8-character string in the MESSAGE ID field and press ENTER.

The following table describes the system fields for the MESSAGE DISPLAY screen:

Field	Description
MODULE ID	64-character field contains the name of the software module that produced the message.
SHORT TEXT	64-character field contains the short version of the message as it is displayed when sent.
LONG TEXT	Fields of twelve 64-character lines contain the long version of the message, with further descriptions of system action and suggested responses.

Displaying the Last Message

About this task

Use the LAST MESSAGE screen to recall and view the contents of the last message received during your Connect:Direct for z/OS CICS activities, the software module producing the message, some details of system action, and suggested response on your part.

Procedure

To access the LAST MESSAGE screen you have the following two options:

- Type M and press ENTER in the CMD field of the SELECT STATISTICS -- SUMMARY TABLE screen. (The M command is completed as part of CICS transaction processing and does not require access to the IBM Connect:Direct DTF API.)
- Press PF2 from any screen that defines PF2, except from the PRIMARY MENU where PF2 is not displayed. The last message may have nothing to do with the current screen activities, when you press PF2.

Following is an example of the LAST MESSAGE screen.

```

                                LAST MESSAGE                                06:15:43
                                node.name

MESSAGE ID ==> SAFA000I
MODULE ID  ==> DMSGNON
SHORT TEXT==> Connect:Direct signon process completed.
LONG TEXT==> Connect:Direct signon processing completed. The user
              ==> record was found on the Authorization Data Set. The user
              ==> supplied password was valid.
              ==>
              ==>
              ==> SYSTEM ACTION: Return to invoker with RC=0.
              ==>
              ==>
              ==> RESPONSE: NONE.
              ==>
              ==>
PF keys:   3 Exit   4 Menu

```

The following table describes the system fields for the LAST MESSAGE screen:

Field	Description
HEADER	This 41-character field contains the system-generated message identification information and occurs right below the screen title. The HEADER field works in conjunction with the SELECT STATISTICS - SUMMARY TABLE screen and displays the current screen upon the selection of the line command M in the CMD field.
MESSAGE ID	This 8-character field contains the message identification number.
MODULE ID	This 64-character field contains the name of the software module issuing the message.
SHORT TEXT	This 64-character field contains the short version of the message as it is displayed when sent.
LONG TEXT	This field of 12 lines of 64 characters contains the long version of the message, with further descriptions of system action and suggested responses.

Displaying Your CICS User Profile

To access the USER INQUIRY screen, press PF6 from any screen that defines PF6, except from the PRIMARY MENU, where PF6 is not displayed. Following is an example of the USER INQUIRY screen.

```

                                USER INQUIRY                                17:45:32
                                node.name

CICS USERID      ==> SYSA      CONNECT:DIRECT VERSION ==> VV
CONNECT:DIRECT  USERID ==> SYSA      CONNECT:DIRECT RELEASE ==> RR
CICS LUNAME      ==> BAN06061  CONNECT:DIRECT MOD-LEVEL ==> MM
CICS TERMID      ==> 6061
CONNECT:DIRECT  NODE   ==> CD.ESA13

                                PRIMARY MENU OPTION      AUTH?
                                -----
                                CF COPY FILE                YES
                                SB SUBMIT PROCESS            YES
                                SP SELECT PROCESS            YES
                                SS SELECT STATISTICS          YES
                                MD MESSAGE DISPLAY           YES
                                SD SIGNON DEFAULTS           YES
                                SN CHANGE SIGNON             YES

PF keys:   3 Exit   4 Menu

```

The following table describes the system fields:

Field	Description
CICS USERID	This 8-character field contains your CICS userid.
CONNECT:DIRECT USERID	This 8-character field contains your IBM Connect:Direct userid.
CICS LUNAME	This 8-character field contains your CICS logical unit name (netname).
CICS TERMID	This 4-character field contains your CICS terminal designation.
CONNECT:DIRECT NODE	This 16-character field contains the name of the IBM Connect:Direct node to which you are currently signed on.
CONNECT:DIRECT VERSION	This 2-character field contains the version number of the CICS Interface software you are using.
CONNECT:DIRECT RELEASE	This 3-character field contains the release number of the CICS Interface software you are using.
CONNECT:DIRECT MOD-LEVEL	This 3-character field contains the modification level number for the CICS Interface software you are using.
COPYFILE	This 3-character field contains YES or NO to show your authorization to perform a COPY operation.
SUBMIT PROCESS	This 3-character field contains YES or NO to show your authorization to submit a Process.
SELECT PROCESS	This 3-character field contains YES or NO to show your authorization to select a Process.
SELECT STATISTICS	This 3-character field contains YES or NO to show your authorization to select statistics.
MESSAGE DISPLAY	This 3-character field contains YES or NO to show your authorization to display messages.
SIGNON DEFAULTS	This 3-character field contains YES or NO to show your authorization to update your signon defaults record.
SIGNON	This 3-character field contains YES or NO to show your authorization to change your signon.

CICS Messages and Problem Isolation

Product Messages

These messages from the system software are arranged in alphabetical order.

Activate rejected, node is already active - node name

This message is displayed if you select option A in the field beside an active node.

All values reset from config file

This message is displayed if you press Clear.

All values reset from signon defaults file

This message is displayed if you press CLEAR on the SIGNON DEFAULTS screen.

A printer must be specified in your Signon Defaults in order to use 'P'

This message is displayed if you forgot to type in a printer ID, before you selected option P, and then pressed Enter.

Auto-return in progress...

This message is displayed after you exited the Connect:Direct for z/OS CICS program, and retyped the IBM Connect:Direct transaction. The automatic return feature must be enabled before any use, however.

CICS USERID required

This message is displayed if you pressed Enter with a blank screen present or with no CICS USERID entry present.

Control record successfully updated

This message is displayed after an edit session when you press PF9.

DTF NODE NAME required

This message is displayed if you pressed Enter, or PF5, or PF6, with a blank screen.

End of file

This message is displayed when you scroll forward to the bottom of the file through repeated use of the PF8 key.

End of file; values read from config file

This message is displayed when you scroll forward to the end of file and then press PF8.

End of file; values read from signon defaults file

This message is displayed if you press PF8 repeatedly and attempt to access data beyond the end of file.

End of node list

This message is displayed if you press PF8 while you are at the bottom of the available node list.

FILENAME and FILETYPE are required

This message is displayed if you pressed Enter on a blank screen, without typing the filename and type of file.

First page

This message is displayed if you press PF7 repeatedly on a screen with more than one page and attempt to go up past the first page.

Immediate shutdown rejected; interface is shut.

This message is displayed if option I is already selected, and the interface is already inactive.

Immediate shutdown started

This message is displayed if you select option I. The INTERFACE STATUS field changes to INACTIVE.

Interface has been started

This message is displayed if you select option A and press Enter.

Interface is already active

This message is displayed if you select option A, and the interface is already active or in the process of starting.

Interface must be active to start monitor.

This message is displayed if you select option M, and the interface is not active. You must select option A before option M.

Last Msgid field is blank

This message is displayed if you select the M option, and press Enter, with the LAST MSGID field blank.

Last page

This message is displayed if you press PF8 repeatedly on a screen with more than one page and attempt to go down past the last page.

Left page

This message is displayed if you repeatedly press PF10 and attempt to access data past the left edge of the screen.

Line command invalid

This message is displayed if you select a different line command other than those defined and press Enter.

M and S are the only valid line commands

This message is displayed, if you select a line command other than those defined in the CMD field and press Enter.

M is the only valid line command

This message is displayed if you type in a character other than M in the command line and press Enter.

Monitor has been started.

This message is displayed if you select option M.

Monitor is already running.

This message is displayed if option M is already selected and you select option M again.

MYNODE.OS.USERID NODE invalid

This message is displayed if the node name you typed in is not authorized for signon.

NETWORK NODE NAME required

This message is displayed if you press Enter, or PF5, or PF6, with a blank screen present or with the NETWORK NODE NAME field entry blank.

Network node successfully added - node name

This message is displayed when the DTF NODE RECORDS screen has your correct data in the fields, and you press PF5.

Network node successfully deleted - node name

This message is displayed when the DTF NODE RECORDS screen has your correct data in the fields, and you press PF6.

No active work queue entries for node node name

This message is displayed if you select option W, and no active subtasks are in the work queue.

Node activation started, node node name

This message is displayed if you select option A in the field beside an inactive node.

Node has been restarted - node name

This message is displayed in the STATUS ALERT MESSAGE field when the NETWORK NODE RECORDS have been updated to activate a node.

NODE invalid

This message is displayed if the node name you typed in is not authorized for signon.

NODE NUMBER invalid

This message is displayed if you typed in a node number that is not in the list of available nodes and pressed Enter.

NODE NUMBER or NODE NAME required

This message is displayed if you press Enter, with nothing typed in.

NODE required

This message is displayed if you pressed Enter on a blank screen without typing in a node name.

NODE TYPE invalid

This message is displayed if the DTF NODE NAME contains an unacceptable naming convention.

No help available

This message is displayed if you press PF1 while at a screen with no online help facilities available.

No password is currently on file

This message is displayed in the IBM Connect:Direct PASSWORD field if you typed an unacceptable or blank IBM Connect:Direct PASSWORD.

Normal shutdown rejected; interface is inactive.

This message is displayed if already select option S, and you select option S again.

Normal shutdown started

This message is displayed if you select option S, press Enter. If you press Enter again, the MONITOR TASK NUMBER field changes to NOT RUNNING.

No signed-on users

This message is displayed if the administrative user selected line command T from the USER STATUS and pressed Enter, thereby terminating the administrative user session. This message is also displayed if you type U from the PRIMARY MENU and no signed-on users of IBM Connect:Direct exist.

No users meeting selection criteria

This message is displayed if a CICS userid is typed in the CICS USERID field that did not match those users logged on. First, check the USER STATUS screen to see who is logged on for a match.

OPTION invalid

This message is displayed if you type in an option other than those defined on the PRIMARY MENU.

OPTION required

This message is displayed if you press Enter, without typing in an option.

Password is on file, but not displayed

This message is displayed if you type a IBM Connect:Direct password that is already in the SIGNON DEFAULT record.

Password is present

This message is displayed if the PASSWORD field on the SIGNON DEFAULTS screen is filled with your password. The PASSWORD field is darkened to maintain security.

Password is absent

This message is displayed if you have not typed your PASSWORD in the field on the SIGNON DEFAULTS screen. As soon as you type the password, the rest of the required data, and press Enter, the message changes to PASSWORD IS PRESENT.

Past end of file - 'Prev' not available

This message is displayed when you scroll backward to the top of the file and then press PF7.

PF key invalid

This message is displayed in the MESSAGE field if you pressed a PF key other than those defined on the screen.

Press Enter to continue

This message is displayed if you typed in all required data correctly and pressed Enter. This message indicates your chance to abort the COPY file Process.

PROCESS NAME required

This message is displayed if you pressed Enter without typing in the Process name.

PROCESS NUMBER number

This message is displayed after a successful execution of a COPY file Process, where number is the system-generated Process number.

Record cannot be deleted, it is not on file

This message is displayed if you type in the DTF NODE NAME and then press PF6. No match exists between the DTF NODE NAME and those specified in the configuration file, and therefore the record cannot be deleted.

Record cannot be updated, it is not on file

This message is displayed if you type in the DTF NODE NAME and then press PF9. No match exists between the DTF NODE NAME and those specified in the configuration file, and therefore the record cannot be updated.

Record not found; hit any key when ready.

This message is displayed if you type in the DTF NODE NAME and press Enter, but no match exists in the DTF NODE RECORDS screen.

Right page

This message is displayed if you repeatedly press PF11 and attempt to access data past the right edge of the screen.

SAFA000I - Connect:Direct signon process completed.

This message is displayed if you signed on successfully.

SCBI190I - Process specified not in process library.

This message is displayed if you typed in a Process name that is not recognized by the system, and pressed Enter. Check the contents of your PROCESS.LIB to determine those available.

SCIA011I - Connect:Direct/DTF not active. Extended Submit Facility now available.

This message is displayed if the DTF is not working under IBM Connect:Direct. ESF is a substitute for the DTF in this case, but is limited to one Process per user.

SCCS002I - No room available on work queue for this node. The Connect:Direct for CICS work queue for this DTF node is full.

This message is displayed when the work queue reaches its maximum capacity of tasks. Try your request again at a lower usage time.

SCCS003I - Connect:Direct for CICS monitor is not active.

This message is displayed when the DGAM transaction is not running.

SCCS007I - DTF node not active to Connect:Direct for CICS.

This message is displayed if the node name you typed in is not active in the IBM Connect:Direct system.

SCCS007I - DTF node not active to Connect:Direct CICS.

This message is displayed if the node name you typed in is not active in the IBM Connect:Direct system. If you are an administrator, check the NODE STATUS screen.

SCCS008I - Connect:Direct for CICS signon failure.

This message is displayed when a system logic error prevented your Connect:Direct for z/OS CICS signon.

SCCS009I - Connect:Direct for CICS connection to DTF node is being shut.

This message is displayed when the logical connection from Connect:Direct for z/OS CICS to the DTF node is being shut down.

SCCS010I - Connect:Direct for CICS has been shut down.

This message is displayed when the Connect:Direct for z/OS CICS interface has been shut down by a system administrator, without waiting for active requests to complete execution.

SCCS011I - Connect:Direct for CICS subtask has been shut.

This message is displayed when the Connect:Direct for z/OS CICS subtask assigned to process your request has been shut down, without waiting for active requests to complete execution.

SCCS013I - Copyfile successfully submitted.

This message is displayed if you typed in all the required data on the SENDING FILE screen and the RECEIVING FILE screen, and pressed Enter. You are returned to the COPYFILE screen from the RECEIVING FILE screen. The message is displayed only when a COPY file Process successfully completes under DTF.

SCCS014I - Output limit has been exceeded.

This message is displayed if the total number of lines returned to the DTF node exceeded the limit defined for the node during a SELECT operation.

SCCS016I - Connect:Direct for CICS connection to DTF node is being shut.

This message is displayed when the logical connection from Connect:Direct for z/OS CICS to the DTF node is being shut down.

SCCS018I - Request could not be assigned to a subtask.

This message is displayed if you typed in incorrect data or unknown data. The system could not place the signon request in the work queue assigned to the DTF node for the length of time required to complete the signon process. Number of maximum users was exceeded, or incorrect entries in the SIGNON DEFAULTS record caused the failure.

SCCS023I - DTF node now available in ESF mode only.

This message is displayed if the DTF is down.

SENDING DSNAME required

This message is displayed if you pressed Enter, without typed a filename.

Sending or receiving node must equal current node - MYNODE.OS.USERID

This message is displayed if you typed in a different node number from yours in the sending number field and pressed Enter.

SESF000I - Process successfully submitted via ESF.

This message is displayed if you typed in all the required data on the SENDING FILE screen and the RECEIVING FILE screen and pressed Enter. You are returned to the COPYFILE screen from the RECEIVING FILE screen. The message is displayed only when you complete successfully a COPY file Process under ESF (when DTF is not working).

Shut immediate started, node node name

This message is displayed if you select option I, and you press Enter on an active node.

Shut normal started, node node name

This message is displayed if you select option N, and you press Enter, on an active node. The STATUS field changes to INACT, and the REQUEST field changes to SHUTNORM.

Shut rejected; node already inactive - node name

This message is displayed if you select option S and you press Enter on an inactive node.

Signon Defaults successfully updated - userid

This message is displayed if you press PF9 while at the SIGNON DEFAULTS screen.

SOPA000I - Select process command was successful.

This message is displayed if you select option S and press Enter.

SOPA006I - No process(es) found matching the search criteria.

This message is displayed if you typed in values on the SELECT PROCESS screen, pressed Enter, and no match was found.

SOPA011I - One or more processes SELECTed.

This message is displayed if you press Enter without selecting a line command option.

SOPS000I - Select Statistics command successfully completed.

This message is displayed if you type in the correct data on the SELECT STATISTICS screen and press Enter.

SOPS006I - No statistics were found matching the criteria specified.

This message is displayed if you type in the correct data on the SELECT STATISTICS screen and press Enter, but there was no Process data in Connect:Direct for z/OS CICS that matched your data.

START DATE invalid

This message is displayed if you typed in a start date that is unrecognizable by Connect:Direct for z/OS CICS.

Start of file

This message is displayed if you scroll back to the top of the file through repeated use of the PF7 key.

Start of file; values read from config file

This message is displayed when the you press PF7 repeatedly to get to the top of the configuration file, then PF8, and then PF7.

Start of file; values read from signon defaults file

This message is displayed if you press PF7 repeatedly and attempt to access data before the beginning of the file.

Start of node list

This message is displayed if you press PF7 while you are already at the top of the available node list.

START TIME invalid

This message is displayed if you pressed Enter, with nothing typed in on the screen, or if you typed in a start time that is unrecognizable by Connect:Direct for z/OS CICS and pressed Enter.

UNSUPPORTED FUNCTION

This message is displayed if the CICS Interface is not started or if you typed in a command unknown to the system.

User does not have active work

This message is displayed if you select line commands F or C and no active subtask exists in the USER STATUS.

USS Command Completed Successfully

This message is displayed when the CICS signon is successful.

Value must be numeric if typed

This message is displayed if you typed in other characters than 0 to 9 and pressed Enter.

Values read from config file * * * * *

This message is displayed if you press PF9. The screen is updated with the values recorded in the configuration file.

Note: The following message begins with a variable field and can change as the node names are changed.

node name not NETWORK NODE IN NETMAP specified

This message is displayed if you type in the DTF NODE NAME and then press PF5. No match exists between the DTF NODE NAME and those specified in the NETMAP.

You must sign on to CICS before using Connect:Direct for CICS

This message is displayed in the STATUS ALERT MESSAGE field if you attempted to sign on to IBM Connect:Direct before CICS.

Problem Isolation

Use the following suggested solutions to software problems when troubleshooting.

DTF Busy Message

The terminal is clocked when you press a PF key or Enter on a IBM Connect:Direct screen. If the processing of the IBM Connect:Direct transaction requires communications with a IBM Connect:Direct node, then your request is put into a CICS WAIT state.

The terminal clock is freed when the CICS Interface handles your request. The CICS Interface administrative function provides an inquiry capability for problem analysis in the event that the clock is not freed. A CICS Interface administrative function exists that aborts any specific IBM Connect:Direct command that is in progress. This command frees the terminal clock and an error message is returned.

The IBM Connect:Direct request is not cancelled, unless the request is not yet in progress, meaning that your request is allowed to complete, but the response is not sent back to you.

You cannot type another IBM Connect:Direct command for the same node until the first request completes (only one command is allowed per user per node at a time).

CICS Transaction ABENDs

If a CICS transaction abends, then you are returned to CICS transaction mode. You can try to retype the transaction that abended, but you must start over, either on the SIGNON screen or on the PRIMARY MENU (if auto-signon is enabled).

In case the transaction abends while a IBM Connect:Direct command is in process (because of a failure of the terminal, or because a CICS operator force-abended his transaction), the same situation occurs.

However, you cannot issue any further IBM Connect:Direct requests until the prior request completes. Completion information from the outstanding request is not returned to you.

Your Terminal Hangs—No Response

CICS Interface requests are placed in a work queue, with one queue per node. IBM Connect:Direct requests are rejected with a DTF busy error message, when a queue for a node reaches its maximum allowed length. If this condition occurs, contact your system administrator.

Chapter 6. Administration Guide

Basic System Administrative

The Administrative Options Menu provides access to system administration functions. To access this menu, do one of the following:

- Select ADMIN from the Primary Options Menu.
- Type =ADMIN on any IBM Connect:Direct screen command line and press ENTER.

The following figure is an example of the Administrative Options Menu. To access a function, type the function abbreviation on the command line and press ENTER.

Note: Various functions on the Administrative Options Menu may not be available to all users. Access is controlled through the User Authorization file.

```
View  Modify  Control  Delete  Secure+  Help
-----
CD.ART          Connect:Direct Administrative Options Menu
Option ==>

Select one of the following:

ST  - View type record
IT  - Insert/Update type record
DT  - Delete type record
SU  - View user authorization record
IU  - Insert/Update user authorization record
DU  - Delete user authorization record
TS  - View Connect:Direct tasks
TF  - Flush a Connect:Direct task
S   - Execute Secure Plus Commands
MD  - Modify Connect:Direct trace characteristics
C   - Enter a native Connect:Direct command
SN  - Terminate Connect:Direct
ARS - ARS reporting facility
NM  - View information in the Connect:Direct network map
UNM - Update the Connect:Direct network map
INQ - Inquire about DTF internal status
STAT - Perform statistics functions

*****
* Today: 08.30.2018 *
* Time: 14:05      *
* UID:  EPETE1     *
* OPT Enabled      *
* OPT Part-Enabled *
* OPT Disabled     *
*****
```

Execute Secure Plus Commands option is not displayed, unless you have functional authority for Connect:Direct Secure Plus. See [Functional Authority Privileges](#) (specifically the information in BYTE8 in the DGA\$MFLG macro).

Select the following options from the Connect:Direct Administrative Options Menu.

- To maintain the Type Defaults file with file attribute information used during Process submission, use the following options.

Option	Description
ST	Displays the Select Type screen where you can examine a record in the Type file and select the output to go to a file, table, or printer.
IT	Displays the Insert/Update Type screen where you can add or change a record in the Type Defaults file.
DT	Displays the Delete Type screen where you can remove a record from the Type file.

- To maintain the User Authorization file that controls access to functions, use the following options.

Option	Description
SU	Displays the Select User screen where you examine the user profile in the Authorization file.
IU	Displays the Insert/Update User screen where you can add a user to the system or change user privileges on the system.
DU	Displays the Delete User screen where you can remove a user from the Authorization file.

- To select and flush tasks, use the following options:

Option	Description
TS	Accesses the Select Task screen where you determine the task ID, type, and task number.
TF	Displays the Flush Task screen where you can remove a task from the execution queue.

- To perform Connect:Direct Secure Plus functions, use the following options. You must first select the Execute Secure Plus Commands from the Connect:Direct Administrative Options Menu.

Option	Description
CR	Executes the Certificate Expiration Validation Command on demand to let you see warnings both for certificates that have expired and will soon expire.
RF	Executes the Refresh Secure Plus Environment Command to update the SSL and TLS environments after you have changed security-related information.
SA	Executes the Connect:Direct Secure Plus Admin Tool and displays the Connect:Direct Secure Plus Admin Tool Main Screen. See the <i>IBM Connect:Direct Secure Plus for z/OS Implementation Guide</i> for more information on using Connect:Direct Secure Plus.

- To initialize traces, suspend and resume sessions, modify initialization parameters, type native commands, terminate IBM Connect:Direct, and access ARS, use the following options.

Option	Description
MD	Displays the Modify Command screen where you can request traces and modify system functions. For information on the MODIFY command and trace and debug settings, see IBM Connect:Direct MODIFY Command . For information on the MODIFY SESSIONS command, see Suspending and Resuming Quiesce and Trace Settings . For information on the MODIFY INITPARMS command, see Modifying Initialization Parameter Settings while IBM Connect:Direct is Running .
C	Displays the Native Command screen where you can type and execute any Connect:Direct for z/OS command by providing it in native syntax. For information, see IBM Connect:Direct Native Command Structure .
SN	Displays the Stop IBM Connect:Direct screen where you stop the operation of IBM Connect:Direct. For more information, see Stopping IBM Connect:Direct .
ARS	Displays the ARS REPORT OPTIONS menu. While IBM Connect:Direct produces statistics, the Activity Reporting System (ARS) gives you access to more information and also provides sorting capabilities. For more information, see the <i>IBM Connect:Direct for z/OS Facilities Guide</i> .

- To view and maintain the network map and translate TCP/IP names, use the following options.

Option	Description
NM	Displays the Select Network Map screen, where you choose to display or print, the defined IBM Connect:Direct nodes from the network map file and translation of TCP/IP host names and network addresses. For more information on the SELECT NETMAP command and the SELECT TCPXLT command, see <i>The Network Map</i> in <i>IBM Connect:Direct for z/OS User Guide</i> .
UNM	Enables you to update the network map dynamically. For more information, see Updating the Netmap through the IUI Interface .

- Start keywords on the next line or break them by a separator (blank or comma).
- To use comments on the Native Command Screen, type an asterisk in the first column of the input line. Typing an asterisk enables you to issue commands without retyping them.
- You cannot continuously wrap commands across lines on the Native Command Screen.

Examples

[illegible]

You can also submit a Process from the command line. In the following example, the Process TEST2 is submitted from the command line. The SELECT PROCESS (line 3) takes place just as in the previous example. The screen sample follows.

[illegible]

Managing Tasks

IBM Connect:Direct tasks perform and manage work in a DTF. This section describes how to display task information and remove (flush) tasks.

The following table lists the IBM Connect:Direct tasks and their functions:

Type	Task	Function
System	Master (M)	Controls the dispatching and logon processing for the DTF
	Timer (T)	Performs timer services for the master task and Process-related timer functions
	Operator interface (C)	Enables you to communicate to the DTF through the operator console
	Extended Submit Facility Scan (W)	Scans the TCQ at predefined intervals and moves submitted Processes that are not on the current processing queue (PCQ) to the PCQ
System	Open/Close Task (O)	Manages the VTAM ACB open/close and TPEND exit
	TCP Task (U)	Monitors incoming TCP/IP session requests
	XCF Communication (Q)	Manages communications between Manager and Servers in a IBM Connect:Direct/Plex
	TCP API Task (D)	Monitors incoming TCP/IP IBM Connect:Direct API session requests
	LOGON (L)	Reserved for use during logon processing
	Statistics (A)	Controls status logging
	Session Creation TCA (F)	Manages Processes and tasks
	Statistics Archive Submit Task (Z)	Submits the Statistics File Archive Process
	CTCA Server Task (Y)	Manages the CTCA tasks
	IBM Connect:Direct/Plex Queue Manager Task (Q)	Manages the VTAM ACB open/close and TPEND exit
	PNODE Task (P)	Manages the work related to a request that initiated the current session
	SNODE Task (S)	Manages the work related to a partner PNODE task
User	IUI Task (I)	Manages the requests from a session with an IUI user
	Background (Batch) Task (B)	Manages the request from a batch user

Displaying Task Status

Use the SELECT TASK command to select and display the status of IBM Connect:Direct tasks. It has the following format and parameters.

Label	Command	Parameters
(optional)	SElect TASK	PRint Operator Table DISplay
		WHERE (SERVER = server name)

The required SELECT TASK parameters are:

Parameter	Description
PRInt Operator Table DISplay	PRInt specifies the command is output in hard copy to a printer. TABLE specifies the command is output to a table. DISplay specifies the command is output to the screen.

The optional SELECT TASK parameters are:

Parameter	Description
WHERE(SERVER = server name)	This parameter specifies the name of the IBM Connect:Direct/Plex member where the SELECT TASK command is performed. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter. This parameter only applies to a IBM Connect:Direct/Plex environment. If this parameter is not specified in a IBM Connect:Direct/Plex environment, the SELECT TASK is performed on the IBM Connect:Direct/Manager.

Examples

The following SELECT TASK command example sends output to the log printer:

```
SEL TASK PRINT
```

The following SELECT TASK example is performed on a IBM Connect:Direct/Server named SERVER3 and sends the output to your terminal in operator table format.

```
SEL TASK O WHERE (SERVER=SERVER3)
```

Selecting a Task through the Batch Interface

About this task

To use the SELECT TASK command from the batch interface:

Procedure

1. Place your commands in a batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Selecting a Task through the IUI

About this task

You must select the appropriate output for the SELECT TASK report. You can either display (in report or operator table format) or print the report.

Procedure

1. Select option TS from the Administrative Options Menu.
The SELECT TASK screen is displayed.

node.name	SELECT TASK	hh:mm
CMD ==>		CMD: OPR
CMD: O ... OPERATOR TABLE P ... PRINT REPORT D ... DISPLAY REPORT SERVER => _____		

2. Select one of the following display types.

Option	Description
D	Displays the report on your screen and is captured in the TMPDSN specified in the SIGNON defaults of the user.
O or OPR	Displays the report to your screen in the operator table format.
P	Sends the report to a printer.

Note: If you are running in a IBM Connect:Direct/Plex environment, type the member name on which you want to perform the SELECT TASK in the SERVER field. If you leave this field blank in a IBM Connect:Direct/Plex environment, the SELECT TASK is performed on the IBM Connect:Direct/Manager. If you are running in a IBM Connect:Direct/Stand-alone Server environment, leave this field blank.

3. Press **ENTER**.

- If you selected Display from the Select Task screen, the following screen is displayed:

```

BROWSE - -XXXXXXXX.XXXXXX.XXXXX.XXXXX.XXXXX--LINE 00000000 COL 001 080
COMMAND ==>                                SCROLL ==> CSR
*****
***** TOP OF DATA *****
=====
          SELECT TASK for C:D/Plex Manager
=====
TASK TASK      XMIT      PNAME/
ID  NUM  STATE  STATE      PNUM
-----
M   001  INACTIVE
-----
T   002  TIMER
-----
A   003  INACTIVE
-----
Z   004  INACTIVE
-----
C   005  MISC I/O
-----
F   006  INACTIVE
-----
U   007  SUBTASK  TCP MAIN TAS
  
```

- If you selected the operator table format, the following screen is displayed:

-----OPERATOR TABLE-----						Row 1 to 20 of 22
==> OPTION	TID	TASKNO	STATE	SUB-STATE	PNAM/UID	SCROLL ==> PAGE PNUM
	M	1	INACTIVE			
	T	2	TIMER			
	A	3	INACTIVE			
	Z	4	INACTIVE			
	C	5	MISC I/O			
	F	6	INACTIVE			
	U	7	SUBTASK	TCP MAIN TAS		
	U	8	ST RUNNG	TCP ACCEPT		
	D	9	SUBTASK	TCP MAIN TAS		
	D	10	API RUN	TCP ACCEPT		
	O	11	INACTIVE			
	I	13	VTAM I/O	RECEIVE	SJONES2	
	I	14	RUNNING		BSMITH1	
	W	12	TIMER			
	Q	37	WAIT4WRK			
	Q	38	WAIT4WRK			
	Q	39	WAIT4WRK			
	Q	40	WAIT4WRK			
	Q	41	WAIT4WRK			
	Q	73	WAIT4WRK			

You can perform the following operations from the Operator Table:

- Type F next to the task ID to flush and suspend nonsystem tasks.
- Type P to suspend a task.

Refresh the OPERATOR TABLE screen by typing Q line and pressing ENTER.

Removing Tasks from Execution

Use the FLUSH TASK command to remove a task from execution. Identify the task by its task number. You cannot flush system or IUI tasks.

Note: Only use the FLUSH TASK command if you cannot flush the Process using the FLUSH PROCESS command.

The FLUSH TASK command has the following format and associated parameters. Required parameters are in bold print.

Label	Command	Parameters
(optional)	FLush TASK	WHERE (
		TASK = (tasknumber (list),
		SERVER = server name
)
		FORCE

The required FLUSH TASK parameter is:

Parameter	Description
WHERE (TASK = (tasknumber (list), SERVER=server name)	<p>(tasknumber (list) specifies the tasks to flush either by task number or a list of task numbers.</p> <p>This parameter is required. SERVER=server name specifies the name of the IBM Connect:Direct server where the FLUSH TASK is performed. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter. This parameter is only valid in a IBM Connect:Direct/Plex environment. If you omit this parameter in a IBM Connect:Direct/Plex environment, the FLUSH TASK is performed on a IBM Connect:Direct/Manager.</p>

The optional FLUSH TASK parameter is:

Parameter	Description
FORCE	<p>Specifies that flush task is forced. Do not use the FORCE parameter when the task is executing on a LU6.2 session. The session terminates immediately and statistics are not exchanged between the two nodes.</p> <p>If you do not specify the FORCE option for the FLUSH TASK command, then an indicator notifies the program executing on behalf of the task that a FLUSH TASK command was issued for that task. If that program is not in control (for example, if it is waiting on a request outside of IBM Connect:Direct to complete), then it does not recognize the FLUSH TASK indicator, and the task is not flushed; otherwise, the program recognizes for the FLUSH TASK indicator and takes the appropriate action.</p> <p>When you specify the FORCE option, then the action taken depends on the STATE and SUBSTATE of the task for which you issued the FORCE FLUSH. See Connect:Direct Process Language for the actions taken for the specific STATE and SUBSTATE.</p> <p>Note: Use the SELECT TASK command to determine the STATE and SUBSTATE of the task.</p>

Examples

The following example shows the FLUSH TASK command force flushing three tasks:

```
FLUSH TASK WHERE (TASK=(100,105,120)) FORCE
```

The following example shows the FLUSH TASK command flushing a task running on a IBM Connect:Direct/Server named OSGOOD:

```
FLUSH TASK WHERE (TASK=9,SERVER=OSGOOD)
```

Removing Tasks from Execution through the Batch Interface

About this task

To use the FLUSH TASK command from the batch interface, perform the following steps:

Procedure

1. Place your commands in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Removing Tasks from Execution through the IUI

About this task

Refer to the example of the SELECT TASK Operator Table for information about how to flush a task using the Operator Table.

To flush a task using the Flush a Task screen:

Procedure

1. Select option TF from the Administrative Options Menu to access the Flush a Task screen.

node.name	FLUSH A TASK	hh:mm
CMD ==>		
SERVER ==> _____		
TASK NUMBERS:		
==> _____	==> _____	==> _____
==> _____	==> _____	==> _____
FORCE: ('Y'-YES, 'N'-NO) FORCE FLUSH A TASK ON A LU 6.2 SESSION MAY		
==> _ TERMINATE THE SESSION IMMEDIATELY AND NO		
STATISTICS OF THE TASK WILL BE EXCHANGED		

2. If you are running in a IBM Connect:Direct/Plex environment, type the IBM Connect:Direct/Plex server name. If you leave this field blank, the Flush Task is performed on the IBM Connect:Direct Manager.

Leave this field blank if you are running in a IBM Connect:Direct/Stand-alone Server.

3. Type the numbers of the tasks you want to flush.
4. In the FORCE field, type Y to force the flush. Type N if you do not want to force the flush. The default is N.

A list of the requested tasks is displayed to indicate a successful flush.

Connect:Direct Secure Plus Commands

When you select the Execute Connect:Direct Secure Plus Commands on the Connect:Direct Administrative Options Menu, the following screen is displayed:

CD.ART	Execute Secure Plus Commands	14:02
CMD ==>		
CR	- Certificate Expiration Validation Command	*****
RF	- Refresh Secure Plus Environment Command	* * *
SA	- Secure Plus Admin Tool	* Today: 08.30.2018 *
		* Time: 14:02 *
		* UID: EPETE1 *
		* * *
		* OPT Enabled *
		* OPT Disabled *
		* * *

Certificate Validity Check Used by Connect:Direct Secure Plus

To check the validity of certificates using Connect:Direct Secure Plus, define the following initialization parameters:

- SECURE.DSN - the name of the parameter file
- CHECK.CERT.EXPIRE to perform a certificate validation check
- CHECK.CERT.EXPIRE.TIME to perform certificate validation checks
- CHECK.CERT.EXPIRE.WARNING.DAYS to indicate how many days before certificate expiration to issue a warning message

After certificate validation check is enabled, IBM Connect:Direct automatically monitors the status of the certificates as specified and whenever IBM Connect:Direct and Connect:Direct Secure Plus are initialized.

When the certificate validation checks are performed, IBM Connect:Direct verifies the label name and node name of all certificates in the Connect:Direct Secure Plus parameter file. After verifying all trusted certificates in the key store, IBM Connect:Direct reads and validates each individual certificate label. When a certificate expires, IBM Connect:Direct displays a CSPA601E error message indicating which certificate expired so that you can take appropriate action to generate or obtain a new certificate. When a certificate is soon to expire, IBM Connect:Direct displays a CSPA600W warning message indicating the specific certificate and the date it will expire.

In addition to issuing messages, IBM Connect:Direct generates statistic records to document the status of the certificate and which node name the certificate is defined for. You can use IBM® Control Center or the SELECT STATS command to audit the certificates and nodes that need attention. On the SELECT STATISTICS panel, type Y in the CHANGE EXTENDED OPTS field, and then specify CX as a RECORD TYPE on the SELECT STATISTICS EXTENDED OPTIONS panel.

Note: If IBM Connect:Direct cannot validate the contents of the Connect:Direct Secure Plus parameter file, it displays a CSPA607W message indicating that it could not retrieve the necessary information for a particular certificate. Make sure that the certificate exists along with the correct label name—note that the label is case-sensitive and must match exactly.

Note that expiration dates in certificates include both a date and time. IBM Connect:Direct uses both date and time to validate the exact expiration period when checking a certificate's validity. For example, if the certificate is set to expire on 12/31/2010 at 10:00:00 and the validation check is performed on 12/01/2010 at 09:00:00 with CHECK.CERT.EXPIRE.WARN.DAYS=30, this certificate will not be flagged with a warning message until a check is performed after 10:00:00 on 12/01/2010.

The Certificate Expiration Validation Command

To perform a certificate validation check upon demand, select the CR option, Execute Certificate Expiration Validation Command, when the Execute Secure Plus Commands menu is displayed.

IBM Connect:Direct displays any warning or error messages related to certificates should such a condition exist.

A TLS/SSL Environment Refresh

When you update security-related information effecting the SSL/TLS environment or database, such as updating the path to the key database or key ring or a certificate's label, use the Execute Refresh Secure Plus Environment Command. By updating the SSL and TLS environment, you ensure that all changes can take effect immediately. To execute this on-demand command, type RF and press Enter when the Execute Secure Plus Commands menu is displayed.

Connect:Direct Secure Plus Display

To display the Connect:Direct Secure Plus Admin Tool: Main Screen when the Execute Secure Plus Commands menu is displayed, type SA and press Enter.

```

File Edit Help
-----
CD.ZOS.NODE          Secure+ Admin Tool: Main Screen          Row 1 of 7
Option ==> ----- Scroll CSR

                        Table Line Commands are:

U Update node        H View History        D Delete node
I Insert node        V View node

Node Filter : *-----

LC Node Name          Type Secure+ Protocol Override Encryption External Client
-----
-- .CLIENT            R   *          N          *          *          *
-- .EASERVER          R   TLSV10       N          *          N          *
-- .PASSWORD          R   Disabled    *          *          *          *
-- CD.UNIX.NODE        R   TLSV10       *          *          *          *
-- CD.UNIX.NODE2       R   TLSV12       *          *          *          *
-- CD.ZOS.NODE         L   Disabled    Y          N          N          N
-- CD.ZOS.NODE2       R   *          *          *          *          *
***** BOTTOM OF DATA *****

```

Displaying Initialization Parameter Settings

Use the INQUIRE INITPARM command to view the current global and local initialization parameter settings.

The INQUIRE INITPARM command has the following format and parameter.

Label	Command	Parameter
(optional)	INQUIRE INITparm	WHERE (SERVER=server name)

The INQUIRE INITPARM parameter is:

Parameter	Description
WHERE (SERVER=server name)	This parameter is optional. This parameter specifies the IBM Connect:Direct/Plex server initialization parameters you want to view. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter. If you do not specify this parameter, the IBM Connect:Direct/Manager initialization parameters are displayed. You do not need this parameter for a IBM Connect:Direct/Stand-alone Server.

Sample Report

The following figure shows a partial sample report.

```

=====
node.name      *INQUIRE INITPARM*  DATE: mm.dd.yyyy TIME: hh:mm:ss
                C:D/Plex *** MANAGER ** in Group SDWGRP
=====
ABEND.CODES.NODUMP    => (SX37 SX13 U0728 SXD9 S9FC)
ABEND.RUNTASK         => ABEND.CODES.NODUMP
ALLOC.CODES           => (020C 0210 0218 0220 0234)
ALLOC.RETRIES         => 3
ALLOC.WAIT            => 00:00:30
ALLOCATION.EXIT        =>
CDPLEX               => Yes
CKPT                 => 0
CKPT.DAYS             => 4
CKPT.MODE             => (RECORD   BLOCK   PDS   NOPDS   VSAM   VSAM)
CTCA                 => No
DATEFORM              => MDY
DEBUG                => '00003001'
DESC.CRIT             => (2)

```

Using the INQUIRE INITPARM Command from the Batch Interface

About this task

To use the INQUIRE INITPARM command from the batch interface:

Procedure

1. Place your commands in a batch job stream.
2. Submit the job while IBM Connect:Direct is running.

Note: You must set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

3. Verify the results.

Issuing the INQUIRE INITPARM Command through the IUI

About this task

To display the DTF initialization parameters from the IUI:

Procedure

1. Select option INQ from the Administrative Options Menu.

The Inquire DTF Internal Status screen is displayed.

2. If you are running in a IBM Connect:Direct/Plex environment and want to view the initialization parameters of a IBM Connect:Direct/Server, type the server name in the Server field. If you want to view the initialization parameters of a IBM Connect:Direct/Manager, leave the Server field blank.

If you are running in a IBM Connect:Direct/Stand-alone Server, leave this field blank.

3. Select the **IPRM** option.
4. Press ENTER.

The current DTF initialization parameter settings are displayed.

Modifying Initialization Parameter Settings while IBM Connect:Direct is Running

You can update certain initialization parameters in the initialization parameter member, and then use the MODIFY INITPARMS command to update IBM Connect:Direct with the new initialization parameter settings without restarting, using one of the following methods:

- [Using the MODIFY INITPARMS Command from the Batch Interface](#)
- [Issuing the MODIFY INITPARMS Command through the IUI](#)
- Performing Configuration Management of Initparms through Control Center. For more information, refer to the documentation for Control Center.

You cannot update local initialization parameters with the MODIFY INITPARMS command using the first two methods listed above but you can using Control Center.

To ensure continuity in operation, create backup copies of the global and local initialization parameter members to use if IBM Connect:Direct cannot initialize after initparm updates. For information, see [Backing up the Global and Local Initialization Parameter Files](#).

The following list contains the initialization parameters that you can update while IBM Connect:Direct is running:

Note: The MODIFY INITPARMS command updates parameters in the following list, after all parameters (global and local) are parsed for correct syntax. Since the MODIFY INITPARMS command reads the entire global initialization parameter member (and local INITPARM members), the entire global initialization parameter member is used for the update (along with local INITPARM members). Therefore, you cannot update individual parameters with the MODIFY INITPARMS command.

ABEND.CODES.NODUMP	ABEND.RUNTASK	ALLOC.CODES
ALLOC.RETRIES	ALLOC.WAIT	CDPLEX.WLM.GOAL
CHECK.CERT.EXPIRE	CHECK.CERT.EXPIRE.TIME	CHECK.CERT.EXPIRE.WARN.DAYS
CKPT	CKPT.DAYS	CKPT.MODE
COMPRESS.EXT	COMPRESS.NEGO.FAIL	COMPRESS.NETMAP.OVERRIDE
COMPRESS.NETMAP.STD	CRC	DESC.CRIT
DESC.NORM	DESC.TAPE	ECZ.COMPRESSION.LEVEL
ECZ.MEMORY.LEVEL	ECZ.WINDOWSIZE	ESF.WAIT
EXPDT	GDGALLOC	GDGENQ
INVOKE.ALLOC.EXIT	INVOKE.ALLOC.EXIT.ON.RESTART	MAX.AGE
MAX.AGE.TOD	MAXRETRIES	MAXSTGIO
MULTI.COPY.STAT.RCD	NETMAP.CHECK.ON.CALL	PDSE.SHARING
PDSENQ	PRTYDEF	REQUEUE
RESET.ORIGIN.ON.SUBMIT	REUSE.SESIONS	ROUTCDE.CRIT
ROUTCDE.NORM	ROUTCDE.TAPE	RUNTASK.RESTART
SNMP	SNMP.DSN	SNMP.MANAGER.ADDR
SNMP.MANAGER.PORTNUM	STAT.EXCLUDE	SYSOUT
TAPE.PREMOUNT	TCP.API.TIMER	TCP.CONNECT.TIMEOUT
TCQ.THRESHOLD	THIRD.DISP.DELETE	TRANS.SUBPAS
WTMESSAGE	WTRETRIES	ZEDC

ZEDC.EXCLUDE	ZFBA	ZIIP.EXTCOMP.DATASIZE.THRES HOLD
--------------	------	-------------------------------------

The MODIFY INITPARMS command has the following format.

Label	Command	Parameter
(optional)	MODify INITparms	

This command has no parameters.

Using the MODIFY INITPARMS Command from the Batch Interface

About this task

To use the MODIFY INITPARMS command features from the batch interface: 1. 2.

Procedure

1. Update the initialization parameters in the initialization parameter data set.
2. Place the MODIFY INITPARMS command in a batch job stream.
3. Submit the job while IBM Connect:Direct is running. A message is displayed indicating the results of the refresh action.

Note: You must set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

Issuing the MODIFY INITPARMS Command through the IUI

About this task

To use the MODIFY INITPARMS command features through the IUI:

Procedure

1. Update the initialization parameters in the initialization parameter data set.
2. Request option MD from the Connect:Direct Administrative Options Menu to access the MODIFY COMMAND screen.

node.name	MODIFY COMMAND		14:34
CMD ==>			
Server ==>	00000000	(Current DEBUG Settings)	
MODIFY DEBUG	==> -----	(nnnnnnnn)	
MODIFY BITS.ON	==> -----	(nnnnnnnn)	
MODIFY BITS.OFF	==> -----	(nnnnnnnn)	
MODIFY DDNAME	==> -----	(ddname,nn)	
MODIFY CLOSE	==> -----	(ddname)	
MODIFY MODDIR.TRACE	==> ---	(YES)	
MODIFY DYN ==>			
MODIFY SESSIONS	==> ()	(Quiesce or Resume)	NODE ==> -----
MODIFY NODE.TRACE.ON	==> ()		
MODIFY NODE.TRACE.OFF	==> -----		
MODIFY NODES	==> ---	(YES)	
MODIFY INITPARMS	==> ---	(YES)	
MODIFY ZIIP	==> -----	(ALL NONE EXTCOMP SSLTLS PROJECT)	
MODIFY ZFBA	==> ---	(YES or NO)	

3. Type YES in the MODIFY INITPARMS field.
4. Press ENTER. A report is displayed indicating the results of the action.
5. Review the report and perform any corrections if necessary.

6. Press ENTER to clear the report.

Repeat the procedure if you updated the initialization parameter data set.

Backing up the Global and Local Initialization Parameter Files

Control Center has the ability to configure global initialization parameters for stand-alone IBM Connect:Direct servers and local initialization parameters for members of a IBM Connect:Direct/Plex environment. As part of the possible changes, the original INITPARM member is overwritten with the updates Control Center makes.

About this task

You can save a copy of the global initialization parameter file and local initialization parameter files using the INITPARM.BACKUP and CDPLEX.INITPARM.BACKUP initialization parameters. These backup files protect against inappropriate or failed updates from Control Center or other facilities capable of updating initialization parameters. IBM Connect:Direct will use these members to complete a backout of failed updates. This also provides you with the ability to manually back out using the specified backup members which contain the last set of initialization parameters used to successfully start IBM Connect:Direct.

Note: If you commented out the members names for the backup initialization parameters, specify new names.

Procedure

1. Specify the name of the backup member using the global initialization parameter, INITPARM.BACKUP = member. In a IBM Connect:Direct/Plex environment, you also specify the local initialization parameter, CDPLEX.INITPARM.BACKUP = member for the IBM Connect:Direct Plex/Manager and each IBM Connect:Direct/Plex server.
2. Start Connect:Direct for z/OS.

When IBM Connect:Direct initializes successfully, it places a copy of the initialization parameter member in the backup member specified in the INITPARM.BACKUP parameter (and does the same thing for each IBM Connect:Direct/Plex-related backup parameter). If ISPF STATS are ON for the PDS, the statistics for the initparms are also copied to the backup member. If ISPF STATS are not ON, the backup member will have ISPF statistics generated based on the current date and time.

Space Requirements for Using the Backup Feature

The PDS containing the initialization parameters must have enough directory and disk space for the following items:

- The backup members for the global initialization parameter file and local initialization parameter files
- Changes to the global and local initialization parameters
- ISPF formatted statistics

If the initparm PDS runs out of space, an Sx37 (or similar) ABEND occurs and the system attempts to back out the changes by copying the initparm backup file (including the ISPF STAT information). To recover space inside the PDS, compress the PDS prior to starting IBM Connect:Direct.

Manually Backing Out or Restoring

About this task

In the event a manual backout or restore of the original initialization parameters is required, follow this procedure:

Procedure

1. Restore the damaged global initialization parameter member using the backup global initialization parameter member. Do the same thing for the backup initialization parameter members for all IBM Connect:Direct/Plex members.
2. Start Connect:Direct for z/OS.

Displaying IBM Connect:Direct/Plex Status

The INQUIRE CDPLEX command displays IBM Connect:Direct/Plex status information. It has the following format.

Label	Command	Parameter
(optional)	INQUIRE CDPLEX	

The INQUIRE CDPLEX command has no parameters.

Example

The following figure shows a partial sample report.

=====			
Inquire CDPLEX	mm.dd.yyyy	hh:mm:ss	
=====			
XCF Group Name	: TPXCFGRP	When Activated	: mm.dd.yyyy hh:mm:ss
SYSPLEX Sys Name	: CSGB	JOB/STC Name	: CD\$MGR
Active Servers	: 1	Maximum Servers	: 32
Server	: SERVER1	When Activated	: mm.dd.yyyy hh:mm:ss
SYSPLEX Sys Name	: CSGB	JOB/STC Name	: CD\$SRV1
Active Processes	: 0	Maximum Processes	: 250
Server Supports	: CTCA SNA IPv4		
Server PLEXCLASSES:	(A B 1 *)	

Using the INQUIRE CDPLEX Command from the Batch Interface

Procedure

1. Place your command in a batch job stream.
2. Submit the job while IBM Connect:Direct is running.

A report is displayed.

Note: You must set the fifth character of the DGADBATC output parameter specification to a Y to print the result of the command that is in the temporary data set.

Issuing the INQUIRE CDPLEX Command through the IUI

Procedure

1. Select option INQ from the Administrative Options Menu. The Inquire DTF Internal Status screen is displayed.
2. Select the IPLX option.
3. Press **ENTER**.

A report is displayed.

Stopping IBM Connect:Direct

The STOP CD command stops IBM Connect:Direct through one of five types of shutdowns:

- Force
- Immediate
- Step
- Quiesce
- Run Task Immediate

This command is usually used for system maintenance.

If you are running IBM Connect:Direct/Plex, shut down the individual IBM Connect:Direct/Servers or the IBM Connect:Direct/Plex environment. Specify if the extended recovery standby system processes the work performed by the system being shut down.

The STOP CD command has the following format and parameters.

Label	Command	Parameters
(optional)	STOP CD	[Force Immediate Quiesce Runtaskimm Step]
		[CDPLEX WHERE (SERVER=server name)]
		RECOVER

The parameters for the STOP CD command are:

Parameter	Description
Force	Stops IBM Connect:Direct through a user U4082 ABEND, and produces a dump. Use this option only when problems occur.
Immediate	<p>Terminates all active transmissions immediately after any executing Run Task Processes complete. IBM Connect:Direct writes the statistics record, closes the files, and shuts down. All Processes resume execution when IBM Connect:Direct is reinitialized. If a Process is set for checkpointing and IBM Connect:Direct stops with this parameter, IBM Connect:Direct starts from the last checkpoint and resumes transferring data when the Process resumes. IMMEDIATE is the default.</p> <p>Note: You can change how the Immediate parameter interprets the shutdown command by using the IMMEDIATE.SHUTDOWN initialization parameter. If IMMEDIATE.SHUTDOWN=I (the default), an immediate shutdown functions as described in the preceding paragraph. However, if IMMEDIATE.SHUTDOWN=R, an immediate shutdown functions as a runtaskimm shutdown, terminating any executing Run Task Processes before shutting down IBM Connect:Direct.</p> <p>Refer to <u>IMMEDIATE.SHUTDOWN = I R (I, nnn 60) (R, nnn 60)</u> for more information.</p>
Quiesce	Enables all active transmissions to run until all executing Process steps complete. No new transmissions are started, and no additional Processes are accepted. All interactive sessions are terminated except for the issuer of the STOP CD command. All active Processes must complete and then you must sign off before IBM Connect:Direct stops.
Runtaskimm	Terminates any Run Task Processes before stopping IBM Connect:Direct. After the Processes are terminated, IBM Connect:Direct writes the statistics record, closes the files, and shuts down. This parameter is provided because the Immediate parameter does not terminate a Run Task until it reaches an interrupt point, such as a checkpoint. A long-running Run Task could delay IBM Connect:Direct shutdown until it completes. When IBM Connect:Direct restarts, if the RUNTASK.RESTART initialization parameter is YES, the checkpoint records for the terminated Run Task restart the Run Task.
Step	Enables active transmissions to run until the current step of each executing Process finishes. IBM Connect:Direct then writes the statistics records, closes the files, and shuts down. All Processes resume execution when IBM Connect:Direct is reinitialized.

Parameter	Description
CDPLEX	Shuts down the entire IBM Connect:Direct/Plex environment. You cannot use this parameter in a IBM Connect:Direct/Stand-alone Server.
WHERE (SERVER=server name)	<p>Specifies which IBM Connect:Direct/Server in a IBM Connect:Direct/Plex environment to shut down. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter. You cannot use this parameter in a IBM Connect:Direct/Stand-alone Server.</p> <p>Use this parameter if you only want to shut down a particular IBM Connect:Direct/Server, but leave the rest of the IBM Connect:Direct/Plex environment running. (Use the INQUIRE CDPLEX command described in “Displaying IBM Connect:Direct/Plex Status” on page 173 to find the name of a server.)</p> <p>Note: When you shut down a IBM Connect:Direct/Plex environment, you must specify CDPLEX. If you are shutting down a server, you must also specify WHERE(SERVER=).</p>
RECOVER	Specifies if the extended recovery standby system continues processing work from the system that is shutting down.

Examples

The following example stops IBM Connect:Direct, and terminates all transactions immediately.

```
STOP CD I
```

The following example stops an entire IBM Connect:Direct/Plex environment after all Processes are complete.

```
STOP CD Q CDPLEX
```

The following example force stops a IBM Connect:Direct/Server named WALTER, but continues processing on the extended recovery standby system.

```
STOP CD F WHERE(SERVER=WALTER) RECOVER
```

Stopping IBM Connect:Direct through the Batch Interface

About this task

To use the STOP CD command from the batch interface:

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify the results.

Stopping IBM Connect:Direct through the IUI

About this task

To issue the STOP CD command through the IBM Connect:Direct IUI:

Procedure

1. Select SN from the Connect:Direct Administrative Options Menu.

node.name	STOP Connect:Direct	hh:mm
CMD ==>		
Q ==> Continue active transmissions until the end of process		
S ==> Continue active transmissions until the end of a step		
I ==> Immediately stop all active transmissions (wait for RUN TASKS to complete)		
R ==> Immediately stop all active transmissions (do not wait for RUN TASKS to complete)		
F ==> Force Connect:Direct to stop via an ABEND		
EXTENDED.RECOVERY and C:D/Plex Options:		
Server ==>	_____ (C:D/Plex server name or '*' for entire C:D/Plex)	
Recover =>	___ (Yes No) Should EXTENDED.RECOVERY standby takeover?	

2. Type one of the five options on the command line.
Refer to [“Stopping IBM Connect:Direct” on page 173](#) for a description of these options. Quiesce (Q) is the default.
3. If you want to shut down a IBM Connect:Direct/Server in IBM Connect:Direct/Plex environment, type the server name in the SERVER field.

If you want to shut down the entire IBM Connect:Direct/Plex environment, leave the SERVER field blank.
4. If you want the extended recovery standby system to continue processing work, type Yes in the RECOVER field. Type No or leave the field blank if you do not want the extended recovery standby system to continue processing work.
5. Press **ENTER**.

A shutdown message is displayed for Immediate, Quiesce, and Step shutdowns. No message is displayed for Force shutdowns.

Suspending and Resuming Quiesce and Trace Settings

IBM Connect:Direct uses a node table to manage Quiesce and Trace settings for the adjacent nodes in the Netmap. When IBM Connect:Direct is initialized, it adds all Netmap adjacent node records to a new node table. It applies the Quiesce and Trace INITPARMs to the node table entries. After IBM Connect:Direct is initialized, node table entries are added or deleted by Netmap updates.

If a Netmap update adds a new node, the node is added to the node table with TRACE OFF and QUIESCE OFF. If a Netmap update does a \$\$REPLACE or a \$\$DELETE followed by a \$\$INSERT in the same run, any pre-existing node table settings for that node are preserved. If a Netmap update does a \$\$DELETE and a subsequent Netmap update run adds the node back, the node is added to the node table with TRACE OFF and QUIESCE OFF.

The Modify Nodes screen and the output from the INQUIRE DEBUG command list all of the adjacent nodes in the node table.

For information about Trace and Debug settings, refer to [IBM Connect:Direct Function Traces](#).

Use the MODIFY SESSIONS command to suspend processing on a single node, multiple nodes, or all nodes, and to resume processing on any or all suspended nodes. For example, you can suspend processing on a node because of problems but leave other nodes operating. Another example would be the suspension of a node that you know will be down for some time.

Note: Node-level MODIFY commands only apply to existing nodes in the node table. If you add a node using Netmap update, enter a node-level MODIFY command to change that node's default settings.

The MODIFY SESSIONS command has the following format and parameters.

Label	Command	Parameters
(optional)	MODIFY	SESSIONS = Quiesce Resume (WHERE(NODE=node name))

The following table describes the parameters of the MODIFY SESSIONS command.

Parameter	Description
SESSIONS = Quiesce Resume (WHERE(NODE=node name))	<p>Controls the automatic establishment of DTF-to-DTF sessions.</p> <p>Quiesce specifies that no new DTF-to-DTF sessions are started after executing Processes complete. Interactive users can sign on. Any Processes that normally execute are placed in the WAIT queue.</p> <p>Resume terminates a quiesce state and returns to normal operation.</p> <p>The WHERE(NODE=) parameter enables you to suspend or resume processing on a single node, multiple nodes, or all nodes. You can use this parameter in the following circumstances:</p> <ul style="list-style-type: none"> • To suspend processing on specific nodes because of problems, but allow other nodes to continue processing. • If you know that a node will be down for some time • To suspend or resume processing on all nodes. <p>The node name subparameter is the 1–16 character local node name specified in the network map of the affected node. You can also use the * (for a string) and ? (for an individual character) wildcard characters to specify a generic node name. For example, the following command suspends processing on all node names that begin with NODE.CHICAGO.</p> <p>SESSIONS=QUIESCE (WHERE(NODE=NODE.CHICAGO*))</p> <p>Note: If you use the * wildcard, and it is not the last character, you must put the entire node name in single quotes. Otherwise, the node name is truncated at the first *, and more nodes may be selected than was intended. For example, node names starting with NODE.CHICAGO can be specified either as NODE.CHICAGO* or 'NODE.CHICAGO*'. But node names containing NODE.CHICAGO must be specified as '*NODE.CHICAGO*', not *NODE.CHICAGO* which would be functionally equivalent to all nodes since the first character is *.</p> <p>You can specify * as the node name to suspend or resume processing on all nodes. For example, the following command suspends processing on all nodes.</p> <p>SESSIONS=QUIESCE (WHERE(NODE=*))</p> <p>If you omit the WHERE(NODE=) parameter, the command applies to the entire IBM Connect:Direct system. However, a system-wide RESUME command does not override the processing of any individually suspended nodes. You must issue the SESSIONS=RESUME command with WHERE(NODE=*) to resume processing on individually suspended nodes.</p> <p>Note: If the command is issued on an SNODE to quiesce processing with a PNODE, the session with the PNODE is established. However, as soon as the PNODE node name is determined, the session is terminated. No processing of data occurs.</p>

Suspending or Resuming Processing on a Node through the Batch Interface

Procedure

1. Place the MODIFY SESSIONS command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify the results by issuing an INQUIRE DEBUG command.

Note: For information about INQUIRE DEBUG, see [Displaying DEBUG Settings](#).

The following example suspends processing on all nodes that begin with NODE.CHICAGO.

```
SESSIONS=QUIESCE (WHERE(NODE=NODE.CHICAGO*))
```

The resulting output from the INQUIRE DEBUG command follows.

```
=====
NODE.NEWYORK      *INQ DEBUG/QUIESCE* DATE: mm.dd.yyyy  TIME: hh:mm:ss
SYSTEM INITIALIZED -----(0000)----- mm.dd.yyyy  hh:mm:ss
=====

DEBUG            => '00200000'
QUIESCE          => No
TCQ DSN          => PROD.CD.ABC.TCQ
TCX DSN          => PROD.CD.ABC.TCX
TCQ Threshold    => No
TCQ File         0% Full. Max.# CI:      1000  # Used CI:      0

NODE TABLE      => NODE NAME      QUIESCE  DEBUG
NODE ENTRY 1    => CD.BOSTON       No      B0BB0BB0
NODE ENTRY 2    => CD.CHICAGO1     Yes
NODE ENTRY 3    => CD.CHICAGO2     Yes
NODE ENTRY 4    => CD.DFW.LU0      No
NODE ENTRY 5    => CD.DFW.LU62     No
NODE ENTRY 6    => CD.DFW.UDT      No
```

Suspending or Resuming Processing on a Node through the IUI

Procedure

1. Request option MD from the Connect:Direct Administrative Options Menu to access the MODIFY COMMAND screen.
2. Type Q in the MODIFY SESSIONS field to suspend processing.
Type R in the MODIFY SESSIONS field to resume processing on a suspended node.
3. Type the 1–16 character node name in the NODE field. (If you use the * wildcard, and it is not the last character, put the entire node name in single quotes.)
4. Press **ENTER**.

The results will be displayed, for example, MODIFY SESSIONS QUIESCE successful.

Suspending or Resuming Processing on Nodes through the Modify Nodes Screen

To suspend or resume processing on nodes through the Modify Nodes screen:

Procedure

1. Request option MD from the Connect:Direct Administrative Options Menu to access the MODIFY COMMAND screen.
2. Type YES in the MODIFY NODES field.

3. Press **ENTER**. The Modify Nodes screen is displayed.

CD.PR.D		MODIFY NODES (Unsaved Changes)		Row 1 of 1054
CMD ==>		SCROLL ==> PAGE		
(CMD commands: CLEAR DEBUG/TRACE QUIESCE RESUME; CANCEL END REFRESH SAVE)				
(SEL commands: C=Clear Q=Quiesce R=Resume. Overtyp e Debug setting.)				
SEL	Node Name	Quiesce	Debug	Chg
---	-----	-----	-----	---
	CD.PR.D.CSGA	OFF	11111111	
	CD.PR.D.CSGB	OFF	11111111	
	CD.PR.D.CSGD	OFF	11111111	
	CD.PR.D.CSGE	OFF	11111111	
	CD.PR.D.CSGF	ON	11111111	Y
	CD.PR.D.CSGG	OFF	11111111	
	CD.PR.D.VIPA	OFF	11111111	
	CD.TST	OFF	FFFFFFEF	
	CD.TST. %%%/%%%	OFF	FFFFFFEF	
	CD.TST.CSGA	ON		Y
	CD.TST.CSGB	OFF	FFFFFFEF	
	CD.TST.CSGD	OFF	FFFFFFEF	
	CD.TST.CSGG	OFF	FFFFFFEF	
	CD.TST.CTCA1	OFF	FFFFFFEF	
	CD.TST.CTCA2	OFF	FFFFFFEF	
	CD.TST.CTCA3	OFF	FFFFFFEF	

4. Use one of the following methods to suspend processing of specific nodes:

- Type Q in the SEL column next to the matching node names. The Chg field for those nodes will display Y.
- From the CMD line, use the QUIESCE nodename command to turn the Quiesce setting on for the rows with matching nodenames.

Note: You can use the * (for a string) and ? (for an individual character) wildcard characters to specify a generic node name. Unlike the MODIFY SESSIONS command, you do not have to enclose the node name in single quotes if you use the * wildcard and it is not the last character. In fact, if you do, as a result, there will probably not be any matches since quotes are treated like any other character.

Use one of the following methods to resume processing of a specific node:

- Type R in the SEL column next to the matching node names. The Chg field for those nodes will display Y.
- From the CMD line, use the RESUME nodename command to turn the Quiesce setting off for the rows with matching nodenames. Refer to the [Suspending and Resuming Quiesce and Trace Settings](#) for information about the RESUME command.

5. Type SAVE in the CMD field and press **ENTER** to save your changes.

Modify Nodes Screen

Use the Modify Nodes screen to Quiesce or Resume processing for all nodes or specified nodes. (To access the Modify Nodes screen, follow Steps 1 through 3 in [Suspending or Resuming Processing on Nodes through the Modify Nodes Screen](#).) You can also use the Modify Nodes screen to modify debug and trace settings for all nodes or specific nodes. See [Modify Trace and Debug Settings Through the Modify Nodes Screen](#) for more information.

CMD Commands

CMD commands are entered in the CMD input field. Unknown commands and syntax errors will result in error messages. CMD commands operate on all nodes in the node table, not just the rows that are currently being displayed.

The CMD input field accepts the following commands:

CMD	Description
CANCEL	Unconditionally exits the Modify Nodes screen, discarding any changes. You can abbreviate CANCEL to CAN, CANC or CANCE.

CMD	Description
END	Exits the Modify Nodes screen if there are no unsaved changes outstanding. If there are unsaved changes, the END command causes the dialog to issue an error message and redisplay the ISPF table.
SAVE	Commits all changes en masse to the DTF. It then gets a new copy of the node table from the DTF. Also, all the ISPF table change indicators are cleared.
CLEAR nodename	Clears the quiesce and debug settings for the rows with a matching node name. It is equivalent to doing both a DEBUG OFF and a RESUME against the node name. The nodename operand can have any number of * or ? wildcard characters in any position.
DEBUG <i>nodename</i> ON <i>debug-bits</i>	Turns on debugging and completely replaces the Debug setting with the debug-bits operand for the rows with a matching node name. This overrides the global DEBUG setting for the individual nodes. 00000000 is a valid debug-bits value and is not the same as DEBUG OFF. The nodename operand can have any number of * or ? wildcard characters in any position.
DEBUG <i>nodename</i> OFF	Turns off debugging and blanks out the debug setting for the rows with a matching nodename. This allows the nodes to revert to using the global DEBUG setting. The nodename operand can have any number of * or ? wildcard characters in any position.
DEBUG <i>nodename</i> BITSON <i>debug-bits</i>	Turns on the bits in the debug-bits operand without changing any other bits in the row's debug setting, but only if the rows with a matching node name have a non-blank debug setting. It does not turn debugging on or off. Though 00000000 is a valid debug-bits value, if it is specified in DEBUG BITSON, the command has no effect. The nodename operand can have any number of * or ? wildcard characters in any position.
DEBUG <i>nodename</i> BITSOFF <i>debug-bits</i>	Turns off the bits in the debug-bits operand without changing any other bits in the row's debug setting, but only if the rows with a matching node name have a non-blank debug setting. It does not turn debugging on or off. Though 00000000 is a valid debug-bits value, if it is specified in DEBUG BITSOFF, the command has no effect. The nodename operand can have any number of * or ? wildcard characters in any position.
QUIESCE <i>nodename</i>	Turns the Quiesce setting on for the rows with a matching nodename. See the CLEAR command above for the description of the nodename operand. The nodename operand can have any number of * or ? wildcard characters in any position.
REFRESH	Discards any pending changes and gets a new copy of the node table from the DTF. It is the equivalent of doing a CANCEL, and then reentering the dialog.
RESUME <i>nodename</i>	Turns the Quiesce setting off for the rows with a matching nodename. See the CLEAR command above for the description of the nodename operand. The nodename operand can have any number of * or ? wildcard characters in any position.

CMD	Description
TRACE <i>nodename</i> BITSOFF <i>debug-bits</i>	TRACE is an alias of DEBUG. They have the same syntax and can be used interchangeably.

SEL commands are entered in the SEL input field for each row. Change the Debug field on each row by overtyping. You can update as many rows as are displayed when you press Enter.

The SEL input field accepts the following line commands:

- C - Clear all settings for the node (same as Debug OFF plus RESUME)
- Q - Quiesce the node (turn Quiesce ON)
- R - Resume the node (turn Quiesce OFF)

Type over the Debug field in any row to change its value. The Debug field accepts the same debug bits as the DEBUG command. Blanking out the Debug field in a row is equivalent to issuing a DEBUG OFF on that nodename.

If you change the Debug field to a non-blank value in a specific row, issue a CMD command for the same row, and press Enter, only the CMD command will take effect.

If you enter a CMD command and one or more SEL commands, when you press Enter, the SEL commands are processed before the CMD command.

Node Table Processing

When you issue a CMD command, a SEL command, or change the Debug field, the node table is not updated until you enter the SAVE command. Until then, MODIFY commands generated by these actions are stacked in first-in-first-out (FIFO) order in a MODIFY command queue.

You can enter a SAVE command only if one or more nodes are flagged as changed. If you change a node and then change it back before you issue the SAVE command, the change flag is cleared. If you back out all node changes, the MODIFY command queue is cleared. However, if any rows remain flagged as changed, SAVE sends all MODIFY commands queued since the last time there were no changed rows.

If multiple IUI users work on a copy of the node table simultaneously, it can be difficult to determine what changes are in effect. If you suspect that another user has changed the node table, discard your changes and retrieve a fresh copy of the node table by issuing the REFRESH command.

Global Signon Defaults

The IBM Connect:Direct administrator can define global signon defaults so that users do not have to individually alter their signon default values to increase their allocation. By implementing a global signon default, you can avoid insufficient space being allocated for the temporary data set upon signon as well as SB37 ABENDs.

A temporary data set (TEMP DSN) is allocated at SIGNON to the IUI, which uses the system default allocation parameters. At times, this allocation is insufficient causing SB37 failures if a large amount of data is returned for a command request. The facilities available via SIGNON defaults (SD) are not global and each individual user must alter their default values to increase the allocation.

To change the default values, use member DGAXCXDF from \$CD.SDGASAMP to assemble and link module DGAXCXDF with the new values for TEMP DSN. You can edit and submit \$CD.SDGAJCL member DGAJCXDF to do the assembly and link edit. Once the module is assembled and linked, all user who sign on via an API will be subject to the new default global signon settings.

After you implement global signon defaults, allocation parameters during SIGNON can come from the following three sources, which are listed from lowest to highest precedence:

- Defaults from macro DGA\$UICB specified by programs such as DGADBATC, DGADCHLA, and DGADCMDP. This is considered a legacy source of SIGNON allocation parameters.

Note: Users including the administrator, do not have control over these programs.

- The SIGNON command, whose parameters come from wherever the invoker of the command decides, for example, the IUI can save SIGNON parameters in a user's ISPF profile dataset. Like the DGA\$UICB macro, this is considered a legacy source of SIGNON allocation parameters.
- The DGAXCXDF load module, which the administrator can control.

DGA\$XDEF Syntax and Parameters

\$CD.SDGASAMP member DGAXCXDF consists of one statement, the macro DGA\$XDEF, in \$CD.SDGAMAC and should not be changed. To modify the DGAXCXDF member, first copy it and then alter it to suit your installation's requirements.

The following shows the DGA\$XDEF syntax and defaults:

name	DGA\$XDEF	OVERWRITE=ALLOW,	/* SIGNON allowed to override? */	X
		ALOCTYPE=CYL,	/* ALLOCATION TYPE CYL,TRK	X
		ALOCPRI=1,	/* ALLOCATION PRIMARY SPACE	X
		ALOCSEC=1,	/* ALLOCATION SECONDARY SPACE	X
		ALOCUNIT=,	/* ALLOCATION UNITNAME	X
		ALOCVOL=	/* ALLOCATION VOLSER	X

The following table lists the parameters that make up the DGA\$XDEF load module:

Parameter	Description
OVERWRITE	Specifies when a legacy source can override the global signon defaults assigned by the DGA\$XDEF load module. If not specified, defaults to ALLOW (a null value is invalid). Values are: <ul style="list-style-type: none"> • DISALLOW— DGA\$XDEF always supplies all allocation parameters. • ALLOW—A legacy source can override DGA\$XDEF parameter by parameter. • ALLOWGT—A legacy source can override DGA\$XDEF if the maximum space it would allocate is greater than the maximum space that DGA\$XDEF would allocate. Otherwise, DGA\$XDEF overrides the legacy source. With this setting, whichever source can supply more space will be used. Note that, unlike ALLOW, the winning source in ALLOWGT is required to supply all allocation parameters. • ALLOWLT—A legacy source can override DGA\$XDEF if the maximum space it would allocate is less than the maximum space that DGA\$XDEF would allocate. Otherwise, DGA\$XDEF overrides the legacy source. With this setting, whichever source limits the maximum space more is used. Note that, like ALLOWGT, the winning source in ALLOWLT is required to supply all allocation parameters.
ALOCTYPE	Specifies the allocation type. If not specified, defaults to space allocation by cylinder—CYL (a null value is invalid).
ALOCPRI	Specifies the primary allocation of storage. If not specified, defaults to a o1.
ALOCSEC	Specifies the secondary allocation of storage. If not specified, defaults to a o1.
ALOCUNIT	Specifies the unit type of the TEMP DSN. If not specified, has no default.
ALOCVOL	Specifies volume serial number of the TEMP DSN. If not specified, has no default.

The other DGA\$XDEF macro parameters are converted into the well known JCL space parameters. All DGA\$XDEF macro parameters other than OVERWRITE are converted into the JCL SPACE parameters and can take any valid value that can be specified in JCL. The one exception is that the ALOCTYPE parameter supports only TRK (Track) and CYL (Cylinder).

ISPF Messages Displayed During Signon

If the DGAXCXDF load module was successfully located at IUI SIGNON, new ISPF messages are displayed upon entry to the IUI Signon Defaults (SD) panel. (If DGAXCXDF was not present at IUI SIGNON, no ISPF message is displayed.) Both the short and long message components are generated. The exact text of the ISPF messages is based on:

- Settings in the DGAXCXDF load module, at the time of SIGNON
- The user's ISPF profile variables for the temporary data set

The user will see one of the following ISPF short messages, which states what the DGAXCXDF OVERRIDE setting was when they signed on:

- OVERRIDE=DISALLOW
- OVERRIDE=ALLOW
- OVERRIDE=ALLOWGT
- OVERRIDE=ALLOWLT

The ISPF long message component provides the following information:

- The values for the current temporary data set allocation
- An explanation of where the values came from and why

The following is an example of a typical ISPF long message example:

```
The current Temporary Data Set space settings are: ALLOCATION
TYPE=CYL, PRIMARY SPACE=00000001, SECONDARY SPACE=00000001, UNIT
TYPE=, VOL=SER=. These are from the Global Signon Default
module DGAXCXDF, because OVERRIDE=ALLOWGT is specified, and the
maximum space specified by this panel/DGA$UICB was not greater
than that specified by DGAXCXDF.
```

Note: In the message above, the phrase, this panel/DGA\$UICB, is the same thing as saying a legacy source of SIGNON allocation parameters.

Implementing Security

IBM Connect:Direct provides a range of security options to meet diverse security requirements. These options can be part of IBM Connect:Direct, part of interfaces to other security software, sample exits, or available from user-customized exit routines.

IBM Connect:Direct provides support for passwords and/or passphrases up to 64 characters. Password can also mean passphrase in most instances with exception when referencing dataset passwords.

When PROCESSES supply PNODEID and/or SNODEID passwords, password longer than 8 characters cannot be used when the PROCESS is using PNODE-SNODE, SNA or SNUF protocols.

Note: Be aware that when a user signs on to Connect:Direct using a passphrase (a password that is longer than 8 characters), all PROCESSES submitted must override the PNODEID and SNODEID userids and passwords if the process is to use PNODE=SNODE, SNA or SNUF protocols (these protocols only support passwords of 8 characters or less).

Note: All sample exits define the proper AMODE and RMODE settings within the source member themselves. All user exits should be link-edited with AMODE=ANY and capable of executing in 31-bit mode. Each user exit should preserve the mode in which it was invoked and return to the caller in the proper mode. Modules written to execute in 31-bit mode can be link-edited with RMODE=ANY or RMODE=24. Check the source for the sample exits to see how IBM Connect:Direct defines the proper AMODE and RMODE settings.

Connect:Direct for z/OS provides the following security features:

Security Option	Description
Security exits	Secures signon processing, job streams, and application programs. IBM Connect:Direct provides four security exits and includes samples in the sample library. See Security Exits for more information.
SECURITY.EXIT initialization parameter	Specifies a stage 2 security exit. This exit is invoked during signon and Process start and data set access. Signon or file access requests are passed directly to the security exit for authorization checking. For more information, see “SECURITY.EXIT SECURITY” on page 477 .
IBM Connect:Direct Authorization Facility	Provides signon security and assigns IBM Connect:Direct functional authority if you do not specify or comment out the SECURITY.EXIT initialization parameter. Use this facility if your installation does not have a security package. See IBM Connect:Direct Functional Authority for information. Note: The IBM Connect:Direct Authorization Facility provides no data set access security checking. Authorization File describes the User Authorization file in detail.
IBM Sterling Connect:Direct Secure Point-of-Entry	Secures the entry of an outside user to your system. Point-of-entry processing occurs before security exits are called. See Sterling Connect:Direct Secure Point-of-Entry for more information.
Trusted Node Security	Enables you to enforce more restrictive security parameters on specific nodes in your network. For example, each adjacent node can be defined as internal or external in its relationship to the local node of that network map. See Trusted Node Security for more information.

IBM Connect:Direct supports the following security options:

Security Option	Description
Connect:Direct Secure Plus	Provides enhanced security for IBM Connect:Direct. It uses cryptography to secure data during transmission. You select the security protocol, cipher suites, and other encryption options to use with the Connect:Direct Secure Plus product. One such option is Strong Password Encryption (SPE), which you can use to secure passwords at rest within the TCQ and AUTH files. SPE uses the TDESCBC112 encryption algorithm of Connect:Direct Secure Plus so if you have the Connect:Direct Secure Plus component configured, and then take the necessary steps to enable the SPE feature, SPE will be in effect. See the <i>IBM Connect:Direct Secure Plus for z/OS Implementation Guide</i> for more information.
CA-ACF2	External security package that secures files, users, and IBM Connect:Direct functions.
IBM® Resource Access Control Facility (RACF®)	External security package that secures files, users, and IBM Connect:Direct functions.
CA-TOP SECRET	External security package that secures files, users, and IBM Connect:Direct functions.
Firewall Navigation	Enables you to control access to a IBM Connect:Direct system running behind a firewall. See Configuring Firewall Navigation .

Security Exits

IBM Connect:Direct provides the following security exits:

- Stage 1 signon security exit
- Stage 2 security exit

- Run Job security exit
- Run Task security exit

The IBM Connect:Direct sample library provides the following security exit routines for use with CA-ACF2, IBM RACF, and CA-TOP SECRET. The High-Level Assembler is required to assemble the sample security exits.

Exit	Description
DGACXSIG	Stage 1 signon security exit interface
DGAXACRJ	RUN JOB security exit
DGAXACFT	RUN TASK security exit
DGAXRACJ	RUN JOB security exit for SAF or IBM RACF
DGAXRACT	RUN TASK exit
DGAMGSAF	Security Exit Stage 2
DGAXSAFT	RUN TASK security interface

The DGA\$SAFW macro in \$CD.SDGAMAC provides maps of the security and interface work area used by the security exits. This area allows for information that can be passed between the exits. IBM Connect:Direct has two major processing flows that invoke security exits, the SIGNON command sequence and the Process execution sequence. This section describes how security exits are invoked during these two Processes.

SIGNON Command Sequence

The SIGNON command sequence is the first flow through which a IBM Connect:Direct terminal user, console operator, or batch application gains access to IBM Connect:Direct functions. During this sequence, one or more of the following control points is invoked:

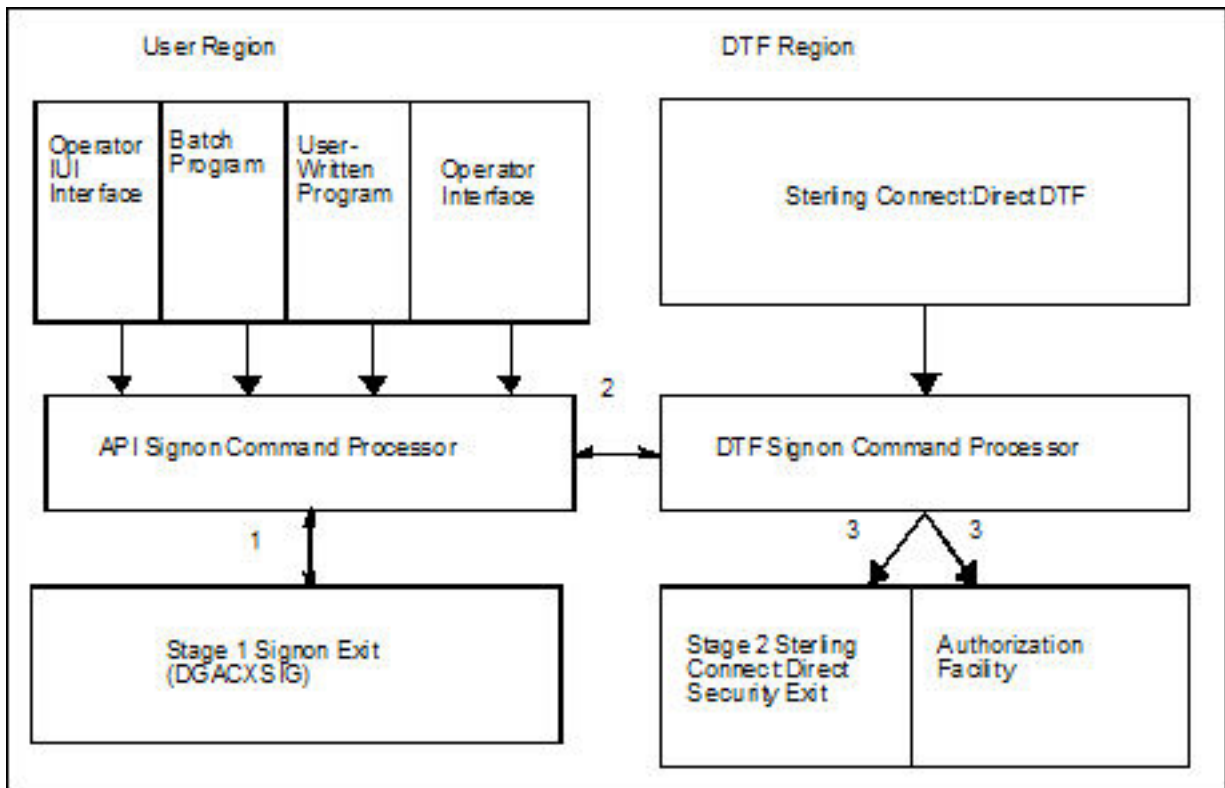
- Stage 1 signon security exit
- Stage 1 ESF signon Security exit
- IBM Connect:Direct Authorization Facility
- Stage 2 security exit

Security during Signon Command

When you execute a SIGNON command through the batch, interactive, or operator interface, security control points exist in the IBM Connect:Direct user region (or API) and the IBM Connect:Direct DTF region.

The stage 1 signon security exit is the initial control point, as shown in the following figure. This optional control point is a user exit that gains control in the region of the user. The exit can inspect and modify the SIGNON command parameters.

The next control point occurs in the DTF region and can be a stage 2 security exit or the IBM Connect:Direct Authorization Facility.



The following SIGNON command flow traces the security flow. The step numbers correspond to the steps in the illustration.

1. When you issue a SIGNON command, the API SIGNON command processor calls the stage 1 signon exit. If the stage 1 exit is not found, normal signon processing continues.

When invoked, the stage 1 exit receives a pointer to the User Interface Control Block (UICB) that contains information regarding the signon attempt. For a listing of UICB fields, refer to the chapter on the application programming interface in *IBM Connect:Direct for z/OS User Guide*.

If you specify a password on the SIGNON command, the stage 1 exit returns control to IBM Connect:Direct without making any modifications to the UICB, and signon processing proceeds. The stage 2 exit verifies the USERID and PASSWORD that are coded on the SIGNON command for system entry validation and all subsequent security calls.

If you do not specify a password on the SIGNON command, IBM Connect:Direct extracts the USERID from the security system control block built for this address space (when the TSO user logged on to TSO or when the BATCH job began execution) and puts that USERID into the UICB.

Note: Stage 1 exit keys off the password, not the user ID. So, if you do not specify a password but do specify a user ID, the stage 1 exit ignores that user ID and overlays it with the address space user ID that is picked up from the security system control block.

When the user ID is moved to the UICB, the exit fills in a special password of IUI, BATCH, or STC, depending upon what environment the signon comes from (IBM Connect:Direct cannot access the password for the address space user ID), and control returns to IBM Connect:Direct.

The benefit of running with a stage 1 signon exit is that IBM Connect:Direct batch jobs do not need hardcoded passwords in their SYSIN data streams.

The sample stage 1 exit is shipped with the dummy passwords of IUI, BATCH, and STC coded in the exit. Change these passwords for each installation to avoid the chance that another site is using the same dummy passwords. You can change these passwords by editing the source for DGACXSIG and the appropriate validation in the macro DGASECUR (for the stage 2 exit). If a user id has a security subsystem password that matches one of the dummy passwords, that user id will be unable to sign on to IBM Connect:Direct under some circumstances until the password is changed.

2. If the stage 1 processing is successful, the API SIGNON command processor passes the SIGNON command to the DTF where the DTF SIGNON command processor is invoked.
3. The DTF SIGNON command processor calls the stage 2 security exit or the IBM Connect:Direct Authorization Facility. The stage 2 exit recognizes special passwords of IUI, BATCH, and STC as being assigned by the stage 1 exit. All calls to the security system for verifications verify authorizations by user ID only.

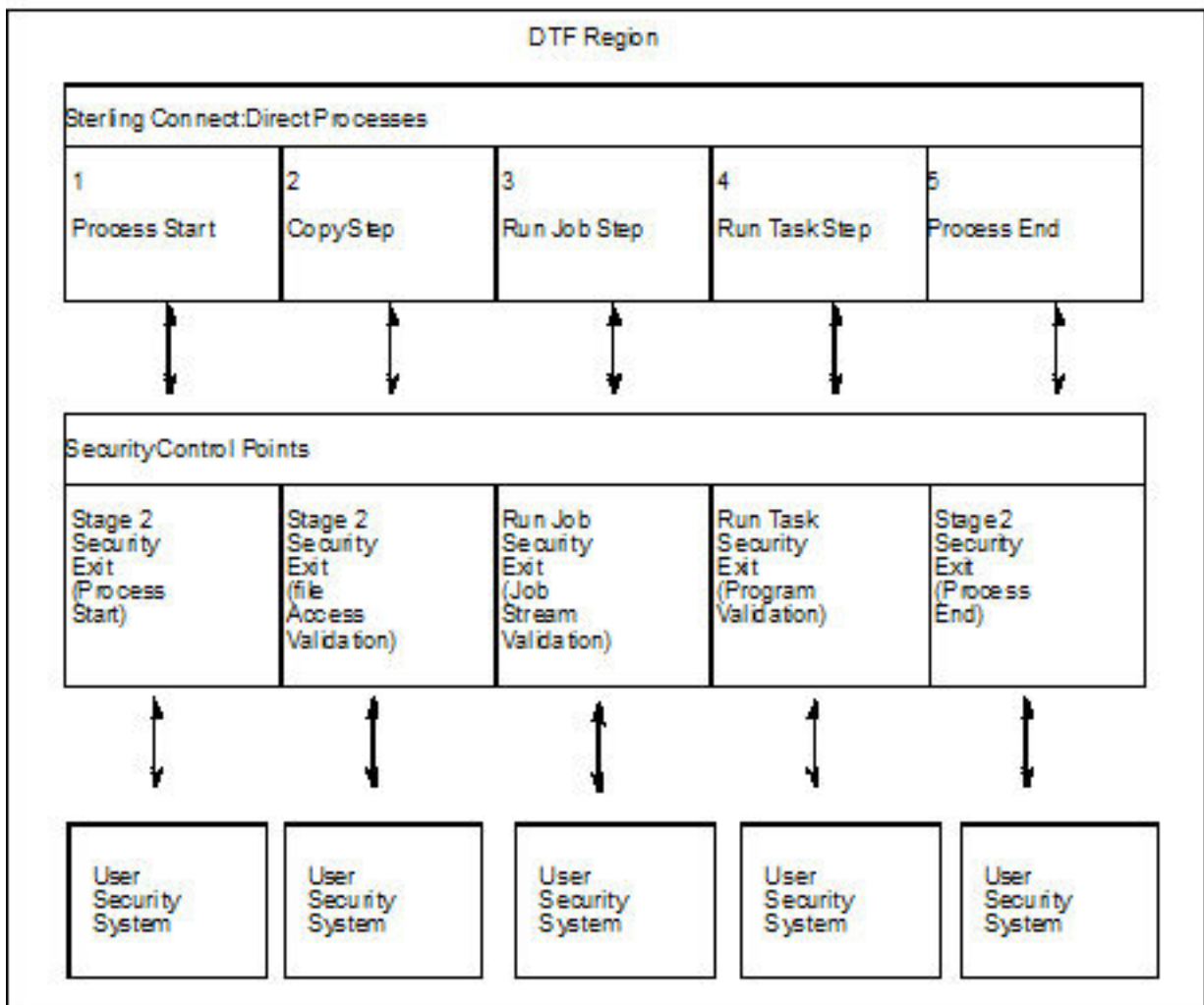
Regardless of how your system is implemented, this processing flow verifies the authority of the requesting user to perform IBM Connect:Direct functions by checking the ABM (Authorization Bit Mask) for this user. The ABM is built through the stage 2 security exit or through the IBM Connect:Direct Authorization Facility at signon and Process start. If signon to the DTF fails, then the Stage 1 ESF SIGNON exit (DGACXESF) is called to supply the ABM. If that exit is not found, a default ABM is used.

Process Execution Sequence

The Process execution sequence is the second flow through which IBM Connect:Direct services execute a user request. During this sequence, one or more of the following control points is invoked:

- Process start invokes the stage 2 security exit.
- Copy statement invokes the stage 2 security exit.
- Run Task statement invokes the Run Task security exit.
- Process end invokes the stage 2 security exit.
- Run Job statement invokes the stage 2 security exit or the Run Job security exit.

When IBM Connect:Direct executes a Process for a user, several DTF security control points exist, as shown in the following figure:



Refer to the numbers in the illustration as you trace the following Process flow:

1. Process start—This point in the stage 2 security exit gains control whenever a Process begins initial execution or restart execution, and enables verification of the authority of the requesting user to perform the IBM Connect:Direct functions contained in the Process.
2. File access—This point in the stage 2 security exit gains control during Process execution whenever a COPY or RUN JOB statement is encountered. It enables verification of the access of the user to read or write the file defined in the COPY statement.

With the RUN JOB statement, the exit enables verification of the user to read the file containing the job stream to be submitted.
3. Run Job—This exit point enables job stream validation and gains control when the following conditions exist:
 - RUN JOB statement is encountered during Process execution
 - RUN.JOB.EXIT initialization parameter is specified
4. Run Task—This exit point enables program validation and gains control when the following conditions exist:
 - RUN TASK statement is encountered during Process execution

- RUN.TASK.EXIT initialization parameter is specified
5. Process end—This point in the stage 2 security exit gains control whenever a Process terminates, whether normally or abnormally. This exit point assists in cleaning up the security resources involved in Process execution.

Note: Copy, Run Job, and Run Task exit functions are entered for every occurrence of the associated statement in a IBM Connect:Direct Process.

Implementing Security Exits

This section describes how to implement each type of security exit, and includes information for specific security environments such as, IBM RACF, CA-ACF2, and CA-TOP SECRET.

Note: You must have the High-level Assembler to assemble the sample exits.



CAUTION: To avoid out-of-storage ABENDS in Connect:Direct for z/OS, examine all user exits to verify that all obtained storage either via GETMAIN or STORAGE OBTAIN is freed. For each GETMAIN/STORAGE OBTAIN that an exit issues, the exit must also issue a corresponding FREEMAIN/STORAGE RELEASE to avoid accumulating storage. If an exit opens a file, you may need to issue a FREEPOOL after the file is closed.

Stage 1 Signon Security Exit

This control point enables the verification of the format and contents of the SIGNON command. The following requirements and restrictions apply:

- Implement the IBM Connect:Direct stage 1 signon exit as an executable load module.
- Name the load module DGACXSIG. To assemble, link-edit the appropriate name with an alias of DMCXSIGN by using DGAJCSG.
- Do not specify NCAL.
- Link-edit the module with either RENT or NORENT, depending on whether it is reentrant or not.
- Link-edit the module with either REUS or NOREUS, depending on whether it is reusable or not.
- Link-edit the module with an authorization code (AC) of 0 or 1. The AC has no effect on the exit, so you can use either.
- For the TSO IUI, the Stage 1 Signon Security exit module can come from a library in the LINKLIST, STEPLIB, TSOLIB, ISPLLIB, or LIBDEF-ISPLLIB concatenation. However, if there is a LIBDEF-ISPLLIB in effect, all Connect:Direct load modules (including the Stage 1 exits) must come from a library in the LINKLIST or the LIBDEF-ISPLLIB concatenation or a combination of the two.
- IBM Connect:Direct will only use the stage 1 signon security exit module if the module is deemed authorized. To be deemed authorized, the module must come from the LINKLIST or an APF-authorized library (partitioned data set). The entire concatenation does not need to be APF-authorized. Connect:Direct branches via BASSM to the Stage 1 exit, so the exit inherits the APF authorization the caller has. The IUI always executes APF unauthorized, so a Stage 1 exit running under the IUI always executes APF unauthorized.
- If a stage 1 signon security exit exists but is not deemed authorized, the SIGNON command fails.
- For DGADBATC and DGADCHLA, the module search is limited to the LINKLIST, JOBLIB, or STEPLIB. For more information, see [“IBM Connect:Direct Exits”](#) on page 315.
- Because the information that is passed to the exit by IBM Connect:Direct is located above the 16-megabyte line, and because IBM Connect:Direct branches via BASSM to the stage 1 exit, you must link-edit the module with AMODE ANY to allow it to access the information that is passed to it.
- The \$CD.SDGASAMP library contains a sample source module exit called DGACXSIG. Edit this module and modify the variable &SECTYPE to reflect the security system in use. Assemble and link-edit the exit. For IBM RACF or CA-TOP SECRET, use the character string RACF for &SECTYPE. For CA-ACF2, use the character string ACF2.

All IBM Connect:Direct nodes in cross-domain signon (or multi-session signon) with a IBM Connect:Direct node that uses the stage 1 signon exit must also use the stage 1 signon exit.

Signon Errors

If you are receiving signon errors about the stage 1 exit, allocate the special DDNAME APISECUR for special diagnostic output using one of the following methods

If you are using the DGADBATC, to route the trace output to SYSOUT, add the following DD statement to the DGADBATC JCL.

```
//APISECUR DD SYSOUT=*
```

If you are using the IUI, to route the trace output to the screen, issue the following TSO command.

```
TSO ALLOC F(APISECUR) DA(*)
```

To route the trace output to a data set, issue the following TSO command:

```
TSO ALLOC F(APISECUR) SHR DSN('data-set')
```

You must preallocate the data set with the following DCB attributes.

```
DSORG=PS  
RECFM=VBA  
LRECL=121,  
BLKSIZE=125 or greater
```

To route the output to spool, issue the following TSO command:

```
TSO ALLOC F(APISECUR) SYSOUT(*)
```

Sample SIGNON Panel

Since the TSO user ID has already been authenticated when logging on to TSO, you can bypass the IBM Connect:Direct IUI Signon panel. The Stage 1 Signon Security Exit and replacing the original Signon panel are both required.

The \$CD.SDGASAMP library contains a member called DGA@NPAN that you can use to replace the existing signon panel (DGA@GNON) after you implement the Stage 1 Signon Security exit. This sample panel in conjunction with the Stage 1 Signon Security exit allows you to bypass the Signon panel when starting the ISPF/IUI interface. Refer to \$CD.SDGASAMP member called DGAUPAN for a sample SMP/E USERMOD which contains instructions to implement this sample panel.

Stage 2 Security Exit

This control point applies to all environments and is implemented as a user-supplied exit. It provides a standard interface for user ID and password verification and for establishing IBM Connect:Direct functional authority and file access verification. Although you can use it for many different purposes, the stage 2 security exit is designed to provide the interface to your security system. You can also use it to invoke an exit to test new applications and customer connections. For more information, see [Process Exit for Testing \(DGAXPRCT\)](#).

The following requirements and restrictions apply:

- The stage 2 security exit is implemented as an executable load module.

- The name of the load module is user-defined, but it cannot conflict with any IBM Connect:Direct load modules.
- Specify the SECURITY.EXIT initialization parameter to activate the stage 2 security exit. This parameter also specifies whether the exit is used for ALL security checking or just DATASET access validation.
- You must link-edit the module as re-entrant and reusable and place it in a load library that the IBM Connect:Direct DTF can access. Do not specify NCAL. For more information, see [“IBM Connect:Direct Exits” on page 315](#).
- To prevent a remote node's security from using Signon dummy passwords, you can use the initialization parameter, REMOTE.DUMMY.PASSWORD. See [REMOTE.DUMMY.PASSWORD=\[YES | INTERNAL \]](#) for more details.
- Because information passed to the exit by IBM Connect:Direct is located above the 16-megabyte line, you must link-edit the module with AMODE 31 to make it capable of executing in 31-bit mode. Also, you must use RMODE 24 for PARM validation to work properly. You must link-edit the module as re-entrant and reusable and place it in a load library that the IBM Connect:Direct DTF can access. Do not specify NCAL. Use SDGAJCL member DGAJSAF to perform assembly and link. For more information, see [“IBM Connect:Direct Exits” on page 315](#).

Considerations for Systems with z/OS UNIX System Services

The following considerations apply to systems with z/OS UNIX System Services:

- Access to HFS files is controlled by UNIX System Services. The user ID under which the DTF runs must have UPDATE authority to the BPX.SERVER facility. In addition, the submitter ID/password, the PNODEID/password, or SNODEID/password must be valid. z/OS UNIX System Services enables or denies access based on the UNIX permission rules.
- IBM Connect:Direct can also check HFS access without requiring a password. To use a password length of zero, you must set up an IBM RACF profile BPX.SRV.userid in class SURROGAT and make sure that the Connect:Direct started task userid has READ access to this profile. For more information, see [Defining a Surrogate for User IDs with No Password](#).
- UNIX System Service (BPX) calls are executed in the IBM Connect:Direct IUI under the TSO or Batch User ID. BPX calls require that a user ID has an OMVS segment defined to it within the external security product, such as IBM RACF, ACF2 or CA-TOP SECRET.

Note: The BPX calls are used to resolve the TCP/IP name or address for reporting purpose in Select Statistics. For more information on adding an OMVS segment to a user ID, see [IBM Connect:Direct for z/OS Release Notes](#).

Sample Source Modules in the SDGASAMP Library

The \$CD.SDGASAMP library contains sample source modules for several release levels of z/OS security systems. These sample routines invoke a common macro called DGASECUR. This macro is the actual source code for the sample exits and is conditionally assembled based on the security system in use. Samples are provided for CA-ACF2, IBM RACF, and CA-TOP SECRET. You can accommodate other systems by using the sample code as a model.

DGAMGSAF is a sample exit routine for all security software supporting the RACROUTE interface. It uses the z/OS Security Access Facility (SAF). The TCB Extensions Feature (TCBSENV) must be present for correct operation.

DGASECUR Parameters

You can edit the parameters in the DGASECUR macro to select the appropriate parameters.

The parameters are described in each source module and summarized in the following table. The parameters related to functional authority levels are listed separately in [IBM Connect:Direct Functional Authority](#).

Parameter	Description
TYPE=[SAF]	Identifies the type of exit.

Parameter	Description
STAGE1=[YES, <u>NO</u>]	Identifies whether the stage 1 signon exit is implemented.
NOPASS=[YES, <u>NO</u>]	Specifies if data set validity calls to the security subsystem are to be made without using passwords.
SECSYS=[ACF,TSS,RACF]	Identifies the security system package installed on your z/OS system.
AUTHENTRY=[YES NO EXTERNAL]	<p>NO specifies to have the exit work like it always has or disabled this feature;</p> <p>YES specifies to require an AUTH file entry for all UIDs using a DUMMY-PWD</p> <p>EXTERNAL specifies to only require it from external PNODEs.</p> <p>Note: If an AUTH file entry is required but not defined, message RACF018* is issued (see AUTHMSG).</p>
AUTHXLAT=[YES NO]	<p>NO specifies that the AUTH file entry does not have to have a Security UID/PWD and if it doesn't, the logic works like it does today - only the submitter UID is checked by the security product.</p> <p>YES specifies that the AUTH file entry must have a valid Security UID/PWD. If a valid Security UID/PWD is required but not defined in the AUTH file entry, new message RACF019* is issued (see AUTHMSG).</p>
AUTHMSG=[WARN FAIL]	<p>FAIL indicates that if:</p> <ul style="list-style-type: none"> AUTHENTRY requirements are not satisfied the process will fail with RC=8 and MSGID RACF018E AUTHXLAT requirements are not satisfied the process will fail with RC=8 and MSGID RACF019E. <p>A value of WARN changes the RC to 0 and the last letter of the MSGID to 'W'.</p>
AUTHWTO=[YES NO]	Specifies that if any of the above new MSGIDs is set, the message ID only appears in the completion message, if present, and the PT STAT record. YES also causes the message to be filled in with the submitter node and user ID and issued via WTO, so that it appears in the JOB LOG and the WT STAT record.
NEWPASS=[YES, <u>NO</u>]	Specifies whether the security system password of a user can be changed at signon time.
PNODEID=[YES, <u>NO</u>]	Specifies if this exit enables security override of a PNODE user ID if one is used in a Process statement.
SNODEID=[YES, <u>NO</u>]	Specifies if this exit enables an incoming node to use a SNODEID.

Parameter	Description
RESTRICT=[YES, <u>NO</u>]	<p>(ACF2 only) Indicates if a user can specify a restricted ID (an ID with no ACF2 password) to access IBM Connect:Direct.</p> <p>When running a stage 1 signon exit, this parameter has no meaning. The stage 1 exit inserts a dummy password into the user security record. In an environment with a stage 1 exit, all data set validity calls to the security subsystem are made with a NOPASS option. Therefore, the security subsystem does not differentiate between a restricted ID and an ID with a valid password.</p>
PROTECTD=[<u>NO</u> ,YES, AUTH]	<p>(IBM RACF only) Indicates if a user can specify a protected ID (an ID with no IBM RACF password) to access IBM Connect:Direct. The User IDs are defined to IBM RACF with the NOPASSWORD and NOIDCARD parameters. A protected ID cannot be used in situations requiring a password.</p> <p>Specifying SECSYS=RACF, PROTECTD=YES, allows users to access Connect:Direct using User IDs without passwords when the Connect:Direct DTF is started with a protected User ID. Therefore, Processes may specify PNODEID= SNODEID without specifying a password..</p> <p>Specifying SECSYS=RACF, PROTECTD=AUTH, works like specifying PROTECTD=YES, except that the RACF protected ID must be specified as the SEC ID in an AUTH file entry for the USERID/NODE being verified. Also, the initialization parameter INVOKE.SPOE.ON.SNODEID must be YES</p> <p>Note: For PNODEID protected id processing, specifying PROTECTD=AUTH is the same as specifying PROTECTD=YES and no entry needs to be in the AUTH file as no SPOE processing is performed for PNODEID overrides.</p> <p>Specifying SECSYS=RACF, PROTECTD=NO generates no support for IBM RACF Protected User IDs.</p>
NPFAIL=[YES, <u>NO</u>]	<p>Specifies whether to refuse a request to copy a file that is not protected. This parameter is only valid for IBM RACF and CA-TOP SECRET users.</p>
TRACE=[NO,DEBUG]	<p>Specifies whether tracing will be turned on for this security exit.</p> <p>TRACE=DEBUG turns control of tracing over to the DEBUG bit settings and/or the existence of the SECURITY DD statement. You can direct the SECURITY DD to SYSOUT or a disk file with attributes of RECFM=VBA, LRECL=121, and BLKSIZE=125 or greater.</p> <p>For more information about starting a security trace, see Security Traces.</p>
CICSID=name	<p>Specifies the dummy USERID name for establishing the initial session between the CICS Interface and a IBM Connect:Direct DTF. Use this parameter for security when using the CICS interface.</p>

Parameter	Description
PASSTK=[YES NO]	Results in the generation of a routine in the Stage 2 exit. This routine generates an IBM RACF PassTicket for PNODE processing and receives a PassTicket for SNODE processing. If PASSTK=NO is coded, no IBM RACF PassTicket processing is done. For more information, see Generating IBM RACF PassTickets .
PROCEXIT=[DGAXPRCT,NO]	Specifies if the DGAXPRCT exit (Process Exit for testing) is to be used or not. To invoke the exit, specify DGAXPRCT. To prevent the DGAXPRCT exit from being invoked, specify NO. For more information, see Setting Up and Using the DGAXPRCT Exit .
UID=local ID	Specifies the local identifier which appears in NDMLOG output along with the PTF maintenance listing.
CLASS=DATASET FACILITY	Provides ability to use the IBM RACF DATASET or FACILITY class to define user authorization profiles within IBM Connect:Direct. For more information, see IBM Connect:Direct Functional Authority .

Generating IBM RACF PassTickets

An IBM RACF PassTicket is a temporary one-time password that is good for only a short period of time. The generation of the PassTicket requires a Userid and an Application Profile Name. To validate the PassTicket, the same Userid and same Application Profile Name must be used. The Application Profile Name must be defined to IBM RACF as the name of a PTKTDATA profile. IBM Connect:Direct allows the specification of a PassTicket Application ID in the AUTH file.

To identify a node as capable of generating PassTickets, the third parameter in the SECURITY.EXIT initialization parm must specify PSTKT as shown in the following example:

```
SECURITY.EXIT=(module name,DATASET|ALL,PSTKT)
```

If a session is established with another Connect:Direct for z/OS that also supports PassTicket generation, a PassTicket is generated under the following conditions:

- The PNODE is PassTicket capable.
- The SNODE is PassTicket capable.
- SNODEID= is specified without a password.
- The AUTH file contains an entry for this SNODEID/SNODE and PassTicket information is defined. The Application Profile Name is passed to the Stage 2 security exit to generate the PassTicket.
- The PassTicket is generated using the Application Profile Name and the SNODEID userid.

A generated PassTicket is passed to the SNODE as the Security Password for the SNODEID, along with an indication that a PassTicket is being used. When the SNODE receives a session start with an indication that a PassTicket is being used, it attempts to retrieve the Application Profile Name by looking in the AUTH file for an entry for the SubmitterID/PNODE with the PassTicket information defined. The Application Profile Name and SNODEID userid are used to validate the PassTicket.

PassTickets can also be used to access HFS files.

Return Codes

The following table describes the valid return codes from the stage 2 exit for signon, Process start, or security delete.

RC	Description
0	No error
8	Insufficient access authority; an SAFB008I is issued
20	Security system inactive (ACF only); an SAFB020I is issued

If none of the return codes in the previous table are returned, IBM Connect:Direct issues the message SAFB003I.

Note: If SQMSGYES is on, IBM Connect:Direct does not overlay the message ID set by the exit, and the Process ends with the message set by the exit.

The valid return codes for the data set create security call are:

RC	Description
0	No error
8	Insufficient access authority; an SVSA908I ABEND is issued
12	Invalid data in SQCB; a U2250 ABEND is issued
16	No storage available for GETMAIN; a U2251 ABEND is issued
20	Security system inactive; IBM Connect:Direct performs a STOP IMMEDIATE
24	ADJ node not allowed to send (RACF100I) or receive (RACF101I) and the node executing the exit is PNODE
28	ADJ node not allowed to send (RACF100I) or receive (RACF101I) and the node executing the exit is SNODE

After control is returned from the exit to the DTF, the return code is set to 8 if the exit was run from PNODE and to 12 if the exit was run from SNODE.

If none of the return codes in the previous table are returned, IBM Connect:Direct ends abnormally with a U2252 ABEND.

IBM Connect:Direct Functional Authority

When you sign on to a IBM Connect:Direct running with security (or when a Process you submit begins executing), you are assigned a 20-byte authorization bit mask (ABM) based on a recommendation by the stage 2 security exit or the IBM Connect:Direct Authorization file. The ABM describes your unique functional authority within IBM Connect:Direct .

IBM Connect:Direct provides four standard security levels in the DGAMGSAF exit described in the following table. The ADMVOL, OPRVOL, DBAVOL, and GENVOL parameters indicate the volumes on which these data sets reside. If you do not specify volume names in the DGAMGSAF stage 2 security exit, the exit provides default volume names for monitoring by your security subsystem.

Note:

If the CLASS=DATASET is defaulted or specified, IBM Connect:Direct uses the four standard security levels described above. If the CLASS=FACILITY is specified, IBM Connect:Direct uses these same four standard security levels as defined in the following table. However, the volser information is not needed or used by the STAGE2 security exit (as provided with the installation media).

Note: To add new functional authority levels or change the privileges in the standard functional authority levels, see [Functional Authority Privileges](#). There are also 10 user-defined functional levels (US0DSN-US9DSN). For more information about these functional levels, see [“Example 3 - Defining Additional Levels of Functional Authority” on page 202](#) and [“Example 4 - Assigning Read-Only Authority to a User Authorization Level” on page 203](#).

Parameter	Description
ADMDSN=file name ADMVOL=volser	Specifies full administrator authority. The specified user is authorized to execute all Process language statements and commands.
DBADSN=file name DBAVOL=volser	Specifies DB2 Data Base Administrator.
OPRDSN=file name OPRVOL=volser	Specifies operator authority. The specified user is authorized to delete, change, display, flush, and submit Processes; stop IBM Connect:Direct; start and stop traces; and display, add, delete, and update type.
GENDSN=NULLFILE filename GENVOL=volser	Specifies general authority. The specified user is authorized to delete, change, display, and flush his own Processes, submit Processes, and display, add, delete, and update Type. If NULLFILE is coded, a user who logs on to IBM Connect:Direct without specific administrator or operator authorization is, by default, classified as a general user. The following is a sample User Authorization screen, showing commands available to a general user.

If a bit in one of these standard ABMs is set to one, you are authorized to perform the IBM Connect:Direct command that is associated with that bit, according to the security levels.

For example, if you have the authority to read the ADMDSN, you are given the administrator bit mask that enables you to perform administrator functions. If you do not have ADMDSN authority, OPRDSN read authority is checked, and so on, according to the sequence described in [Functional Authority Validation Sequence](#).

To assign IBM Connect:Direct functional authority, define four data sets or resources on your system to correspond to the administrator, operator, database administrator, and general user data sets.

You can specify IBM Connect:Direct functional authority to individual users by verifying access to one of the named resources. These resource names refer to IBM Connect:Direct functional authority grouped by the four categories. IBM Connect:Direct users are given access to the resource that corresponds to their level of authority.

In addition, you can modify the standard ABMs provided by IBM Connect:Direct to change the default privileges for a functional authority level. See [Functional Authority Privileges](#). In addition, you can expand the number of functional authority levels by creating authorization bit masks for new user-defined levels. See [Defining Additional Levels of Functional Authority](#).

Example Functional Authority Profiles

In the sample screens below, YES next to a command means that the security level is authorized to execute the command, NO means that the security level is not authorized to execute the command, and SUB means that the security level is authorized to execute the command only if the Process was submitted by the particular user.

The following example shows the User Authorization screen for the administration authority (ADMDSN=file name, ADMVOL=volser).

Note: You can access the User Authorization screen through the IBM Connect:Direct Primary Options Menu. This menu and its options are discussed in the chapter on the Interactive User Interface in *IBM Connect:Direct for z/OS User Guide*. The User Authorization screen lists all the commands a particular user is authorized and not authorized to execute.

```

node.name                                USER AUTHORIZATION                                13:30
CMD ==>

      AUTH  COMMAND                                AUTH  COMMAND
      -----                                -----
1) YES - CHANGE PROCESS                      15) YES - SELECT TASK
2) YES - DELETE PROCESS                     16) YES - SELECT TYPE
3) YES - DELETE TYPE                       17) YES - SELECT USER
4) YES - DELETE USER                      18) YES - SUBMIT PROCESS
5) YES - FLUSH PROCESS                     19) YES - SUBMIT WITHIN PROC
6) YES - FLUSH TASK                       20) YES - SUSPEND PROCESS
7) Y/Y - INSERT/UPDATE TYPE                21) YES - STAT COMMAND
8) Y/Y - INSERT/UPDATE USER                22) YES - EVENT COMMAND
9) YES - MODIFY (TRACE)                    23) YES - VIEW PROCESS
10) YES - STOP Connect:Direct               24) YES - PERFORM CRC OVERRIDES
11) YES - UPDATE NETWORK MAP                25) NO - CONFIRM DELETE
12) YES - SELECT NETWORK MAP                26) NO - CONFIRM DEL OFF
13) YES - SELECT PROCESS                    27) Y/Y - SECURE+ ADMIN/CMDS
14) YES - SELECT STATISTICS                 28) NO - UPDATE INITPARM

```

The following example shows the User Authorization screen for the DB2 data base administrator authority (DBADSN=file name, DBAVOL=volser).

```

node.name                                USER AUTHORIZATION                                hh:mm
CMD ==>

      AUTH  COMMAND                                AUTH  COMMAND
      -----                                -----
1) SUB - CHANGE PROCESS                      15) SUB - SELECT TASK
2) SUB - DELETE PROCESS                     16) YES - SELECT TYPE
3) YES - DELETE TYPE                       17) NO - SELECT USER
4) NO - DELETE USER                      18) YES - SUBMIT PROCESS
5) SUB - FLUSH PROCESS                     19) YES - SUBMIT WITHIN PROC
6) SUB - FLUSH TASK                       20) SUB - SUSPEND PROCESS
7) Y/Y - INSERT/UPDATE TYPE                21) NO - STAT COMMAND
8) N/N - INSERT/UPDATE USER                22) NO - EVENT COMMAND
9) NO - MODIFY (TRACE)                     23) SUB - VIEW PROCESS
10) NO - STOP Connect:Direct                24) YES - PERFORM CRC OVERRIDES
11) NO - UPDATE NETWORK MAP                 25) NO - CONFIRM DELETE
12) NO - SELECT NETWORK MAP                 26) NO - CONFIRM DEL OFF
13) SUB - SELECT PROCESS                    27) N/N - SECURE+ ADMIN/CMDS
14) YES - SELECT STATISTICS                 28) NO - UPDATE INITPARM

```

The following example shows the User Authorization screen for the operator authority (OPRDSN=file name, OPRVOL=volser).

```

node.name                                USER AUTHORIZATION                                hh:mm
CMD ==>

      AUTH  COMMAND                                AUTH  COMMAND
      -----                                -----
1) YES - CHANGE PROCESS                      15) YES - SELECT TASK
2) YES - DELETE PROCESS                     16) YES - SELECT TYPE
3) YES - DELETE TYPE                       17) NO - SELECT USER
4) NO - DELETE USER                      18) YES - SUBMIT PROCESS
5) YES - FLUSH PROCESS                     19) YES - SUBMIT WITHIN PROC
6) YES - FLUSH TASK                       20) YES - SUSPEND PROCESS
7) Y/Y - INSERT/UPDATE TYPE                21) NO - STAT COMMAND
8) N/N - INSERT/UPDATE USER                22) NO - EVENT COMMAND
9) YES - MODIFY (TRACE)                    23) YES - VIEW PROCESS
10) YES - STOP Connect:Direct               24) YES - PERFORM CRC OVERRIDES
11) NO - UPDATE NETWORK MAP                 25) NO - CONFIRM DELETE
12) NO - SELECT NETWORK MAP                 26) NO - CONFIRM DEL OFF
13) YES - SELECT PROCESS                    27) N/N - SECURE+ ADMIN/CMDS
14) YES - SELECT STATISTICS                 28) NO - UPDATE INITPARM

```

The following example shows the User Authorization screen for the general user authority (GENDSN=NULLFILE|filename, GENVOL=volser).

node.name	USER AUTHORIZATION				hh:mm
CMD ==>					
	AUTH	COMMAND		AUTH	COMMAND
	-----			-----	
	1)	SUB - CHANGE PROCESS		15)	SUB - SELECT TASK
	2)	SUB - DELETE PROCESS		16)	YES - SELECT TYPE
	3)	YES - DELETE TYPE		17)	NO - SELECT USER
	4)	NO - DELETE USER		18)	YES - SUBMIT PROCESS
	5)	SUB - FLUSH PROCESS		19)	YES - SUBMIT WITHIN PROC
	6)	SUB - FLUSH TASK		20)	SUB - SUSPEND PROCESS
	7)	Y/Y - INSERT/UPDATE TYPE		21)	NO - STAT COMMAND
	8)	N/N - INSERT/UPDATE USER		22)	NO - EVENT COMMAND
	9)	NO - MODIFY (TRACE)		23)	SUB - VIEW PROCESS
	10)	NO - STOP Connect:Direct		24)	YES - PERFORM CRC OVERRIDES
	11)	NO - UPDATE NETWORK MAP		25)	NO - CONFIRM DELETE
	12)	NO - SELECT NETWORK MAP		26)	NO - CONFIRM DEL OFF
	13)	SUB - SELECT PROCESS		27)	N/N - SECURE+ ADMIN/CMDS
	14)	YES - SELECT STATISTICS		28)	NO - UPDATE INITPARM

Functional Authority Validation Sequence

The security-checking sequence follows:

1. When the stage 2 security exit is called to determine IBM Connect:Direct functional authority for a user (at signon or Process start), it first checks with the security subsystem (that is, CA-ACF2, IBM RACF, or CA-TOP SECRET) to determine if the user can read the Administrator data set. If so, the authority of the user is set as an Administrator.
2. If the user is not allowed to read the Administrator data set, the exit checks to see if the user can read the Operator data set. If yes, the user is given Operator authority.
3. If the user is not allowed to read the Operator data set, the exit checks to see if the user can read the Data Base Administrator data set. If so, the user is given Data Base Administrator authority.
4. If the user is not allowed to read the Data Base Administrator data set, and the stage 2 exit includes GENDSN=NULLFILE, the user is given General User authority. If you specify a data set name for GENDSN, the exit either assigns the user General User authority if the user can read the data set, or disables the IBM Connect:Direct function requested (signon or Process execution) if the user cannot read the data set.

Functional Authority Privileges

The privileges set for each of the four standard IBM Connect:Direct functional authority levels are the default privileges provided in the base product. This section describes how to change the privileges in the standard functional authority levels or add new functional authority levels.

You can modify the IBM Connect:Direct stage 2 security exit macro, DGAXAUTH, to change the functions a user can perform in a particular authorization level. The \$CD.SDGAMAC library contains a macro called DGA\$MFLG that describes each of the 20 bytes of functional authorization.

The following is the generic 20 byte mask that is mapped by a dummy section (DSECT) in the DGA\$MFLG macro along with a definition of each byte, the general function that the bits represent, and the specific settings:

Byte	Function	Setting
BYTE00	Reserved for future use	
BYTE01	Display, Add, Update, and Delete User Commands	ADDUSR–Add user UPDUSR–Update user DELUSR– Delete user DSPUSR–Display user
BYTE02	Reserved for future use	

Byte	Function	Setting
BYTE03	Reserved for future use	
BYTE04	Display, Add, Update, and Delete Network Map Commands	ADDNET–Add network map UPDNET–Update network map DELNET–Delete network map DSPNET–Display network map
BYTE05	Change and Delete Process Commands	CHGPRC–Change Process DELPRC–Delete Process
BYTE06	Display Process, Statistics, and Traces, Flush Process, and Use Stats commands	DSPPRC–Display Process DSPSTA–Display Statistics DSPTRC–Display Trace FLSPRC–Flush Process STATCMD–Use Statistics Commands
BYTE07	Start/Stop IBM Connect:Direct, Start/Stop Traces, Modify Init parms, Suspend/Resume Sessions, Use Event Services, Update APKey commands, and Update initialization parameters in Control Center	STPNDM–Start/Stop IBM Connect:Direct SSTRAC–Start/Stop Traces and Modify Initparms EVENTCMD–Use Event Services Commands REFSH–Update init parms UPDKEY–Update license key
BYTE08	Perform Connect:Direct Secure Plus Parm file and netmap administration functions in Control Center	S#RNCR–Read Connect:Direct Secure Plus parm file S#WNCR–Write to Connect:Direct Secure Plus parm file DSPNCR–Display netmap control functions in Control Center UPDNCR–Allow netmap update functions in Control Center
BYTE09	Display, Add, Update, and Delete Type commands	ADDTYP–Add type UPDTYP–Update type DELTYP–Delete type DSPTYP–Display type
BYTE10	Use COPY, RUN JOB, MODALS, and SUBMIT Statements, and View Process and CRC Override commands	GCOPY–Use COPY statement GRUNJ–Use RUN JOB statement GMODALS–Use MODAL statement GSUBMIT–Use SUBMIT statement VIEWPR–View Process GOVCRC–Perform CRC overrides

Byte	Function	Setting
BYTE11	Use Submit within a Process and RUN TASK statements, display Confirm Delete prompt, and turn Confirm Delete prompt off for a session. Note: The Confirm Delete function also includes the Flush and Suspend commands, that is, the user is prompted to confirm before the Flush and Suspend Commands in addition to the Delete command.	GSUB–Use Submit within a Process statement GRUNT–Use RUN TASK statement GCDEL–Display Confirm Delete, Flush, and Suspend prompts GCDELOFF–Turn off Confirm delete prompt off for session
BYTE12	General User Functions– Select, Delete, Flush, Change, and View Process, and Display Statistics, and Display Plex environment (the last command for an Administrator only). Note: The General User functions enable you to restrict applying each command to Processes associated with a submitter ID.	GDSPPRC–Display Process GDELPRC–Delete Process GDFLSPRC–Flush Process GDSPSTA–Display Statistics GCHGPRC–Change Process GVIEWPR–View Process DSPPLX–Display Plex Environment
BYTE13	Reserved for future use	
BYTE14	Reserved for future use	
BYTE15	Reserved for future use	
BYTE16	Reserved for future use	
BYTE17	Reserved for future use	
BYTE18	Reserved for future use	
BYTE19	Reserved for future use	

The sample exit macro DGAXAUTH contains authorization bit masks for the four standard IBM Connect:Direct authority groups. The default settings shown in the following ABMs are in the DGAXAUTH macro in the \$CD.SDGAMAC library. The DGA\$MFLG bit mask contains all possible functions for each byte whereas the bit masks for a particular IBM Connect:Direct authority group may contain only a subset of the available functions. For example, BYTE 10 (DBA10) in the DB2 data base authority level authorization bit mask (DBAABM) below does not contain the View Process function (VIEWPR) while BYTE 10 in the ABM for the Operator authority level does. (Bytes reserved for future use are not shown.)

DGAXAUTH Authorization Bit Mask Examples

The following example shows the authorization bit mask for the Administrator authority level (ADMABM).

ABYTE1	DC	AL1 (ADDUSR+UPDUSR+DELUSR+DSPUSR)
ABYTE4	DC	AL1 (ADDNET+UPDNET+DELNET+DSPNET)
ABYTE5	DC	AL1 (CHGPRC+DELPRC)
ABYTE6	DC	AL1 (DSPPRC+DSPSTA+FLSPRC+STATCMD)
ABYTE7	DC	AL1 (STPNDM+SSTRAC+EVENTCMD+UPDKEY)
ABYTE8	DC	AL1 (UPDNCR+DSPNCR+S#WNCR+S#RNCR)
ABYTE9	DC	AL1 (ADDTYP+UPDTYP+DELTYP+DSPTYP)
ABYTE10	DC	AL1 (GCOPY+GRUNJ+GMODALS+GSUBMIT+VIEWPR+GOVCRC)
ABYTE11	DC	AL1 (GSUB+GRUNT)
ABYTE12	DC	AL1 (DSPPLX)

The following example shows the authorization bit mask for the Operator authority level (OPERABM).

```

OPER1   DC   XL1(00)      NULL - Not Set
OPER4   DC   XL1(00)      NULL - Not Set
OPER5   DC   AL1(CHGPRC+DELPRC)  DELETE/CHANGE PROCESS
OPER6   DC   AL1(DSPPRC+DSPSTA+FLSPRC)  DISPLAY/FLUSH PROCESS
*
OPER7   DC   AL1(STPNDM+SSTRAC)  STOP START-STOP TRACE
OPER9   DC   AL1(ADDTYP+UPDTYP+DELTYP
+DSPTYP)
*
OPER10  DC   AL1(GCOPY+GRUNJ+GMODALS+GSUBMIT+VIEWPR+GOVCRC)
*
OPER11  DC   AL1(GSUB+GRUNT)  REMOTE SUBMIT/RUN TASK

```

The following example shows the authorization bit mask for the DB2 data base authority level (DBAABM).

```

DBA1    DC   XL1(00)      NULL - Not Set
DBA4    DC   XL1(00)      NULL - Not Set
DBA9    DC   AL1(ADDTYP+UPDTYP+DELTYP+DSPTYP)
*
DBA10   DC   AL1(GCOPY+GRUNJ+GMODALS+GSUBMIT+GOVCRC)
*
DBA11   DC   AL1(GSUB+GRUNT)  COPY/RUN JOB/MODALS/SUBMIT
REMOTE SUBMIT/RUN TASK

```

The following example shows the authorization bit mask for the General User authority level (GUSRABM).

```

GUSR1   DC   XL1(00)      NULL - Not Set
GUSR4   DC   XL1(00)      NULL - Not Set
GUSR9   DC   AL1(ADDTYP+UPDTYP+DELTYP+DSPTYP)
*
GUSR10  DC   AL1(GCOPY+GRUNJ+GMODALS+GSUBMIT+GOVCRC)
*
GUSR11  DC   AL1(GSUB+GRUNT)  REMOTE SUBMIT/RUN TASK
GUSR12  DC   AL1(GDSPPRC+GDELPRC+GFLSPRC+GDSPSTA+GCHGPRC+GVIEWPR)
*
*
*

```

To change the bits in any given authorization byte, locate the bit labels in the DGA\$MFLG macro and update the DGAXAUTH macro. To implement any changes made and put your new exit into effect, you must stop and restart IBM Connect:Direct.

Example 1 - Broadening Privileges for General Users

To authorize general users to perform a SELECT PROCESS command and a SELECT STATISTICS command for Processes submitted with any user ID rather than just with the ID of the submitter, perform the following steps:

Procedure

1. Look in the DGA\$MFLG macro for the bits that allow the user to perform these two commands. These bits are located in BYTE06 of DGA\$MFLG.
2. Find the bits that allow these two commands only for the ID of the submitter. These bits are located in BYTE12 of DGA\$MFLG.
3. Locate the label in the DGAXAUTH macro that indicates the General User authorization bit mask (GUSRABM). General user BYTES 06 and 12 are currently set to the following values.

```

GUSR6   DC   XL1'00'
GUSR12  DC   AL1(GDSPPRC+GDELPRC+GFLSPRC+GDSPSTA+GCHGPRC+GVIEWPR)

```

4. To allow general users to perform SELECT PROCESS and SELECT STATISTICS commands for any user ID, remove GDSPPRC and GDSPSTA from BYTE12 and copy the DSPPRC and DSPSTA bits from BYTE06

in the DGA\$MFLG authorization bit mask and put them in BYTE06 in the GUSRABM, changing these bytes to the following values:

```
GUSR6    DC AL1(DSPPRC+DSPSTA)
GUSR12   DC AL1(GDELP RC+GFLSP RC+GCHGPRC+GVIEWPR)
```

5. Reassemble and link-edit your security module that uses the DGAXAUTH macro.
6. To put the new exit into effect, stop and restart IBM Connect:Direct.

Example 2 - Forcing the Confirm Prompt for General Users

If a user has the authority to delete, flush or suspend a Process, the default setting allows the user to perform the action automatically. As soon as the user enters the command, it is executed instantly. However, you can modify this default privilege and require a user to confirm the action before it is executed. In addition, you can specify whether a user can turn off the Confirm Delete/Flush/Suspend Command prompt for a particular session after the prompt displays at least one time.

About this task

The following sample procedure shows you how to turn on the Confirm Delete/Flush/Suspend Command prompt for users in the general user authority category but at the same time allow them to turn off the prompt for a particular session.

Procedure

1. Locate BYTE11 in the DGA\$MFLG macro. Two of the four bits, GCDEL and GCDELOFF, turn on the Confirm Delete/Flush/Suspend Command prompt and if turned on, permit a user to turn off the Confirm Delete/Flush/Suspend Command prompt temporarily for the current session. (The other two bits pertain to the Submit within a Process and RUN TASK commands.)
2. Locate the label in the DGAXAUTH macro that indicates the General User authorization bit mask setting (GUSRABM). General user BYTE 11 is currently set to the following values.

```
GUSR11 DC AL1 (GSUB+GRUNT)
```

3. To ensure that the Confirm/Delete/Suspend Command prompt displays for all users in the general user category, add GCDEL to change BYTE11 as follows:

```
GUSR11 DC AL1 (GSUB+GRUNT+GCDEL)
```

4. To let users in the general user category turn off the Confirm/Delete/Suspend Command prompt for a particular session, add GCDELOFF to change BYTE11 as follows:

```
GUSR11 DC AL1 (GSUB+GRUNT+GCDEL+GCDELOFF)
```

5. Reassemble and link-edit your security module that uses the DGAXAUTH macro.
6. To put the new exit into effect, stop and restart IBM Connect:Direct.

Example 3 - Defining Additional Levels of Functional Authority

IBM Connect:Direct provides additional authorization bit masks (US0DSN through US9DSN) that you can use to expand the number of functional authority levels beyond the standard four levels.

About this task

The following example shows the authorization bit mask for the user-definable ABM.

U0BYTE1	DC	XL1(00)	NULL - Not Set
U0BYTE4	DC	XL1(00)	NULL - Not Set
U0BYTE5	DC	AL1(CHGPRC+DELPRC)	DELETE/CHANGE PROCESS
U0BYTE6	DC	AL1(DSPPRC+DSPSTA+FLSPRC)	DISPLAY/FLUSH PROCESS
*			DISPLAY STATISTICS
U0BYTE7	DC	AL1(STPNDM+SSTRAC)	STOP START-STOP TRACE
U0BYTE8	DC	XL1'00'	NOT USED
U0BYTE9	DC	AL1(ADDTYP+UPDTYP+DELTYP+DSPTYP)	DISPLAY/ADD/DELETE TYPE
*			COPY/RUN JOB/MODALS/SUBMIT
U0BYTE10	DC	AL1(GCOPY+GRUNJ+GMODALS+GSUBMIT+VIEWPR+GOVCRC)	COPY/RUN JOB/MODALS/SUBMIT
*			REMOTE SUBMIT/RUN TASK
U0BYTE11	DC	AL1(GSUB+GRUNT)	REMOTE SUBMIT/RUN TASK

Procedure

1. To create your own authorization level, locate the user-definable ABM you want to use, and change the bytes.
2. Add the data set and volume names to the DGAMGSAF stage 2 security exit.
3. Assemble and link-edit the DGAMGSAF exit.
4. To implement the new authorization level, into effect, you must stop and restart IBM Connect:Direct.

Example 4 - Assigning Read-Only Authority to a User Authorization Level

To define a new security profile to allow read-only authority for users, follow this procedure. After you implement it, when a user signs on to IBM Connect:Direct, they are assigned an authorization bit mask that allows them to display and view Processes, and display statistics but they cannot submit or run a Process.

Procedure

1. Modify the DGASECUR macro by locating the USR0ABM label and making the following changes:
 - a) Delete both the CHGPRC and DELPRC bits in BYTE05.
 - b) Delete the FLSPRC bit in BYTE06.
 - c) Delete both the STPNDM and SSTRAC bits in BYTE07.
 - d) Delete all bits in BYTE09.
 - e) Delete all bits in BYTE10 except for VIEWPR.
 - f) Delete all bits in BYTE11.

The USR0ABM should look like the following:

USR0ABM	DS	0XL20	DEFINES User Group Zero ABM Flags
U0BYTE0	DC	XL1'00'	NOT USED
U0BYTE1	DC	XL1'00'	NOT USED
U0BYTE2	DC	XL1'00'	NOT USED
U0BYTE3	DC	XL1'00'	NOT USED
U0BYTE4	DC	XL1'00'	NOT USED
U0BYTE5	DC	XL1'00'	DELETE/CHANGE PROCESS
U0BYTE6	DC	AL1(DSPPRC+DSPSTA)	Display Process
*			DISPLAY STATISTICS
U0BYTE7	DC	XL1'00'	STOP START-STOP TRACE
U0BYTE8	DC	XL1'00'	NOT USED
U0BYTE9	DC	XL1'00'	DISPLAY/ADD/DELETE TYPE
U0BYTE10	DC	AL1(VIEWPR)	View Process only
U0BYTE11	DC	XL1'00'	REMOTE SUBMIT/RUN TASK
U0BYTE12	DC	XL1'00'	NOT USED

2. Modify the DGAMGSAF example in the \$CD.SDGASAMP library to assign a file name to the new US0DSN parameter and indicate which volume it resides on.

```

DGAMGSAF    DGASECUR TYPE=SAF,                                X
              .                                              X
              .                                              X
              ADMDSN=$CD.ADMIN,                             X
              ADMVOL=VOLSER,                                 X
              OPRDSN=$CD.OPER,                               X
              OPRVOL=VOLSER,                                 X
              DBADSN=$CD.DBA,                                X
              DBAVOL=VOLSER,                                 X
              GENDSN=$CD.GUSER,                              X
              GENVOL=VOLSER,                                 X
              US0DSN=$CD.NEW.USER.LEVEL,                     X
              US0VOL=VOLSER

```

3. Assemble and link-edit the DGAMGSAF module using the sample JCL in \$CD.SDGAJCL(DGAJSAF).

```

//ASM EXEC PGM=ASMA90,
// PARM='OBJECT,NODECK,XREF(SHORT),RENT,USING(WARN(0),NOMAPX
// ),FLAG(NOCONT),SYSPARM(GEN),NOTEST'
//SYSIN DD DISP=SHR,DSN=connect.direct.SDGASAMP(DGA*****)
//SYSLIB DD DISP=SHR,DSN=connect.direct.samplib
// DD DISP=SHR,DSN=SYS1.MACLIB
// DD DISP=SHR,DSN=SYS1.AMODGEN
// DD DISP=SHR,DSN=SYS1.AMACLIB
// DD DISP=SHR,DSN=security.maclib
//SYSLIN DD DISP=(,PASS),DSN=&&OBJ,
// UNIT=SYSDA,SPACE=(CYL,(1,1)),
// DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//*****
//* LKED *
//*****
//LKED EXEC PGM=IEWL,COND=(0,LT,ASM),
// PARM=('SIZE=(256K,13K),LIST,LET,XREF,RENT',
// 'REUS')
//SYSLIB DD DISP=SHR,DSN=connect.direct.SDGALINK
// DD DISP=SHR,DSN=security.loadlib
//SYSLIN DD DISP=(OLD,DELETE),DSN=&&OBJ
//SYSLMOD DD DISP=SHR,DSN=connect.direct.SDGALINK(DGA*****)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1),,CONTIG)

```

4. If necessary, update the IBM Connect:Direct initialization parameter, SECURITY.EXIT, to specify the new exit.

```
SECURITY.EXIT = (mod-name,ALL)
```

5. Initialize IBM Connect:Direct.

Example 5 - Defining a New Administrator Level

To define a new security profile giving the administrator all normal administrator functions except the ability to run Processes, follow this procedure. Authorization BYTES 10 and 11 represent the functions that are to be disallowed. If you have used USR0DSN to define another level, use USR1DSN for this new profile.

Procedure

1. Modify the DGASECUR macro by locating the USR1ABM label and updating the various bytes as follows:

USR1ABM	DC	0XL20
U1BYTE0	DC	XL1'00' NOT USED
U1BYTE1	DC	AL1(ADDUSR+UPDUSR+DELUSR+DSPUSR)
U1BYTE2	DC	XL1'00' NOT USED
U1BYTE3	DC	XL1'00' NOT USED
U1BYTE4	DC	AL1(ADDNET+UPDNET+DELNET+DSPNET)
U1BYTE5	DC	AL1(CHGPRC+DELPRC)
U1BYTE6	DC	AL1(DSPPRC+DSPSTA+FLSPRC+STATCMD)
U1BYTE7	DC	AL1(STPNDM+SSTRAC+EVENTCMD+UPDKEY)
U1BYTE8	DC	AL1(UPDNCR+DSPNCR)
U1BYTE9	DC	AL1(ADDTYP+UPDTYP+DELTYP+DSPTYP)
U1BYTE10	DC	AL1(VIEWPR)
U1BYTE11	DC	XL1'00'
U1BYTE12	DC	AL1(DSPPLX)
U1BYTE13	DC	XL1'00'
U1BYTE14	DC	XL1'00'

2. Modify the DGAMGSAF example in THE \$CD.SDGASAMP library to define the new US1DSN parameter with the new security profile as follows:

DGAMGSAF	DGASECUR TYPE=SAF,	X
	.	X
	ADMSDN=\$CD.ADMIN,	X
	ADMVOL=VOLSER,	X
	OPRDSN=\$CD.OPER,	X
	OPRVOL=VOLSER,	X
	DBADSN=\$CD.DBA,	X
	DBAVOL=VOLSER,	X
	GENDSN=\$CD.GUSER,	X
	GENVOL=VOLSER,	X
	US0DSN=\$CD.NEW.USER.LEVEL,	X
	US0VOL=VOLSER,	X
	US1DSN=\$CD.NEW.ADMIN,	X
	US1VOL=VOLSER	

3. Assemble and link-edit the DGAMGSAF module using the sample JCL in \$CD.SDGAJCL(DGAJSAF).
4. If necessary, update the initialization parameter, SECURITY.EXIT, to specify the new exit.
5. Initialize IBM Connect:Direct in the normal manner.

Example 6 - Defining a Surrogate for User IDs with No Password

Use the BPX.SERVER profile to set the scope of z/OS resources that the server can access when acting as a surrogate for its clients. BPX.SERVER UPDATE access lets the server establish a thread level (task-level) security environment for clients connecting to the server. When the IBM RACF identity of the application server is granted UPDATE authority to BPX.SERVER in the IBM RACF FACILITY class, the server can act as a surrogate for the client.

About this task

This procedure contains sample IBM RACF commands. For more information, refer to IBM RACF manuals. For more information about how to define SURROGAT in other external security products, such as ACF2 or CA-TOP SECRET, refer to the manuals of the specific vendor.

Procedure

1. Make sure that the Stage 2 Security exit can verify if Stage 1 has set the dummy password in SQCB. The DGASECUR macro contains label STG1NPW which includes the following instruction:

```
0I SQFLAG,SQDUMMY DUMMY PASSWORD USED P768101
```

2. Identify all user IDs that will access HFS without supplying their password.
3. To activate the SURROGAT class support in IBM RACF, if it has not already been set up on your system, issue the following command:

```
SETRPTS CLASSACT(SURROGAT)
```

Note: You only have to activate this feature one time.

4. If you want to cache the SURROGAT profiles in storage to enable you to refresh and immediately put all IBM RACF changes in effect immediately, issue the following command:

```
SETRPTS RACLIST(SURROGAT)
```

Note: If you do not use the RACLIST option, the changes made during this procedure will not take effect until the next IPL.

5. To create the SURROGAT class profile for a particular user, issue the following command:

```
RDEFINE SURROGAT BPX.SRV.UUUUUUUU UACC(NONE)
```

where UUUUUUUU is the user ID you are creating a profile for.

6. Repeat Step 5 for each user ID that requires HFS support without a password with a SURROGAT profile.

Note: To define all users in one command, you can specify BPX.SRV.*.

7. To give a user the authority to create a thread-level security environment for another user, issue the following command:

```
PERMIT BPX.SRV.UUUUUUUU CLASS(SURROGAT) ID(CDIRECT) ACCESS(READ)
```

where the DTF user called CDIRECT is the user you are granting permission to create the security environment for another user called UUUUUUUU.

8. Repeat Step 8 for each user ID that requires HFS support without a password with a SURROGAT profile.

Note: To define all users in one command, you can specify BPX.SRV.*.

9. Verify that the DTF User ID has sufficient access to HFS files along with both IBM RACF access and z/OS UNIX System Services permissions.
10. If you are using the RACLIST option, issue the following command to refresh and put your changes in effect for the SURROGAT class:

```
SETRPTS RACLIST(SURROGAT) REFRESH
```

11. To check whether the DTF Userid has been defined to the BPX.SRV.uuuuuuuu SURROGAT class profile, use the following RLIST command:

```
RLIST SURROGAT BPX.SRV.uuuuuuuu AUTHUSER
```

where uuuuuuuu is the user ID whose requests IBM Connect:Direct needs to process.

The system displays the user ID (which should be the DTF Userid) and access rights of the user ID that can act as a surrogate for uuuuuuuu.



CAUTION: Be aware of the REMOTE.DUMMY.PASSWORD and Adjacent Node settings for Node to Node communication.

```
SAF022I - DGADABMB - Dummy password usage by Adjacent Node rejected.
          An attempt was made by an Adjacent Node to use a dummy
          password to authorize access to the Connect:Direct local
          node. If the Init Parm REMOTE.DUMMY.PASSWORD setting is
```


INTERNAL, only Adjacent Nodes having the INTERNAL attribute in the Netmap may use a dummy password for this purpose.

Run Job Security Exit

The Run Job security exit control point provides a standard interface for security verification of job streams before they are submitted to the job entry system. Specific implementation details include the following:

- The Run Job exit is implemented as an executable load module.
- The name of the load module is user-defined and cannot conflict with any IBM Connect:Direct load module names.
- Specify RUN.JOB.EXIT=(modname) in the initialization parameters to activate the Run Job exit.
- You must link-edit the module as re-entrant and place it in a load library that the IBM Connect:Direct DTF can access.
- Because information passed to the exit is located above the 16 megabyte line, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.

For additional information about exits, see [“IBM Connect:Direct Exits” on page 315](#).

Sample Run Job Security Exits

The \$CD.SDGASAMP library contains a sample source module for the most used z/OS security systems. Sample exit routines are:

- DGAXRACJ for IBM RACF and CA-TOP SECRET
- DGAXACRJ for CA-ACF2

The sample exits are designed to ensure that correct security information is coded on each JOB statement in the job stream.

- For IBM RACF and CA-TOP SECRET, a check is made for a valid USER and PASSWORD on each JOB card. If not found, a USER=submitter keyword is added to each JOB card.
- For CA-ACF2, a JOBFROM=submitter keyword is added immediately following each JOB card to ensure that the correct security information is transferred to each submitted job:

If you use one of these exits without coding a value for the RUN.JOB.EXIT initialization parameter, IBM Connect:Direct does not use the default for the RUNJOBID initialization parameter.

Note: Use the Run Job security exit to achieve user propagation for security checks when the job that executes is submitted by the user ID assigned to IBM Connect:Direct rather than the user ID that submitted the job. In most environments, this Exit is not needed.

Run Task Security Exit

The Run Task security exit control point provides a standard interface to verify that the user is authorized to run the specified program. IBM Connect:Direct passes the exit security information about the user, the program name, and the parameters being passed to the program. Specific implementation details include the following:

- The Run Task exit is implemented as an executable load module.
- The name of the load module is user-defined, but cannot conflict with any IBM Connect:Direct load module names.
- Specify RUN.TASK.EXIT=(modname) in the initialization parameters to activate the Run Task exit.
- You must link-edit the module as re-entrant and place it in a load library that the DTF can access.
- Because information passed to the exit by IBM Connect:Direct is located above the 16 megabyte line, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.

For additional information about exits, see [“IBM Connect:Direct Exits” on page 315](#).

Sample Run Task Security Exit

The \$CD.SDGASAMP library contains a sample source module for the most used security systems. Sample exit routines are:

- DGAXRACT for IBM RACF and CA-TOP SECRET
- DGAXACFT for CA-ACF2
- DGAXSAFT for CA-ACF2 using the Security Access Facility

You can use the sample exit as a model to implement specific requirements.

IBM Connect:Direct Secure Point-of-Entry

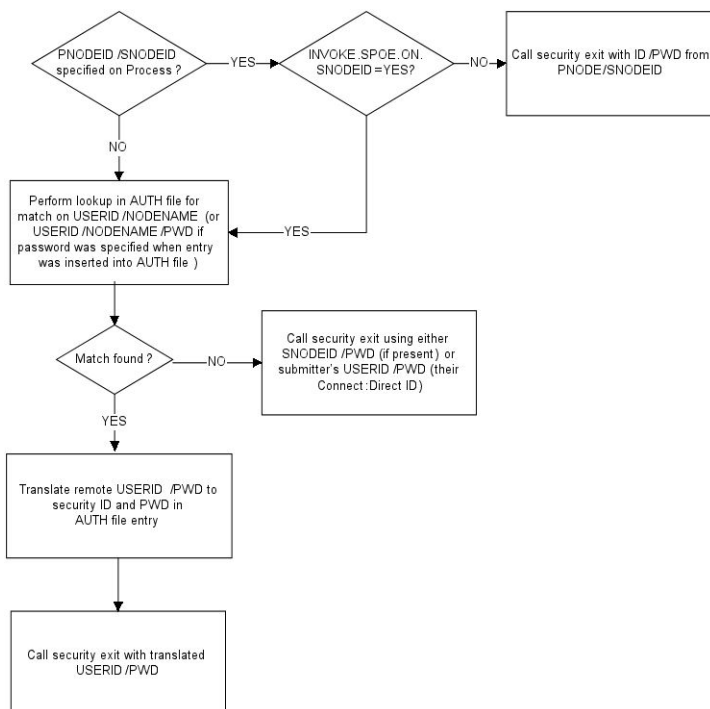
You need security on the local node because adjacent nodes need access to local nodes in order to transfer files. IBM Connect:Direct administrators have three options for security on transfers initiated at a remote node:

- No security for either functional authority or data protection
- Matching user ID/password combinations for all adjacent nodes
- SNODEID/SNODE password overrides on incoming access requests

Point-of-entry security secures the entry of an outside user to your system. It works with your current security setup (including all current exits) to provide additional security that addresses concerns about users from other nodes knowing a user ID and password combination on your system. Both data protection and IBM Connect:Direct functional authority are accomplished with exits.

Point-of-Entry Processing is internal within IBM Connect:Direct, and happens prior to calling the security exit for validations.

The following figure illustrates the flow of security checking for secure point-of-entry:



Security checks are made automatically for every incoming Process, so no parameter is needed to activate point-of-entry security. To implement point-of-entry, add user ID and node name combinations to your Authorization file. For instructions on manipulating the IBM Connect:Direct Authorization file, see [Authorization File](#).

The security exit determines if a secure point-of-entry translation was performed on a user ID by checking the bit SQIDXLAT of the SQCB control block (DGA\$SQCB macro in the \$CD.SDGAMAC library).

Point-of-Entry Concept

When a Process is submitted by another node, the receiving IBM Connect:Direct node has access to the user ID of the person who submitted the Process and the name of the node of the submitted Process.

For example, the local node is CD.HOUSTON, and a user SMITH submits a Process on CD.CHICAGO to copy a file to CD.HOUSTON. By placing an entry of SMITH/CD.CHICAGO into the local IBM Connect:Direct authorization file, the security administrator for CD.HOUSTON can associate this user with a valid user ID and password on the local system. The IBM Connect:Direct Authorization file has the following values:

USERID	=	SMITH
NODE	=	CD.CHICAGO
SECURITY ID	=	JONES
SECURITY PSWD	=	DALLAS

In this scenario, when user ID SMITH on node CD.CHICAGO submits a Process to run with node CD.HOUSTON, the functional authority and the data set validation for that Process are done under the authority of user ID JONES, which is a valid user ID on CD.HOUSTON. The user from the Chicago node never needs to know the related valid user ID and password on the CD.HOUSTON node.

USERID	=	SMITH
NODE	=	CD.HOUSTON
SECURITY ID	=	JONES
SECURITY PSWD	=	DALLAS

Secure Point of Entry Optional Variations

Note the following variations:

- If the CD.HOUSTON node enables SNODEID overrides and user SMITH puts an SNODEID parameter in the Process, the Authorization file is not checked and the translation of the user ID and password is not done.

Note: If the INVOKE.SPOE.ON.SNODEID initialization parameter is set to YES, then the Authorization file is checked and the user ID and password are translated.

For example, if the incoming Process in the previous example is coded with SNODEID=(BROWN,PWB), even if it is submitted by SMITH from CD.CHICAGO, the CD.HOUSTON node will send the user ID BROWN, not user ID JONES to the security subsystem for validation.

Note: To produce a completely secure point-of-entry security system, disable SNODEID overrides. To disable SNODEID overrides, specify SNODEID=NO in your stage 2 security exit.

- Although the point-of-entry system requires some maintenance of the Security ID and Security Password fields in the Authorization file, you can assign the same user ID and password combination on your system to multiple incoming users.

For instance, you can specify JONES/DALLAS as the user ID and password for all users coming into your node from CD.CHICAGO. In addition, if you are running a stage 1 signon exit, you can specify the security password for all users as IUI, BATCH, or STC, and avoid the need to update the Authorization file as the password changes.

- IBM® IBM Connect:Direct® for OpenVMS is not able to pass an OpenVMS password with the OpenVMS user ID. If you are using secure point-of-entry with incoming OpenVMS nodes, you must leave the User Password field blank in your z/OS authorization file, or all incoming OpenVMS Processes will fail.

Trusted Node Security

The Trusted Node Security feature enables you to enforce more restrictive security parameters when dealing with specific nodes in your network, enabling you to define each adjacent node in the network map as internal or external in its relationship to the local node of that network map.

When a Process begins execution, the security exit gets control and a bit in the Security Control Block (SQCB), SQEXTNOD, turns on if the adjacent node is defined as EXTERNAL. If the adjacent node is defined as INTERNAL, the bit turns off. Based on this information, the administrator can code the security exit to take the appropriate action.

In the adjacent node definition, the fifth positional parameter is required for the Trusted Node Security option. The parameter description follows:

Parameter	Description
EXTERNAL EXT	Indicates that the node is external.
INTERNAL INT	Indicates that the node is internal. This value is the default.

Cross-Domain Signon Environment

The cross-domain signon environment is an extension of the Trusted Node Security feature. This feature enables you to easily identify whether a signon is entering from an internal or external node.

You can use the same network map parameters for cross-domain Trusted Node Security as the node-to-node Trusted Node Security enhancement, for example, EXTERNAL or INTERNAL in the adjacent node definition. The SQEXTSGN bit in the security exit Control Block (SQCB) designates this feature.

When a cross-domain signon is entering from a node defined as EXTERNAL in the local node network map definition, the SQEXTSGN bit is on when the security exit gets control during signon processing. You can then modify the security exit to take whatever action is appropriate for that installation.

The DGASECUR macro, included in the \$CD.SDGASAMP library, contains the code to implement this enhancement. The lines of code with 117200 identify the Trusted Node Security feature in the cross-domain signon environment.

Data Direction Restriction

In addition to the Trusted Node feature, the Data Direction Restriction specifies whether each adjacent node can initiate a RECEIVE, SEND, or RECEIVE and SEND to or from the local node in the network map. The bits located in the SQCB, SQRECV, and SQSEND indicate the sending and receiving status.

In the adjacent node definition, the sixth positional parameter enables you to restrict the direction of data on a transfer with a specific adjacent node. This security applies regardless of where the Process is submitted, for example, local or remote node. The parameter descriptions follow:

Parameter	Description
RECEIVE RECV	Indicates that when the adjacent node initiates a transfer, it is only allowed to receive data from this node. It is never allowed to send data to this node.
SEND	Indicates that when the adjacent node initiates a transfer, it is only allowed to send data to this node. It is never allowed to receive data from this node.
BOTH	Indicates that when the adjacent node initiates a transfer, it is allowed to both send and receive data from this node. This value is the default.
NONE	Indicates that when the adjacent node initiates a transfer, it is neither allowed to send or receive data from this node.

The following example represents the Trusted Node Security and Data Direction Restriction features defined in the network map. The parameters are the fifth and sixth positional parameters in the adjacent node definition.

```

LOCAL.NODE=(CD.LOCAL LOCAPPL,,SUPUSRPW) -
TCQ=(CD.TCX CD.TCQ))
ADJACENT.NODE=(PARSESS=(4 2) (CD.LOCAL LOCAPPL -
APPLIDS=(1011 1012 1013))
ADJACENT.NODE=(PARSESS=(4 2) -
(CD.REMOTE RMTAPPL,,EXTERNAL,RECV) -
APPLIDS=(1011 1012 1013))

```

The following two bits are identified in the security exit:

Bit	Description
SQSNODE	Identifies if the node where the security exit is running is the SNODE for this Process. The bit is on if the node is the SNODE and off if the node is the PNODE.
SQIDXLAT	Identifies if a point-of-entry security ID translation was performed prior to calling the security exit. If a PNODEID/SNODEID is not specified when the Process is submitted and a match is found in the Authorization file for that USERID and NODE combination, then the bit turns on when the security exit gets control.

Implementing a CA-ACF2 Environment

When assembling both the stage 1 signon exit and the stage 2 security exit, you must provide the following data definition (DD) statements.

Procedure

1. For the assembly step, ensure that the SYSLIB concatenation contains the following information:

```

//SYSLIB DD DSN=ACF.MACLIB
// DD DSN=$CD.SDGAMAC
// DD DSN=SYS1.MODGEN
// DD DSN=SYS1.MACLIB

```

2. Replace \$CD with the high-level qualifier for your IBM Connect:Direct data sets.
3. For the link-edit step, provide the following DD statements.

```

//SYSLIB DD DSN=SYS1.ACFMOD
// DD DSN=SYS1.ACFAMOD
// DD DSN=$CD.SDGALINK

```

Note: You must have the High-Level Assembler for correct assembly. Do not specify NOALIGN as an option. The correct option is ALIGN.

4. Specify the IBM Connect:Direct DTF logon ID (LID) with the following attributes:

LID Attribute	Comment
MUSASS	Required for ACF2.
NON-CNCL	Only required if you are not running DGAMGSAF.
NO-SMC	Required.
SECURITY	Only required if NEWPASS=YES is specified for the stage 2 security exit.
ACCOUNT	Only required if NEWPASS=YES is specified for the stage 2 security exit.
JOBFROM	Only required if the Run Job statement is allowed and the Run Job exit is active.
STC	Required if IBM Connect:Direct is run as a started task.
RESTRICT	Optional.

LID Attribute	Comment
PROMPT	Optional. Enables IBM Connect:Direct to receive prompts from the operating system. For example, with the PROMPT attribute, IBM Connect:Direct receives a prompt for the password of a password protected data set if it was not supplied in the COPY statement DSN=filename/password.

If you are executing IBM Connect:Direct as a started task, CA-ACF2 monitors started tasks and the IBM Connect:Direct logon ID specifies STC=YES. See the GSO OPTS field STC/NOSTC in the CA-ACF2 Administrator's Guide for more information.

If you are using the program-pathing facility of CA-ACF2 that requires that the user logon ID be defined with the RESTRICT and SUBAUTH attributes, then the program name specified in the PROGRAM attribute for the user logon ID must be BPXPTATT for IBM Connect:Direct authorization.

The SAF interface requires definitions (SAFDEF) for both BPXPTATT and DGADRNT\$.

Implementing an IBM RACF Environment

When assembling both the stage 1 signon exit and the stage 2 security exit, you must provide the following DD statements.

Procedure

1. For the assembly step, ensure that the SYSLIB concatenation contains the following information:

```
//SYSLIB DD DSN=$CD.SDGAMAC
//      DD DSN=SYS1.MODGEN
//      DD DSN=SYS1.MACLIB
```

2. Replace \$CD with the high-level qualifier for your IBM Connect:Direct data sets.
3. For the link-edit step, provide the following DD statement.

```
//SYSLIB DD DSN=$CD.SDGALINK
```

Note: You must have Assembler H or the High Level Assembler for correct assembly. Do not specify NOALIGN as an option. The correct option is ALIGN.

4. Observe the following restrictions or requirements:
 - If the z/OS Task Control Block (TCB) Extension Feature is installed, give IBM Connect:Direct update authority to IBM Connect:Direct system files, such as the TCQ and network map. These prerequisites allow IBM Connect:Direct to use the Security Access Facility (SAF) of z/OS.
 - If you are using the IBM RACF PROGRAM ACCESS authority to set up access authority by program name and user ID, define the program name DMGATTIS to IBM RACF for IBM Connect:Direct.

Providing Program Access to Data Sets (PADS)

If your system has z/OS UNIX System Services or if you use PADS functionality in your security system, include all data sets in the IBM Connect:Direct JCL STEPLIB DD concatenation in your Program Control List (PCL). If any data set in the STEPLIB concatenation is not in the PADS list, the “dirty bit” will be turned on, and IBM Connect:Direct initialization will fail and display message SITA997I. Use the following procedure:

Procedure

1. Type the following command to display the data sets (libraries) in the PCL.

```
rlist program *
```

The access-controlled data sets are displayed.

```

CLASS      NAME
-----
PROGRAM    *
MEMBER      CLASS      NAME
-----
PMBR
DATA SET NAME                      VOLSER      PADS CHECKING
-----
CEE.SCEERUN                          NO
TCPIP.SEZALINK                       NO
USER01.HOST4100.LOADLIB              NO

```

2. Define the IBM Connect:Direct libraries to IBM RACF PADS. The following screen is an example of a definition.

```

RDEFINE PROGRAM ** UACC(READ) ADDMEM +
                    ('$CD.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM) REFRESH

```

Note: Refer to the IBM RACF documentation to verify IBM RACF command formats and keywords.

Implementing a CA-TOP SECRET Environment

About this task

When assembling both the stage 1 signon exit and the stage 2 security exit, you must provide the following DD statements.

Procedure

1. For the assembly step, ensure that the SYSLIB concatenation contains the following information:

```

//SYSLIB DD DSN=$CD.SDGAMAC
//        DD DSN=$CD.SDGALINK
//        DD DSN=$CD.SDGALIB

```

2. Replace \$CD with the appropriate high-level qualifier for your IBM Connect:Direct data sets.
3. For the link-edit step, provide the following DD statements.

```
//SYSLIB DD DSN=$CD.SDGALINK
```

Note: You must have Assembler H or the High-Level Assembler for correct assembly. Do not specify NOALIGN as an option. The correct option is ALIGN.

4. Add IBM Connect:Direct as a CA-TOP SECRET Facility.
5. Observe the following restrictions or requirements:

- If you are using CA-TOP SECRET Release 4 or later, issue the following commands.

```

TSS CREATE(NDM) NAME('...') DEPT(...)
TSS MASTFAC(NDM) FAC(STC) PASSWORD(NOPW)
TSS ADDTO(STC) PROC(NDM) ACID(NDM)
TSS PERMIT(NDM) DSN(NDM) ACCESS(ALL)

```

Issue TSS MODIFY commands to obtain the following list of attributes.

```
NDM PGM=DMG          ID=your choice
ATTRIBUTES=ACTIVE, SHRPRF, ASUBM, MULTIUSER, NOXDEF,
SIGN(M) NORNDPW NOAUDIT, RES, NOABEND,
NOPROMPT, NOTSOC
```

- If you are using CA-TOP SECRET Release 4 or later, and if the z/OS TCB Extension Feature is installed, IBM Connect:Direct only needs update authority to its system files, such as TCQ and network map. These prerequisites allow IBM Connect:Direct to use the SAF of z/OS. Otherwise, the ACID referenced previously must provide full access authority to all files IBM Connect:Direct accesses. Alternatively, you can identify IBM Connect:Direct in the privileged program name table as having access to all files by setting bit 6 (bypass password checking) in the program properties table (IEFSDPPT) to 1.

Configuring Firewall Navigation

Firewall navigation enables controlled access to a IBM Connect:Direct system running behind a packet-filtering firewall without compromising your security policies or those of your trading partners. You control this access by assigning a specific TCP or UDT source port number or a range of source port numbers with a specific destination address (or addresses) for IBM Connect:Direct sessions.

About this task

Before you configure source ports in the IBM Connect:Direct initialization parameters, you need to review the information in this section, especially if you are implementing firewalls for UDT.

Procedure

1. Coordinate IP address and associated source port assignment with the local firewall administrator before updating the firewall navigation record in the initialization parameters file.
2. Add the following parameters to the IBM Connect:Direct initialization parameters file as needed, based on whether you are using TCP or UDT:
 - TCP.SRC.PORTS
 - TCP.SRC.PORTS.LIST.ITERATIONS
 - UDP.SRC.PORTS
 - UDP.SRC.PORTS.LIST.ITERATIONS

In a IBM Connect:Direct/Plex environment, specify these parameters in the local initialization parameters file of the IBM Connect:Direct/Plex member that communicates with an external firewall.

3. Reinitialize Connect:Direct for z/OS.
4. Coordinate the specified port numbers with the firewall administrator at the remote site. These ports must also be available for IBM Connect:Direct communications on the firewall of your trading partner.

Firewall Navigation Rules Overview

Firewall rules need to be created on the local firewall to allow the local IBM Connect:Direct node to communicate with the remote IBM Connect:Direct node. A typical packet-filtering firewall rule specifies that the local firewall is open in one direction (inbound or outbound) to packets from a particular protocol with particular local addresses, local ports, remote addresses, and remote ports. Firewall navigation differs between TCP and UDT; as a result, firewall rules for TCP and UDT should be configured differently.

TCP Firewall Navigation Rules

In the following table, the TCP rules are presented in two sections: the first section applies to rules that are required when the local node is acting as a PNODE; the second section applies to rules that are

required when the local node is acting as an SNODE. A typical node acts as a PNODE on some occasions and an SNODE on other occasions; therefore, its firewall will require both sets of rules.

TCP PNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE session	Outbound	Local IBM Connect:Direct source ports	Remote IBM Connect:Direct listening port
TCP SNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports
SNODE session	Inbound	Local IBM Connect:Direct listening port	Remote IBM Connect:Direct source ports

UDT Firewall Navigation Rules

UDT firewall rules are applied to the UDP protocol. The recommended default firewall rule for UDP packets is to block packets inbound to the local system and outbound from the local system to prevent the confusion that could occur due to the callback feature of UDT session establishment.

In the following table, the UDT rules are presented in two sections: the first section applies to rules that are required when the local node is acting as a PNODE; the second section applies to rules that are required when the local node is acting as an SNODE. A typical node acts as a PNODE on some occasions and an SNODE on other occasions; therefore, its firewall will require both sets of rules.

UDT PNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE Session Request	Outbound	Local IBM Connect:Direct source ports	Remote IBM Connect:Direct listening port
PNODE Session	Outbound	Local IBM Connect:Direct source ports	Remote IBM Connect:Direct source ports
UDT SNODE Rules			
Rule Name	Rule Direction	Local Ports	Remote Ports
SNODE listen	Inbound	Local IBM Connect:Direct listening port	Remote IBM Connect:Direct source ports
SNODE session	Inbound	Local IBM Connect:Direct source ports	Remote IBM Connect:Direct source ports

Firewall Configuration Examples

In the firewall configuration examples for TCP and UDT, the following IP addresses and source ports will be used:

Note: The IP addresses in the examples are samples and are not intended to be valid IP addresses.

- The **local node** has IP address 222.222.222.222 and listening port 2264. Its source ports for communicating with the remote node are 2000–2200.
- The **remote node** has IP address 333.333.333.333 and listening port 3364. Its source ports for communicating with the local node are 3000–3300.

See [Session Establishment](#) for the differences between establishing a UDT or TCP session.

TCP Firewall Configuration Example

The IBM Connect:Direct administrator configures the **local node** to listen on port 2264, and the following initialization parameter settings are used to configure the local node's source ports:

- TCP.SRC.PORTS = (333.333.333.333, 2000–2200)
- TCP.SRC.PORTS.LIST.ITERATIONS = 1

This configuration specifies to use a source port in the range 2000–2200 when communicating with the remote node's address 333.333.333.333 and to search the port range one time for an available port. The local node will act as both a PNODE and an SNODE when communicating with the remote node.

Based on this scenario, the firewall rules for the local node are the following:

Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE session request	Outbound	2000–2200	3364
SNODE session	Inbound	2264	3000–3300

UDT Firewall Configuration Example

The IBM Connect:Direct administrator configures the **local node** to listen on port 2264, and the following initialization parameter settings are used to configure the local node's source ports:

- UDP.SRC.PORTS = (333.333.333.333, 2000–2200)
- UDP.SRC.PORTS.LIST.ITERATIONS = 1

This configuration specifies to use a source port in the range 2000–2200 when communicating with the remote node's address 333.333.333.333 and to search the port range one time for an available port. The local node will act as both a PNODE and an SNODE when communicating with the remote node.

Based on this scenario, the firewall rules for the local node are the following:

Rule Name	Rule Direction	Local Ports	Remote Ports
PNODE session request	Outbound	2000–2200	3364
PNODE session	Outbound	2000–2200	3000–3300
SNODE listen	Inbound	2264	3000–3300
SNODE session	Inbound	2000–2200	3000–3300

Blocking Outbound Packets

The recommended default rule for outbound UDP packets from the local system is to block the packets. If you do not follow this recommendation, port usage may, at first sight, appear to violate the firewall's inbound rules.

An example will help illustrate this situation. Suppose that in the example in the previous section:

- The local node is the SNODE.
- The default outbound rule allows all outbound UDP packets from the local system.
- The “SNODE session” rule is accidentally omitted.

Because of the callback feature of UDT session establishment, SNODE sessions are still likely to succeed on ports 2000–2200. This may cause confusion because ports 2000–2200 are blocked to inbound UDP packets.

If you use the recommended default outbound rule and apply the PNODE and SNODE rules described in the previous section, there will be no confusion about which port to use, and the UDT callback feature will function as designed, thus supporting reliability.

Session Establishment

Session establishment differs between TCP and UDT; these differences affect how you set up firewall rules and configure the firewall navigation initialization parameters in IBM Connect:Direct.

TCP Session Establishment

A IBM Connect:Direct TCP client contacts a IBM Connect:Direct TCP server on its listening port. The IBM Connect:Direct client scans the list of ports (specified using the TCP.SRC.PORTS initialization parameter) and looks for a port to bind to. The number of times IBM Connect:Direct scans the list is specified using the TCP.SRC.PORTS.LIST.ITERATIONS initialization parameter. If IBM Connect:Direct finds an available port, communication with the remote node proceeds.

UDT Session Establishment

When a IBM Connect:Direct UDT client contacts a IBM Connect:Direct UDT server on its listening port to request a session, the UDT server responds with a different server port to use for the session. The client attempts to contact the server on the session port. The IBM Connect:Direct client scans the list of ports (specified in the UDP.SRC.PORTS initialization parameter) and looks for an available port to bind to. The number of times IBM Connect:Direct scans the list is specified using the UDP.SRC.PORTS.LIST.ITERATIONS initialization parameter. If the IBM Connect:Direct client finds an available port, communication with the remote IBM Connect:Direct server proceeds. If a session cannot be established after a certain time interval, the server attempts to contact the client.

Common Problems in Establishing a Session

The following message indicates that IBM Connect:Direct cannot find an idle port.

```
SCPA001I - TGT.ADDR=nnn.nnn.nnn.nnn, TCP.SRC.PORTS exhausted
```

If this message occurs frequently, increase the pool of available ports.

You may need to reserve ports in TCP/IP to ensure they are available for firewall navigation. Reserve ports with the PORT statement in IBM TCP/IP. An example follows:

```
PORT
5000 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
5001 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
5002 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
5003 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
5004 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
5005 TCP connect-jobname NOAUTOLOG ; Connect:Direct Firewall pool
```

A pool of available ports that is too small can affect performance because the outbound connections are limited.

Troubleshooting Security Errors

Security errors can show up at signon, at Process start, or at any step of a Process. In general, a return code of 8 means that the error occurred on the PNODE, and a return code of C means the error occurred on the SNODE. This section tells you how to determine the cause of security errors.

Many IBM Connect:Direct security-related messages begin with the prefix RACF. This fact does not mean that IBM RACF was necessarily involved with the failure. It is merely a naming convention for IBM Connect:Direct message identifiers.

Often, it is helpful to run a security trace to determine exactly where and why a security failure occurred. See [Security Traces](#), for information on security traces.

Condition: Signon Denied

When you sign on from either batch or the IUI, you receive the a message that indicates the Stage 1 Signon Exit has failed.

Error	Cause	Action	Collect
RACF0971	<p>The Stage 1 Signon exit, DGACXSIG, failed to execute properly.</p> <p>In the Connect:Direct for z/OS IUI, verify that DGACXSIG is in an APF authorized library and correctly allocated.</p>	<p>Review both the short text and long text IBM Connect:Direct messages. If you are receiving the message during signon to the IUI, run a batch job after verifying that DGACXSIG is available to the job. For either batch or interactive signon, allocate the APISECUR DD as described in Security Traces. You will be able to view the progression of BLDLs, along with output showing where IBM Connect:Direct looked for DGACXSIG and the results from the search.</p>	<ul style="list-style-type: none">Output written to the APISECUR DD when you allocated APISECUR as described in Security Traces.

Condition: Lack Authority to Perform a Connect:Direct for z/OS Function

You attempt to perform a Connect:Direct for z/OS function but receive a message that says you are not authorized to perform that function.

Error Messages					
SCBB001I	SCBC030I	SCBD001I	SCBE001I	SCBF001I	SCBF063I
SCBF064I	SCBG001I	SCBH001I	SCBI001I	SCBJ001I	SCBK005I
SCBL001I	SCBN001I	SCBO001I	SCBP001I	SCBR002I	SCBS001I
SCBT005I	SCBU003I	SCBV001I	SCBW001I	SCBX001I	SCBY001I
SCPA008I	SFIA002I	SFIA003I	SRJA014I	SRTA008I	SSUB100I

Cause	Action	Collect
If you are running a Stage 2 security exit, your user ID is defined using an authorization bit mask that does not include the function you are attempting. A security trace will show you the general category of IBM Connect:Direct user assigned to your userid (administrator, operator, or general user). See Security Traces for more information about how to initiate a security trace. If you are using the IBM Connect:Direct authorization file, the functional authority of your userid does not include the function you are trying to perform.	Review both the short text and long text IBM Connect:Direct messages. Have the IBM Connect:Direct administrator at your site ensure that your userid has the authority necessary to perform the function, either by updating your userid record in the IBM Connect:Direct authorization file or by assigning the authority within the Stage 2 security exit	Output from a security trace showing the validation of your authority to perform the IBM Connect:Direct function

Condition: Access Denied to File or Data Set on COPY Step

You are denied access to a data set or a file on a COPY step.

Error	Cause	Action	Collect
RACF095I	The security subsystem either on your node (RC=8) or the remote node (RC=C) has denied your userid access to the data set.	Review both the short text and long text IBM Connect:Direct messages. Ensure that your userid has the correct access to the data set. If you continue getting this message, run a security trace. See Security Traces for more information about how to initiate a security trace. It might be necessary to use a PNODEID or SNODEID statement to send a valid userid and password to the security system.	<ul style="list-style-type: none"> Output from a security trace

Condition: User Record not Found in the Authorization Data Set

When you sign on to IBM Connect:Direct or submit a Process to another node, you receive message SAFA002I, The user record was not found in the Authorization Data Set.

Error	Cause	Action	Collect
SAFA002I	If you are using the IBM Connect:Direct authorization file for security, be aware that the key to that file is a combination of userid and node name. For example, if you are signed on to node CDA with userid USERA and transmitting to node CDB (not using an SNODEID override), the authorization file on CDB must have an entry for the userid USERA and node CDA.	Review both the short text and long text IBM Connect:Direct messages. Check the appropriate IBM Connect:Direct authorization file and verify that the correct userid/node combination is specified. User records in the IBM Connect:Direct authorization file can be added or modified with the Insert User or Update User commands.	None

Maintaining User Authorization

The IBM Connect:Direct Authorization Facility controls access to IBM Connect:Direct functions. It is an alternative source of security information to the Stage 1 Signon and Stage 2 Security exits. If you use the Authorization Facility, you must identify all IBM Connect:Direct users in all nodes that execute Processes.

The following example shows how the IBM Connect:Direct Authorization Facility is used. This example includes two IBM Connect:Direct nodes, called SYSTEMA and SYSTEMB. Joe has access to SYSTEMA under the IBM Connect:Direct user ID of JOEA and access to SYSTEMB under the IBM Connect:Direct user ID of JOEB.

Joe requires two entries in the Authorization Facility of each system, as illustrated in the following tables. These entries give him access to IBM Connect:Direct on both systems and the authorization to move files between both systems.

SYSTEMA Authorization File			
Node	User ID	Password	Authorized Functions
SYSTEMA	JOEA	[pswd]	Y,Y,N,Y
SYSTEMB	JOEB	[pswd]	N,Y

SYSTEMB Authorization File			
Node	User ID	Password	Authorized Functions
SYSTEMA	JOEA	[pswd]	Y,Y,N,Y
SYSTEMB	JOEB	[pswd]	N,Y

The combination of logical node name and user ID is used to access the Authorization file on the remote node to obtain the user ID, password, and associated functional authority.

For example, if Joe sent a file from SYSTEMA to SYSTEMB, the combination of SYSTEMA and JOEA enables him to access the authorization file on SYSTEMB. This entry then determines what IBM Connect:Direct functional authority Joe has on SYSTEMB when coming from SYSTEMA.

Note: The password is optional, but if specified in the Authorization Facility, you must specify it on the SIGNON command. Make the password available at Process execution time through the signon or SNODEID override.

Authorization File

The Authorization file contains user attribute default records. Each record defines which IBM Connect:Direct features the user can access for each node.

Individual users can access the User Authorization screen to display information about their own authorization record. The following table describes the User Authorization file maintenance commands:

Command	Description
INSERT USER	Inserts a User record in the Authorization file.
UPDATE USER	Updates a User record in the Authorization file.
DELETE USER	Deletes a User record from the Authorization file.
SELECT USER	Selects a User record from the Authorization file.

You can execute these commands through the batch interface, the Interactive user interface (IUI), or the operator interface.

INSERT USER and UPDATE USER Commands

The INSERT USER and UPDATE USER commands add or update a user in the IBM Connect:Direct Authorization file. The commands have the following format and parameters. The required parameters and keywords are in bold. (The NAME parameter is required only for INSERT USER.) Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	INSert USER UPDate USER	USERID = (nodename, user ID)
		NAME = 'username'
		ADD TYPE = Y <u>N</u>
		ALTER TYPE = Y <u>N</u>
		READ TYPE = Y <u>N</u>
		REMOVE TYPE = Y <u>N</u>
		ADD USER = Y <u>N</u>
		ALTER USER = Y <u>N</u>
		READ USER = Y <u>N</u>
		REMOVE USER = Y <u>N</u>
		CASE = Y N
		CDEL = Y <u>N</u> Note: Valid only in the Interactive user interface.
		CDELOFF = Y <u>N</u> Note: Valid only in the Interactive user interface.

Label	Command	Parameters
		CHange = Y <u>N</u> Note: The setting for this parameter overrides the setting specified for the GEN.CHG.PROCESS parameter.
		COPY = Y <u>N</u>
		DELPR = Y <u>N</u> Note: The setting for this parameter overrides the setting specified for the GEN.DEL.PROCESS parameter.
		EVENTCMD = Y <u>N</u>
		FLUSH = Y <u>N</u> Note: The setting for this parameter overrides the setting specified for the GEN.FLS.PROCESS parameter.
		GEN.CHG.PROCESS = Y <u>N</u> Note: Valid only in the Interactive user interface.
		GEN.DEL.PROCESS = Y <u>N</u> Note: Valid only in the Interactive user interface.
		GEN.FLS.PROCESS = Y <u>N</u> Note: Valid only in the Interactive user interface.
		GEN.SEL.PROCESS = Y <u>N</u> Note: Valid only in the Interactive user interface.
		GEN.SEL.STATISTICS = Y <u>N</u> Note: Valid only in the Interactive user interface.
		GVIEW = Y <u>N</u> Note: Valid only in the Interactive user interface.
		MAXSA = max signon attempts
		MODALS = Y <u>N</u>
		MODIFY = Y <u>N</u>
		NSUBMIT = Y <u>N</u>
		OVCRC = Y <u>N</u>
		PASSword = initial password
		PHone = 'phone number'
		PTKTDATA = (APPL profile name, secured signon key)
		RESETSA
		RUNJOB = Y <u>N</u>
		RUNTASK = Y <u>N</u>
		SECURERD= Y <u>N</u>

Label	Command	Parameters
		SECUREWR = Y <u>N</u>
		SECURITY = (security id, security pswd)
		SELNET = Y <u>N</u>
		SELPR = Y <u>N</u> Note: The setting for this parameter overrides the setting specified for the GEN.SEL.PROCESS parameter.
		SELSTAT = Y <u>N</u> Note: The setting for this parameter overrides the setting specified for the GEN.SEL.STATISTICS parameter.
		STATCMD = Y <u>N</u>
		STOPCD = Y <u>N</u>
		SUBMIT = Y <u>N</u>
		SUBMITTER.CMDS = (Y <u>N</u> , Y <u>N</u> , Y <u>N</u> , Y <u>N</u> , Y <u>N</u>) Note: This parameter is valid only in the batch interface.
		UPDNET = Y <u>N</u>
		REFSH = Y <u>N</u> Note: This parameter updates the INITPARM file.
		VIEW PROCESS = Y N Note: The setting for this parameter overrides the setting specified for the GEN.VIEW.PROCESS parameter.

Required Parameters

The following parameters are required for the INSERT USER command. The USERID parameter is required for the UPDATE USER command, but the NAME parameter is not.

Parameter	Description
USERID = (nodename, user ID)	<p>Specifies the user node and user ID of the record being added or updated. nodename specifies the user node of the User record. It is a 1–16 character alphanumeric string.</p> <p>Note: Connect:Direct for z/OS does not accept the following characters for the node name:</p> <ul style="list-style-type: none"> • < less than • ¬ logical not • , comma • > greater than • = equal sign • / forward slash • * asterisk • \ backward slash • ' single quote • " double quote • (open parenthesis •) close parenthesis <p>Important: Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$. _ - to ensure that the entries can be properly managed by Control Center, Connect:Direct Browser User Interface, or Sterling Connect:Direct Application Interface for Java (AIJ) programs.</p> <p>user ID specifies the user ID of the User record. The user ID can contain 1–64 characters of any kind.</p>
NAME = 'username'	<p>Specifies the full name of the user. The NAME is a string of 1–20 characters. If blanks are embedded in the NAME parameter, you must enclose the NAME in single quotation marks. This parameter is not required by the UPDATE USER command.</p>

Optional Authorization Record Parameters

Optional parameters for INSERT and UPDATE USER commands are separated into two categories: authorization record parameters and functional authorization parameters.

The following table describes the authorization record parameters for the INSERT USER and UPDATE USER commands. You can authorize each user to add, alter, read, or remove a record. Specify the authorization by indicating the action (ADD, ALTER, READ, REMOVE) followed by the record type. If you do not specify an action for a Type or User record, the action defaults to No.

Parameter	Description
ADD TYPE = Y <u>N</u>	Specifies whether the user is allowed to insert new records into the Type Defaults file.
ALTER TYPE = Y <u>N</u>	Specifies whether the user is allowed to update records in the Type Defaults file.
READ TYPE = Y <u>N</u>	Specifies whether the user is allowed to read records from the Type Defaults file.

Parameter	Description
REMOVE TYPE= Y <u>N</u>	Specifies whether the user is allowed to delete records from the Type Defaults file.
ADD USER = Y <u>N</u>	Specifies whether the user can insert new records into the Authorization file.
ALTER USER = Y <u>N</u>	Specifies whether the user is allowed to update records in the Authorization file.
READ USER = Y <u>N</u>	Specifies whether the user is allowed to read records from the Authorization file.
REMOVE USER = Y <u>N</u>	Specifies whether the user is allowed to delete records from the Authorization file.

Optional Functional Authorization Parameters

Following are the functional authorization parameters for INSERT USER and UPDATE USER:

Parameter	Description
CASE = Y N	Specifies whether accounting data, user ID, password, and data set name parameters are case sensitive. This choice overrides the case designation selected at session signon, and is in effect only for this command. The default is the designation made at session signon.
CDEL = Y <u>N</u>	Specifies whether the Confirm Delete/Suspend/Flush Command prompt displays for a particular user.
CDELOFF = Y <u>N</u>	Specifies whether the user can turn off the Confirm Delete/Flush/Suspend Command prompt for the current session. If you do not change the default of No to Yes, the user will always see the Confirm Delete/Flush/Suspend Command prompt and will not be given this option.
CHange = Y <u>N</u>	Specifies whether the user is allowed to use the CHANGE PROCESS command.
COPY = Y <u>N</u>	Specifies whether the user is allowed to use the COPY statement.
DELPR = Y <u>N</u>	Specifies whether the user is allowed to use the DELETE PROCESS command.
EVENTCMD = Y <u>N</u>	Specifies whether the user is allowed to use the Event Services Support commands.
FLUSH = Y <u>N</u>	Specifies whether the user is allowed to use the FLUSH PROCESS and SUSPEND PROCESS commands.
GEN.CHG.PROCESS = Y <u>N</u>	Specifies whether the user can change any Processes or only Processes that are submitted. If you specify GEN.CHG.PROCESS=Y, the user can only change Processes that he or she submitted (valid only in the IUI). Can be overridden by the CHange parameter setting.
GEN.DEL.PROCESS = Y <u>N</u>	Specifies whether the user can delete any Processes or only Processes that are submitted. If you specify GEN.DEL.PROCESS=Y, the user can only delete Processes that he or she submitted (valid only in the IUI). Can be overridden by the DELPR parameter setting.
GEN.FLS.PROCESS = Y <u>N</u>	Specifies whether the user can flush any Processes or only Processes that the user submitted. If you specify GEN.FLS.PROCESS=Y, the user can only flush Processes that he or she submitted (valid only in the IUI). Can be overridden by the FLUSH parameter setting.

Parameter	Description
GEN.SEL.PROCESS = Y <u>N</u>	Specifies whether the user can select any Processes or only Processes that the user submitted. If you specify GEN.SEL.PROCESS=Y, the user can only select Processes that he or she submitted (valid only in the IUI). Can be overridden by the SELPR parameter setting.
GEN.SEL.STATISTICS = Y <u>N</u>	Specifies whether the user can select any statistics or only statistics for Processes that the user submitted. If you specify GEN.SEL.STATISTICS=Y, the user can only select statistics for Processes that he or she submitted (valid only in the IUI). Can be overridden by the SELSTAT parameter setting.
GVIEW = Y <u>N</u>	Specifies whether the user can view only Processes submitted with a matching USERID or all Processes regardless of who submitted them. Can be overridden by the VIEW PROCESS parameter setting.
MAXSA = max signon attempts	Specifies the maximum number of signon attempts the user is allowed per hour. The range is 0–99. The default is 60. Zero (0) indicates no maximum number. (See the RESETSA parameter to see how to temporarily reset this value.)
MODALS = Y <u>N</u>	Specifies whether the user is allowed to use the modal statements IF, ELSE, EIF, GOTO, and EXIT.
MODIFY = Y <u>N</u>	Specifies whether the user is allowed to request traces and modify initialization parameters.
NSUBMIT = Y <u>N</u>	Specifies whether the user is allowed to use the SUBMIT statement to submit a Process.
OVCRC = Y <u>N</u>	Specifies whether the user is allowed to use the CRC statement to override the initial CRC settings.
PASSword = initial password	Defines the initial password for the user ID. The password is a 1–64 character alphanumeric string.
PHone = 'phone number'	Phone number of the user. Enclose the phone number in single quotation marks. The quotation marks allow for a space after the area code.
PTKTDATA=(APPL prof name, secured signon key)	Specifies the values required for the Stage 2 security exit to rewrite an IBM RACF PassTicket password. APPL prof name is the value specified when the profile is defined for the PTKTDATA class. The secured signon key is the value associated with the PTKTDATA class and the name specified in the APPL Prof name. See Generating IBM RACF PassTickets .
RESETSA	Specifies that the signon attempt count is reset to 0. (See the MAXSA parameter to see how to set the signon attempt count.) This parameter enables the user to try to sign on, even if he or she has previously exceeded the maximum number of signon attempts. This parameter is used in the UPDATE USER command only.
RUNJOB = Y <u>N</u>	Specifies whether the user is allowed to use the RUN JOB statement.
RUNTASK = Y <u>N</u>	Specifies whether the user is allowed to use the RUN TASK statement.
SECURERD = Y <u>N</u> or SECURE.READ = Y <u>N</u>	Specifies whether the user can display (read) the Connect:Direct Secure Plus Parameters file in Control Center.
SECUREWR = Y <u>N</u> or SECURE.WRITE = Y <u>N</u>	Specifies whether the user can update (write to) the Connect:Direct Secure Plus Parameters file in Control Center.

Parameter	Description
SECURITY = (security ID, security pswd)	<p>Specifies the security ID and security password to identify the file authorization of the user. Security support includes CA-ACF2, CA-TOP SECRET, and IBM RACF.</p> <p>Security ID specifies the 1–64 character security system ID for the user. This ID must meet the standards of the security subsystem at the location of the user. The security ID is required if this parameter is specified.</p> <p>Security pswd specifies the 1–64 character security system password for the user. This password must meet the standards of the security subsystem at the location of the user.</p>
SELNET = Y <u>N</u>	Specifies whether the user is allowed to use the SELECT NETMAP command.
SELPR = Y <u>N</u>	Specifies whether the user is allowed to use the SELECT PROCESS command.
SELSTAT = Y <u>N</u>	Specifies whether the user is allowed to use the SELECT STATISTICS command.
STATCMD = Y <u>N</u>	Specifies whether the user is allowed to use the STATISTICS COMMAND command.
STOPCD = Y <u>N</u>	Specifies whether the user is allowed to use the STOP CD command.
SUBMIT = Y <u>N</u>	Specifies whether the user is allowed to use the SUBMIT statement to define and submit within a Process.
SUBMITTER.CMDS = (Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)	<p>Specifies whether the user is allowed to issue certain commands concerning the Processes that he or she submitted (valid only in the batch interface). These commands are:</p> <ul style="list-style-type: none"> • SELECT PROCESS • DELETE PROCESS • FLUSH PROCESS • CHANGE PROCESS • SELECT STATISTICS <p>For more information, see the IUI definitions for these commands in Functional Authorization Parameters for the INSERT and UPDATE USER Command . (The IUI definitions begin with GEN and the command name is abbreviated.)</p>
REFSH = Y <u>N</u>	Specifies whether the user is allowed to update the initialization parameters file in Control Center.
UPDNET = Y <u>N</u>	Specifies whether the user is allowed to use UPDATE NETMAP.
VIEW PROCESS = Y <u>N</u>	Specifies whether the user is allowed to use VIEW PROCESS.

Inserting and Updating Users through the Batch Interface

To use the INSERT or UPDATE USER commands from the batch interface, follow this procedure.

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Example of Adding a User Record

The following example shows a User record for user ID Smith being added to the Authorization file:

```
SIGNON USERID=(user ID, password)

INSERT USER          USERID=(DALLAS, SMITH)  -
  NAME='RB SMITH'    PASS=XYZZY              -
  PH='214 555-5555'  -                      -
  ADD USER=Y        ALTER USER=Y            -
  READ USER=Y        REMOVE USER=Y          -
  SUBMIT=Y           -                      -
  SUBMITTER.CMDS=(Y Y Y Y N Y)
SIGNOFF
```

In the example definition, the user Smith can perform the following functions:

- Add users to the Authorization file
- Update and read User records
- Delete User records
- Define and submit Processes for execution
- Select, delete, flush/suspend, and change submitted Processes

In the example, Smith cannot perform the Select Statistics command on any Processes, regardless of who submitted them. Smith's initial password is XYZZY, and Smith's phone number is (817) 555-5555.

Example of Updating a User Record

The following commands update the record of a user named Smith in the Authorization file.

```
UPDATE USER          USERID=(DALLAS, SMITH)  -
  NAME='RB SMITH'    -                      -
  PASS=XYZZY         -                      -
  PH='214 555-5555'  -                      -
  ADD USER=Y        ALTER USER=Y            -
  READ USER=Y        REMOVE USER=Y          -
  CH=Y              FLUSH=Y  DELPR=Y
```

With these updates, Smith can perform the following functions:

- Add Users to the Authorization file
- Update and read User records
- Delete User records
- Change a Process in the TCQ
- Delete an executing Process from the TCQ
- Delete an inactive Process from the TCQ

Inserting and Updating Users through the IUI

Use the Insert/Update/Select/Delete User Record screen (following) to insert, update, select, or delete a record. Select option IU from the Administrative Options Menu to access this screen.

CD.ART CMD ==>	INSERT/UPDATE/SELECT/DELETE USER RECORD		15:04
FUNCTION ==> SEL ('I'-INS, 'U'-UPD, 'S'-SEL , 'D'-DEL)			
ENTER USER INFORMATION: NAME ==> _____			
USER ID ==>	_____	PHONE ==>	_____
USERNODE==>	CD.ART		
PASSWORD==>	_____		
SEC ID ==>	_____		
SEC PASS==>	_____		
MAX SIGNON ATTEMPTS==>	60	PASSTICKET DATA ==>	(_____ , _____)
DO YOU WANT VALUES FOR THIS COMMAND TO BE CASE SENSITIVE? ==> NO_			
DEFINE USER FUNCTIONS: (RESPOND WITH 'Y'-YES OR 'N'-NO)			
FLUSH PROCESS => -	SELECT NETMAP => -	UPDATE NETMAP => -	
INSERT USER => -	SELECT PROCESS => -	MODALS FUNCTION => -	
DELETE USER => -	SUBMIT PROCESS => -	RUNTASK FUNCTION => -	
SELECT USER => -	SUBMIT WITHIN PROC => -	INSERT TYPE => -	
UPDATE USER => -	RUNJOB FUNCTION => -	DELETE TYPE => -	
COPY FUNCTION => -	CONTROL TRACING => -	SELECT TYPE => -	
CHANGE PROCESS => -	STOP Connect:Direct=> -	UPDATE TYPE => -	
DELETE PROCESS => -	SELECT STATISTICS => -	GEN.FLS.PROCESS => -	
STAT COMMAND => -	GEN.DEL.PROCESS => -	RESET SIGNON => N	
GEN.SEL.PROCESS => -	GEN.SEL.STATISTICS => -	VIEW PROCESS => -	
GEN.CHG.PROCESS => -	EVENT COMMAND => -	CRC OVERRIDES => -	
GEN.VIEW.PROC => -	CONFIRM DELETE => -	CONFIRM DEL OFF => -	

The DEFINE USER FUNCTIONS portion of the screen is scrollable. More + or - indicates additional data. Press PF8 to scroll forward. Press PF7 to scroll back. See the description of the INSERT and UPDATE USER command parameters in “Optional Functional Authorization Parameters” on page 225 for the valid values of the fields, or press the PF1 key for Help.

Deleting Users from the Authorization File

The DELETE USER command removes a User record from the Authorization file. Following is the command format and parameters. The required parameters and keywords appear in bold print.

Label	Command	Parameters
(optional)	DE lete USER	WHERE (
		USERID = (nodename, user ID) (list)
)
		CASE = YES NO

Parameter	Description
WHERE (USERID = (nodename, user ID) (list))	<p>This parameter specifies which user records to delete.</p> <p>USERID = (nodename, user ID) (list)</p> <p>This parameter specifies the record to delete from the IBM Connect:Direct Authorization file.</p> <p>nodename specifies the node ID of the User record that is searched. The nodename is a 1–16 character alphanumeric string.</p> <p>Note: Connect:Direct for z/OS does not accept the following characters for the node name:</p> <ul style="list-style-type: none"> • < less than • ¬ logical not • , comma • > greater than • = equal sign • / forward slash • * asterisk • \ backward slash • ' single quote • " double quote • (open parenthesis •) close parenthesis <p>Important: Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$. _ - to ensure that the entries can be properly managed by Control Center, Connect:Direct Browser User Interface, or Sterling Connect:Direct Application Interface for Java (AIJ) programs.</p> <p>The user ID parameter specifies the user ID of the User record. The complete user ID consists of the nodename and the user ID enclosed in parentheses and separated by a comma.</p> <p>The list parameter specifies a list of user IDs.</p>

CASE is the only optional parameter.

Parameter	Description
CASE = Yes No	This parameter specifies whether nodename and user ID parameters are case sensitive. This choice overrides the case designation selected at session signon and is in effect only for this command. The default is the designation made at session signon.

Deleting Users through the Batch Interface

About this task

To use the DELETE USER command from the batch interface , follow this procedure.

Procedure

1. Place your command in the batch job stream.

2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Example of Deleting a User Record

The following example shows how to delete single and multiple User records:

```
* DELETES A SINGLE User record
  DELETE USER WHERE (USERID=(MPLS, SMITH))

* DELETES MULTIPLE User records
  DELETE USER WHERE (USERID=(DALLAS, JONES), -
    (MPLS, SMITH), (CHICAGO, BROWN))
```

Deleting Users through the IUI

About this task

You can delete a user in the IBM Connect:Direct IUI using the Delete A User Record screen or the Insert/Update/Select/Delete User Record screen.

To delete a User record using the Insert/Update/Select/Delete User Record screen, select the IU option from the Administrative Options Menu. See the description of the DELETE USER command parameters for the valid values of the fields on this screen, or pressthe PF1 key for Help.

You can delete up to four user records simulataneously on the Delete A User Record screen.

To delete a user from the Delete A User Record screen, follow this procedure.

Procedure

1. Select option DU from the Administrative Options Menu to display the Delete A User Record screen.
2. Type the user ID and user node of the records to delete.
3. Press ENTER.
4. Verify your results.

Selecting User Information from the Authorization File

The SELECT USER command displays a User record in the Authorization file. You can specify the search criteria and the format in which the information is presented.

The command has the following format and parameters. Required parameters and keywords appear in bold. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SElect USER	WHERE (
		USERID = (nodename, user ID)
		(generic (list))
		EXCLUDE = (AUTH)
)
		PRint <u>TABLE</u>
		CASE = YES <u>NO</u>

WHERE is the only required parameter for the SELECT USER command and USERID is the only required subparameter.

Parameter	Description
<p>WHERE</p> <p>(USERID = (nodename, user ID) (generic list))</p> <p>EXCLUDE = (AUTH)</p>	<p>This parameter specifies which User records you want to examine.</p> <p>USERID = (nodename, user ID) (generic list)</p> <p>This parameter specifies the record to search for in the Authorization file. This subparameter of the WHERE parameter is required. The complete user ID consists of the nodename and the user ID enclosed in parentheses and separated by a comma.</p> <p>nodename specifies the node ID of the User record that is searched. Type a 1–16 character alphanumeric string. If the user node is not specified, nodename defaults to the IBM Connect:Direct system that receives the command.</p> <p>Note: Connect:Direct for z/OS does not accept the following characters for the node name:</p> <ul style="list-style-type: none"> • < less than • ¬ logical not • , comma • > greater than • = equal sign • / forward slash • * asterisk • \ backward slash • ' single quote • " double quote • (open parenthesis •) close parenthesis <p>Important: Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$. _ - to ensure that the entries can be properly managed by Control Center, Connect:Direct Browser User Interface, or Sterling Connect:Direct Application Interface for Java (AIJ) programs.</p> <p>user ID specifies the user ID of the User record.</p> <p>generic specifies generic selection of user IDs. To specify user nodes and user IDs generically, type a 1–7 character alphanumeric string with the first character alphabetic, plus an asterisk (*). For instance, if you specify a user ID of B*, examine records for BLACK, BRADFORD, and BROWN.</p> <p>list specifies a list of user IDs.</p> <p>EXCLUDE = (AUTH)</p> <p>This parameter specifies that the function-by-function authorization description is not included in the output. This subparameter of the WHERE parameter is not required.</p>

The following table describes the optional parameters for the SELECT USER command:

Parameter	Description
PRint TABLE	<p>This parameter specifies the output destination.</p> <p>PRINT specifies that the output of the SELECT USER command is printed rather than displayed. Printed output is in tabular format, the same as that produced by the TABLE parameter. Output is routed to the destination specified in the PRINT keyword of the IBM Connect:Direct SIGNON command.</p> <p>TABLE specifies that the output of the SELECT USER command is stored in a temporary file in tabular format and is displayed upon successful completion of the command. The default for the output is TABLE.</p>
CASE = Yes No	<p>This parameter specifies whether parameters associated with nodename and user ID are case sensitive. This choice overrides the case sensitivity designation selected for the session at signon and is in effect only for this command. The default is the designation made at session signon.</p>

Selecting a User through the Batch Interface

About this task

To use the SELECT USER command from the batch interface, follow this procedure.

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Results

The following command searches for user BILL at the local (default) node.

```
SELECT USER WHERE (USERID=(, BILL))
```

Selecting a User through the IUI Interface

About this task

The Select a User Record screen enables you to simultaneously display user information and authorized functions for up to four users. You can also use the Insert/Update/Select/Delete User Record screen (option IU) to select users. See the SELECT USER command parameters in [“Selecting User Information from the Authorization File”](#) on page 231 for a description of the valid values for the fields or press PF1 for Help.

Procedure

1. From the Administrative Options Menu, select option SU to display the Select a User Record screen and press Enter.
2. Type the user ID and user node for the records you want to view.
3. Indicate the output destination, the case sensitivity, and whether you want to exclude the function-by-function authorization description.
4. Press Enter.

Maintaining the Type File

The Type file contains file attribute records. Each record is associated with a Type key. The key is used by the TYPE parameter in the COPY statement to create new files or access existing files.

The following example illustrates a Copy Process using the TYPE parameter.

```
COPY1 COPY FROM (DSN=MYFILE) -  
          TO (DSN=YOURFILE TYPE=TEXT)
```

The Type file serves two purposes:

- Saves retyping parameters such as DCB, DISP, and SPACE within Processes for files with common attributes.
- Facilitates the use of previously-defined attribute specifications of non-z/OS systems. This usage is especially useful for remote users who are not familiar with z/OS data set organizations and allocation parameters.

The Type key that is referenced in the TYPE= parameter must be in the Type file on the destination system, which is the system responsible for allocating the new file.

To maintain the Type file, use the INSERT TYPE, UPDATE TYPE, DELETE TYPE, and SELECT TYPE commands. Type these commands through the IUI, Batch or Operator interface.

Overriding File Attributes

If you specify file attributes in conjunction with the TYPE parameter on the COPY statement, the parameters in the COPY statement override similar parameters in the Type record. Use this functionality when you want to override a specific Type record subparameter.

Type Keys

Four predefined Type keys are provided to communicate with other IBM Connect:Direct nodes:

- TEXT
- DF
- DF2
- BINARY

The four Type keys contain file allocation information as defined in the following figure.

```
TYPE KEY => TEXT  
  DISP=(RPL,CATLG,DELETE)  
  DCB=(DSORG=PS,LRECL=255,BLKSIZE=2554,RECFM=VB)  
  SPACE=(TRK,(10,10))  
  UNIT=SYSDA  
  
TYPE KEY => DF (Data File)  
  DISP=(RPL,CATLG,DELETE)  
  DCB=(DSORG=PS,LRECL=255,BLKSIZE=2554,RECFM=VB)  
  SPACE=(TRK,(10,10))  
  UNIT=SYSDA  
  
TYPE KEY => DF2 (Data File 2)  
  DISP=(RPL,CATLG,DELETE)  
  DCB=(DSORG=PS,LRECL=80,BLKSIZE=3120,RECFM=FB)  
  SPACE=(TRK,(10,10))  
  UNIT=SYSDA  
  
TYPE KEY => BINARY  
  DISP=(RPL,CATLG,DELETE)  
  DCB=(DSORG=PS,BLKSIZE=6144,RECFM=U)  
  SPACE=(TRK,(10,10))
```

INSERT TYPE and UPDATE TYPE Commands

Use the INSERT TYPE command to insert a new Type record into the Type file. Use the UPDATE TYPE command to update an existing Type record. These commands use the following format and parameters. The required parameters appear in bold. Default values for parameters are underlined.

Label	Command	Parameters
(optional)	INSert TYPE UPDate TYPE	TYPEKEY =typekey
		DCB=(BLKSIZE = no. bytes ,DSORG = (DA PS PO VSAM) ,LRECL = no. bytes ,RECFM = record format)
		DISP=((<u>NEW</u> OLD MOD RPL SHR) (,KEEP , <u>CATLG</u> , DELETE) (,KEEP , <u>CATLG</u> , DELETE))
		DSNTYPE=PDS LIBRARY BASIC LARGE EXTPREF EXTREQ
		AVGREC=(U K M)
		DATACLAS=data_class_name
		KEYLEN=bytes
		KEYOFF=offset_to_key
		LIKE=model_data_set_name
		LRECL=bytes
		MGMTCLAS=management_class_name
		RECORD=(KS ES RR LS)
		SECMODEL=(profile_name [,GENERIC])
		STORCLAS=storage_class_name
		SPACE=(CYL TRK <u>blk</u> , (prim, sec, (dir)) (,RLSE ,(CONTIG ,)(ROUND)) (ave_rec_len,(primary_rcds,secondary_rcds))
		UNIT=unit type
		VOL=SER = volume serial number
		IOEXIT=exitname (exitname[,parameter,parameter...])

Following is the required parameter for the INSERT TYPE or UPDATE TYPE command.

Parameter	Description
TYPEKEY=typekey	Name associated with the entry being added or updated. The Type key is a 1–8 character alphanumeric string. The first character must be alphabetic.

Following are the optional parameters for the INSERT TYPE or UPDATE TYPE command:

Parameter	Description
DCB= ([BLKSIZE=no.bytes, DSORG=[DA PS PO VSAM], LRECL=no.bytes, RECFM=record format])	<p>This parameter specifies DCB information associated with the data set name on the COPY statement.</p> <p>BLKSIZE specifies the length of the block in bytes.</p> <p>DSORG specifies the file organization. File organizations supported are DA, PS, PO, and VSAM.</p> <p>LRECL specifies the length of the records in bytes.</p> <p>RECFM specifies the format of the records in the file. Specify any valid format, such as F, FA, FB, FBA, FBM, FM, U, V, VB, VBA, VBM, and VBS.</p> <p>Any file attributes specified on the COPY statement take precedence over those in the Type file. If you do not include attributes on the COPY statement, attributes in the Type file take precedence. If attributes are defined in the COPY TO or the Type file, the attributes are taken from the FROM side (source).</p>

Parameter	Description
DISP= ([NEW OLD MOD RPL SHR] [,KEEP ,CATLG ,DELETE] [,KEEP ,CATLG ,DELETE])	<p>This parameter specifies the default destination file status on the receiving node.</p> <p>The first DISP subparameter specifies the status of the file. Only the OLD and RPL dispositions apply to VSAM files.</p> <p><u>NEW</u> (default) specifies that the Process step creates the destination file.</p> <p>OLD specifies that the destination file existed before the Process began executing and that the Process is given exclusive control of the file. The destination file can be a VSAM file, a SAM file, or a PDS (IBM Connect:Direct for z/OS).</p> <p>MOD specifies that the Process step modifies the SAM file by adding data to the end of the file. If a system failure occurs when MOD is specified, the system is designed not to restart because data loss or duplication is difficult to detect.</p> <p>RPL specifies that the destination file replaces any existing file or allocates a new file. You can specify DISP=RPL for SAM or VSAM files. If the file is VSAM, you must define it with the REUSE attribute. You cannot specify RPL if VOL=SER is specified.</p> <p>SHR specifies that the source file existed before the Process began executing and that the file can be used simultaneously by another job or Process.</p> <p>The second subparameter specifies the normal termination disposition. It does not apply to VSAM files.</p> <p>KEEP specifies that the system keeps the file after the Process step completes. If DISP=(NEW,KEEP), you must also specify a volume serial number.</p> <p><u>CATLG</u> (default) specifies that the system keeps the file after the Process step completes and places an entry in the catalog.</p> <p>DELETE specifies that the system deletes the file after the Process step completes.</p> <p>The third subparameter specifies abnormal termination disposition. It does not apply to VSAM files. This subparameter has no default.</p> <p>KEEP specifies that the system keeps the file after the Process terminates abnormally.</p> <p><u>CATLG</u> (default) specifies that the system keeps the file after the Process step terminates abnormally and places an entry in the catalog.</p> <p>DELETE specifies that the system deletes the file after the Process if IBM Connect:Direct terminates abnormally.</p> <p>The third subparameter specifies abnormal termination disposition. It does not apply to VSAM files. This subparameter has no default.</p>

Parameter	Description
DSNTYPE = LIBRARY PDS BASIC LARGE EXTPREF EXTREQ	<p>Defines a specific data set organization for an SMS controlled data set of a PARTITIONed type.</p> <p>LIBRARY specifies a partitioned data set extended (PDSE).</p> <p>PDS specifies a partitioned data set.</p> <p>LARGE specifies specifies a sequential data set which can contain more than 65535 tracks per volume.</p> <p>BASIC specifies a sequential data set which can have no more than 65535 tracks per volume.</p> <p>EXTPREF specifies that the extended attribute is preferred. If an extended format data set cannot be allocated, a data set is created without the attribute.</p> <p>EXTREQ specifies that the extended attribute is required. If an extended format data set cannot be allocated, a data set is not created.</p>
AVGREC=(U K M)	<p>Requests that IBM Connect:Direct allocate the data set in records. The primary and secondary space quantities represent the number of records requested in units, thousands, or millions of records. This parameter is mutually exclusive with the TRK, CYL, and ABSTR subparameters of the SPACE keyword.</p> <p>U specifies a record request where primary and secondary space quantities are the number of records requested. It is a multiple of 1.</p> <p>K specifies a record request where primary and secondary space quantities are the number of records requested in thousands of records. It is a multiple of 1024.</p> <p>M specifies a record request where primary and secondary space quantities are the number of records requested in millions of records. It is a multiple of 1,048,576.</p>
DATACLAS=data_class_name	<p>Requests the data class for a new data set. The class selected must be previously defined by the SMS administrator. You can use this keyword with VSAM data sets, sequential data sets, or partitioned data sets.</p> <p>data_class_name specifies the 1–8 character name of the data class to which this data set belongs. The name of the data class is assigned by the SMS administrator.</p>
KEYLEN=bytes	<p>This parameter specifies the length of the keys in the file. This keyword is not restricted as a subparameter of the DCB keyword to support use with VSAM KS data sets.</p> <p>bytes specifies the length in bytes of the keys used in the file. You must specify this value with a decimal integer from 0–255 for non-VSAM data sets or 1–255 for VSAM data sets.</p>
KEYOFF=offset_to_key	<p>This parameter specifies the offset within the record to the first byte of the key in a new VSAM KS data set. The relative first byte of the record is byte 0. The range is 0–32760.</p> <p>offset_to_key specifies the position of the first byte of the key in the record.</p>

Parameter	Description
LIKE=model_data_set_name	<p>Requests that allocation attributes for a new data set are copied from an existing cataloged data set. Any or all of the following attributes are copied to the new data set: REORG or RECFM, LRECL, KEYLEN, KEYOFF, DSNTYPE, AVGREC, and SPACE. Any attributes specified for the data set override the values from the model data set. Neither EXPDT nor RETPD is copied from the model data set.</p> <p>model_data_set_name specifies the name of the data set from which the allocation attributes are copied.</p>
LRECL=bytes	<p>This parameter specifies the length in bytes of the records in the new data set. This parameter is allowed outside of the DCB keyword to allow use with SMS VSAM data sets. Do not specify LRECL with REORG=LS type data sets.</p> <p>bytes specifies the length of the records in the data set. For a non-VSAM data set, this length is 1–32760 bytes. For VSAM data sets, this length is 1–32761 bytes. The LRECL must be longer than the KEYLEN value for VSAM KS data sets.</p>
MGMTCLAS= management_class_name	<p>Determines to which of the previously defined management classes this new data set belongs. The attributes in this class determine such things as when a data set is migrated and when the data set is backed up.</p> <p>management_class_name specifies the 1–8 character name of the management class to which this data set belongs. The name of the management class is assigned by the SMS administrator.</p>
REORG=(KS ES RR LS)	<p>Defines the organization of records in a new VSAM data set. If REORG is not specified, then SMS assumes that the data set is either a physical sequential (PS) data set or a partitioned (PO) data set.</p> <p>KS specifies a VSAM key-sequenced data set</p> <p>ES specifies a VSAM entry-sequenced data set</p> <p>RR specifies a VSAM relative record data set</p> <p>LS specifies a VSAM linear space data set</p>
SECMODEL= (profile_name,GENERIC)	<p>Copies an existing IBM RACF profile as the discrete profile for this new data set. The following information is copied along with the profile: OWNER, ID, UACC, AUDIT/GLOBALAUDIT, ERASE, LEVEL, DATA, WARNING, and SECLEVEL.</p> <p>profile_name specifies the name of the model IBM RACF profile, discrete data set profile, or generic data set profile that is copied to the discrete data set profile created for the new data set.</p> <p>GENERIC identifies that the profile_name refers to a generic data set profile.</p>

Parameter	Description
STORCLAS=storage_class_name	<p>This parameter specifies the storage class to which the data set is assigned. The SMS administrator must define the storage class name to the SMS system by the SMS administrator. The storage class defines a storage service level for the data set and replaces the UNIT and VOLUME keywords for non-SMS data sets. You cannot use JCL keywords to override any of the attributes in the storage class. You can use an Automatic Class Selection (ACS) routine to override the specified class.</p> <p>storage_class_name specifies the 1–8 character name of the storage class to which this data set is assigned.</p>
SPACE= (CYL TRK blk, (prim, [sec],[dir]) [,RLSE ,] [,CONTIG ,] [,ROUND]) (ave_rec_len,[primary_rcds, secondary_rcds])	<p>This parameter specifies the amount of storage allocated for new files on the destination node. If SPACE is specified, the DISP of the destination file must be NEW. If SPACE is not specified and the DISP is either NEW or CATLG, space allocation defaults to the value obtained from the source file. The default is blk (blocks) with the ROUND option, which provides device-independent space allocation.</p> <p>If the AVGREC keyword is specified, the allocation of the data set is done on a record size basis instead of TRK, CYL, or BLK. This restriction is also true when the AVGREC keyword is present in the COPY statement.</p> <p>CYL specifies that space is allocated in cylinders.</p> <p>TRK specifies that space is allocated in tracks.</p> <p>blk specifies that space is allocated by the average block length of the data. The system computes the number of tracks to allocate. If the subparameter ROUND is also specified, the system allocates the space in cylinders. ROUND is preferred because allocation is performed on cylinders in a device-independent manner.</p> <p>prim specifies the primary allocation of storage.</p> <p>sec specifies the secondary allocation of storage.</p> <p>dir specifies the storage allocated for the PDS directory.</p> <p>RLSE releases the unused storage allocated to the output file.</p> <p>CONTIG specifies that the storage for the primary allocation must be contiguous.</p> <p>ROUND specifies that the storage allocated by average block length be rounded to an integral number of cylinders.</p> <p>ave_rec_len specifies the average record length, in bytes, of the data. IBM Connect:Direct computes the BLKSIZE and the number of tracks to allocate. The record length must be a decimal value from 1–65535.</p> <p>primary_rcds specifies the number of records that the data set contains. IBM Connect:Direct uses this number and the value of the AVGREC keyword to compute the primary space allocation.</p> <p>secondary_rcds specifies the number of additional records to allocate space for when the primary space is exhausted. IBM Connect:Direct uses this value and the AVGREC keyword to compute the number of tracks to allocate.</p>
UNIT = unit type	<p>This parameter indicates the unit address, device type, or user-assigned group name that contains the data. For SAM-to-SAM copies, where the destination file is new and the UNIT parameter is not coded with the TO parameter, the device type from the source file is used.</p>

Parameter	Description
VOL = SER = volume serial number	This parameter specifies the volume serial number containing the file. If VOL=SER is not specified with the FROM parameter, you must catalogue the file.
IOEXIT = exitname (exitname [,parameter, parameter,...])	<p>This parameter indicates that a user-written program is given control to perform I/O requests for the associated data.</p> <p>exitname specifies the name of the user-written program given control for I/O related requests for the associated data. The character length for IOEXIT is a variable of 1–510 characters.</p> <p>parameter (parameter,parameter,...) specifies a parameter, or list of parameters, passed to the specified exit. A parameter consists of a data type followed by the value in single quotes, for example C'ABC'. For a full description of valid parameter formats, see the RUN TASK statement parameters in the Connect:Direct Process Language help.</p>

Inserting and Updating Type Files through the Batch Interface

About this task

To use the INSERT TYPE or UPDATE TYPE command from the batch interface, perform the following steps:

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

The following example adds a Type record named NEWALLOC to the Type file:

```
INSERT TYPE  TYPEKEY=NEWALLOC -
              DCB=(DSORG=PS) -
              DISP=(NEW,CATLG) -
              UNIT=3380
```

IBM Connect:Direct users can then use the NEWALLOC Type key in a COPY command to allocate a new physical sequential file on a 3380 unit device that is cataloged on normal termination.

The following example updates a record in the Type file. When referring to the NEWALLOC Type key, the destination file is an existing PS file allocated on a 3390 disk pack:

```
UPDATE TYPE  TYPEKEY=NEWALLOC -
              UNIT=3390
```

Inserting and Updating Type Files through the IUI

About this task

To access the Type Record Selection List to perform this procedure, follow these steps.

Procedure

1. Select option IT from the Administrative Options Menu. The Insert/Update Type Record screen is displayed.

The Insert/Update Type Record screen prompts you for the action to perform (Insert or Update) and the Type key.

2. Type the requested information and press ENTER. IBM Connect:Direct displays an error message if you attempt to insert an existing record or update a nonexistent record.
3. From the Type Record Selection List screen, select either the General Dataset Attributes, IOEXIT Parameters, or the SMS/VSAM Attributes option by typing an S next to the appropriate entry and pressing **Enter**.

node.name	Type Record Selection List	hh:mm
CMD ==>		
Operation	==> UPDATE	
Type Key	==> X	
	- General Dataset Attributes	
	- IOEXIT Parameters	
	- SMS/VSAM Attributes	
Select the entries above to view and/or update the respective parameters.		
Enter the END command to perform the updates.		
-OR-		
Enter the CANcel command to abandon your changes.		

4. Type END from the Type Record Selection List or select another option to perform the update.
5. From the Type Record General Dataset Attributes screen, define or update the file attributes for the Type record. See [“INSERT TYPE and UPDATE TYPE Commands” on page 235](#) for a description of the fields on this screen you can update.

node.name	Type Record General Dataset Attributes	hh:mm
CMD ==>		
(DISP=)	Initial file status ==> ____ (SHR,NEW,OLD,MOD,RPL)	
	Normal step termination ==> _____ (KEEP,CATLG,DELETE)	
	Abend step termination ==> _____ (KEEP,CATLG,DELETE)	
(DCB=)	DSORG ==> ____ LRECL ==> ____ BLKSIZE ==> ____ RECFM ==> ____	
(SPACE=)	Allocation type ==> _____ (CYL,TRK,Average blksize)	
	Primary extent ==> _____ (Required with CYL,TRK,BLK)	
	Secondary extent ==> _____ (Optional)	
	Directory storage ==> _____ (Optional)	
	RLSE,CONTIG,ROUND ==> (_ , _ , _) ('Y'-Yes, 'N'-No)	
(UNIT=)	Unitname ==> _____	
(VOL=SER=)	Volume serial ==> _____	

Type the END command to return to the Type Record Selection List.

6. The Type Record IOEXIT Parameters screen (a TSO edit screen) enables you to define or update the IOEXIT parameters associated with the Type record. Refer to the IOEXIT parameter description for valid values.

```
node.name                                Type Record IOEXIT Parameters                                hh:mm
Cmd ==>                                Scroll ==> PAGE

-----
IOEXIT Specifications (exitname,{parameter,parameter,...})
-----
***** ***** Top of Data *****
RIOEX01,(C'MAXLEN',X'0120')

***** ***** Bottom of Data *****
```

After defining or updating the IOEXIT specifications, type the END command to return to the Type Record Selection List.

7. The Type Record SMS/VSAM Parameters screen enables you to define or update parameters related to SMS controlled data sets and VSAM files.

```
CD.ART                                Type Record SMS/VSAM Attributes                                15:14
CMD ==>

Operation    ==> UPDATE
Type Key     ==> EAV

(SMS=)  Data Class      ==> -----
        Management Class ==> -----
        Storage Class   ==> -----

(Other=)  EATTR          ==> OPT      (OPT or NO)
          DSNTYPE        ==> ----- (PDS, LIBRARY, LARGE, BASIC, EXTREQ,
                                         EXTPREF)

(MODEL=)  Like Data Set Name ==> -----
          Security Profile  ==> -----
          Generic Profile   ==> --- (YES or NO)

(SPACE=)  Average Record Value ==> _ (U, K or M)

(VSAM=)   Organization ==> -- (ES - ESDS, KS - KSDS, RR - RRDS, LS - LDS)
          Key Offset    ==> ---- (0 - 32760)
          Key Length     ==> --- (1 - 255)
```

After defining or updating the SMS/VSAM attributes, type the END command to return to the Type Record Selection List.

DELETE TYPE Command

Use the DELETE TYPE command to delete a Type record from the Type file. Use the following format and parameters. The required parameters and keywords are in bold print.

Label	Command	Parameters
(optional)	DELeTe TYPE	WHERE (TYPEKEY = typekey (list))

WHERE is the required parameter for the DELETE TYPE command. No optional parameters exist for this command.

Parameter	Description
WHERE (TYPEKEY = typekey (list))	<p>This parameter specifies which records in the Type file to delete. You can specify one Type key or a list of Type keys.</p> <p>typekey specifies the name associated with the record being deleted. The Type key is a 1–8 character alphanumeric string, with the first character alphabetic.</p> <p>list specifies multiple Type keys. Specify a list of keys by enclosing them in parentheses.</p>

Deleting Types Files through the Batch Interface

About this task

To use the DELETE TYPE command from the batch interface, follow this procedure.

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

For example, the following commands delete the records under the Type keys MYALLOC, NEWALLOC, and RPLALLOC from the Type file.

```

/* DELETES A SINGLE TYPE RECORD                               */
DELETE TYPE WHERE (TYPEKEY=MYALLOC)                           */
/* DELETES MULTIPLE TYPE RECORDS                               */
DELETE TYPE WHERE (TYPEKEY=(NEWALLOC RPLALLOC))               */

```

Deleting Type Files through the IUI

About this task

To issue the DELETE TYPE command from the IBM Connect:Direct IUI, follow this procedure.

Procedure

1. Select DT from the Administrative Options Menu.
2. Type in the names of the TYPE KEYs that you want to delete.
3. Press ENTER.

A list of the deleted records is displayed. If the delete is unsuccessful, a list of the records not deleted is displayed.

4. Press PF3/END to return to the Delete Data Set Type Defaults screen.
5. Use the Delete Data Set Type Defaults message display to verify a successful delete request.

SELECT TYPE Command

The SELECT TYPE command enables you to examine a record in the Type file. You can specify the search criteria and the form in which the information is presented.

The SELECT TYPE command uses the following format and parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SELECT TYPE	WHERE (
		TYPEKEY = typekey generic (list)
)
		PRint <u>TABLE</u>

WHERE is the required parameter for the SELECT TYPE command.

Parameter	Description
WHERE (TYPEKEY = typekey generic (list))	<p>This parameter specifies which records in the Type file to select.</p> <p>TYPEKEY = typekey generic (list) specifies the key or list of keys of the records to select.</p> <p>typekey specifies the name associated with the record selected. You created the typekey name when originally adding the entry to the Type file. The typekey is a 1–8 character alphanumeric string, with the first character alphabetic.</p> <p>generic specifies generic selection of type keys. To specify type keys generically, type a 1–7 character alphanumeric string, with the first character alphabetic, plus an asterisk (*). For instance, if your network includes the type keys SENDDAY, SENDMO, and SENDWK, a specification of SEND* provides information about those keys.</p> <p>list specifies multiple type keys. A list of keys is specified by enclosing them in parentheses.</p>

The optional parameter is described below.

Parameter	Description
PRInt TABLE	<p>Parameters specify the method of display for the output of the select.</p> <p>PRint specifies that the output to a printer in tabular format. Output is routed to the destination specified in the PRINT keyword of the SIGNON command.</p> <p>TABLE specifies that the output is stored in a temporary file in tabular format and is displayed upon successful completion of the command. This parameter is the default.</p>

Example of Selecting a Type Record

The following command selects a record in the TYPE file.

```
SELREC    SELECT TYPE    WHERE (TYPEKEY=DF*)
```

The output follows.

```
=====
                        SELECT TYPE DEFAULTS
=====
Type Key   => DF           Date Created => mm/dd/yyyy

DISP=(RPL,CATLG,DELETE)
DCB=(DSORG=PS,LRECL=255,BLKSIZE=2554,RECFM=VB)
SPACE=(TRK,(10,10))
UNIT=SYSDA
```

Selecting Type Records through the Batch Interface

About this task

To use the SELECT TYPE command from the Batch Interface, follow this procedure.

Procedure

1. Place your command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Use the output to verify your results.

Selecting Type Records through the IUI

About this task

To issue the SELECT TYPE command through the IBM Connect:Direct IUI, follow this procedure.

Procedure

1. Access the Select Data Set Type Defaults screen by selecting option ST from the Administrative Options Menu.
2. Type the typekey for each member you want to select. Refer to the description of parameters for the “SELECT TYPE Command” on [page 244](#), or press PF1 to view the Help.
3. Indicate the output destination in the Output Destination field.
4. Provide the requested information and press ENTER.

An output destination of DIS produces a display similar to the following figure.

```
BROWSE -- XXXXXXXX.XXXXXXX.XXXXX.XXXXX.XXXXX ---- LINE 00000000 COL 001 080
COMMAND ==>                                SCROLL ==> CSR
***** TOP OF DATA *****
=====
                        SELECT TYPE DEFAULTS
=====
Type Key => BINARY      Date Created => mm/dd/yyyy

DISP=(RPL,CATLG,DELETE)
DCB=(DSORG=PS,BLKSIZE=6144,RECFM=U)
SPACE=(TRK,(10,10))

-----
Type Key => DF          Date Created => mm/dd/yyyy

DISP=(RPL,CATLG,DELETE)
DCB=(DSORG=PS,LRECL=255,BLKSIZE=2554,RECFM=VB)
SPACE=(TRK,(10,10))
UNIT=SYSDA

-----
```

Maintaining the Network Map

The network map (also known as the NETMAP) identifies the local IBM Connect:Direct node and the nodes it communicates with. It consists of one local node entry and one or more adjacent node entries that identify the communication name and protocol of each IBM Connect:Direct node.

The network map source is generated during installation. The network map load utility, DGADNTLD, uses this source to create a network map (a VSAM file).

See the *IBM Connect:Direct for z/OS Configuration Guide* for rules governing the network map for cross-domain VTAM definitions.

Local Node Entry

The local node entry specifies the logical name of the local IBM Connect:Direct and its associated communications name. The local node entry also contains the name of the Transmission Control Queue (TCQ) and the SUPERUSR ID password, if specified. The syntax is displayed in the following figure.

```
LOCAL.NODE=(
    (node name,communications name,,superuser password)
    TCQ=(tcxdsn, tcqdsn)
    CONTACT.NAME="name"
    CONTACT.PHONE="phone information"
    DESCRIPTION="description information"
)
```

Local Node Positional Parameters

The network map local node entry contains the following positional parameters.

Parameter	Description
node name	<p>The first positional parameter is the 1–16 alphanumeric character node name. It specifies the logical name of the local IBM Connect:Direct DTF.</p> <p>Note: Connect:Direct for z/OS does not accept the following characters for the node name:</p> <ul style="list-style-type: none">• < less than• ¬ logical not• , comma• > greater than• = equal sign• / forward slash• * asterisk• \ backward slash• ' single quote• " double quote• (open parenthesis•) close parenthesis <p>Important: Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$. _ - to ensure that the entries can be properly managed by Control Center, Connect:Direct Browser User Interface, or Sterling Connect:Direct Application Interface for Java (AIJ) programs.</p>
communications name (for SNA only)	<p>The second positional parameter is the 1–8 character communications name. It specifies the VTAM APPLID that IBM Connect:Direct uses to communicate over the network.</p> <p>If the node uses only TCP/IP communications (SNA=NO is specified in the initialization parameters), define this parameter as NO-VTAM. Refer to <i>Initializing IBM Connect:Direct without SNA Support</i> in the <i>IBM Connect:Direct for z/OS Configuration Guide</i> for more information.</p>
null	The third positional parameter is not used.

Parameter	Description
superuser password	<p>The fourth positional parameter is the 1–8 character SUPERUSR ID password. The initial value for this parameter is specified during installation.</p> <p>The SUPERUSR ID is provided to bypass your usual security system at signon. This bypass can be necessary if IBM Connect:Direct is configured improperly, resulting in the inability to signon. SUPERUSR still goes through usual data set verification done by the Stage 2 security exit.</p> <p>Note: The default SUPERUSR ID is SUPERUSR and it is recommended that you change it.</p>

Local Node Keyword Parameters

The network map local node entry contains the following keyword parameters:

Parameter	Description
TCQ= (tcxdsn,tcqdsn)	<p>The TCQ is a VSAM relative record data set (RRDS) that holds all Processes submitted to IBM Connect:Direct.</p> <p>The TCQ parameter specifies the two files that comprise the Transmission Control Queue (TCQ). This parameter is required.</p> <p>Note: Use the correct order (tcxdsn,tcqdsn) when you specify the two files.</p> <p>tcxdsn identifies the data set name of the TCQ index (TCX).</p> <p>tcqdsn identifies the data set name of the TCQ.</p>
CONTACT.NAME="name" CONTACT.PHONE="phone information" DESCRIPTION="description information"	<p>These are free-form text parameters, which provide additional general information about an adjacent node entry. The CONTACT.NAME and CONTACT.PHONE parameters are limited to a maximum of 40 characters. The DESCRIPTION parameter is limited to a maximum of 255 characters.</p>

When to Define the Local Node as the Adjacent Node

You must also define the local node as an adjacent node to:

- Specify the VTAM application IDs that are used for IUI and batch sessions. The APPLIDS match those defined during installation preparation as described in the *IBM Connect:Direct for z/OS Configuration Guide*.
- Provide the ability to run Processes where the local node is both the initiating and target node (PNODE and SNODE). The communications name matches the APPLID defined during installation preparation.
- Supply the communication address for a TCP API, if it is not specified during the user signon.

If you are using TCP/IP only and SNA is set to NO, refer to *Initializing IBM Connect:Direct without SNA Support* in the *IBM Connect:Direct for z/OS Configuration Guide*.

Adjacent Node Entry

Adjacent node entries specify network nodes that the local IBM Connect:Direct can communicate with. Each entry specifies a locally used IBM Connect:Direct name, its associated network communications

name, and session control parameters for these nodes. The syntax in the following figure is for a typical adjacent node entry.

```

ADJACENT.NODE=(
    (nodename,
        communications name | channel-range-start-addr,
        remote library name | IP address or Alias | addr-count,
        session type,
        security node type,
        data direction restriction)
    PARSESS=(max,default)
    ZFBA = (addr1 ,
addrn)
    COMPRESS.EXT=ALLOW | DISALLOW | FORCE
    COMPRESS.STD=ALLOW | DISALLOW | FORCE
    COMPRESS.STD.PRIMECHAR=C'x'| X'xx'
    SOURCEIP=IP address
    FASP=NO|SSP
    FASP.BANDWIDTH=nnn | nM, nG
    FASP.FILESIZE.THRESHOLD=nnn | nM | nG
    FASP.POLICY=FAIR | HIGH | LOW | FIXED
    SESS.SNODE.MAX = 255
    LDNS=hostname
    ENVIRONMENT=operating environment
    LOGMODE=logmode entry name
    APPLIDS=(vtam applid1 [,vtam applid2,...] )
    BATCHAPPLIDS=(batch.applid1 [,batch.applid2,...] )
    TSO.APPLIDS=(tso.applid1 [,tso.applid2,...] )
    INTERACTIVE.APPLIDS=(interactive.applid1 [,interactive.applid2,...] ) -
    CICS.APPLIDS=(cics.applid1 [,cics.applid2,...] )
    NETID=networkid | CTCA server name
    PNODE.LUS=(luname1 [,luname2,...] )
    SNODE.LUS=(luname1 [,luname2,...] )
    USE.SERVER.NODE = NO | YES
    TCPAPI= (port number, IP address)
    CRC= (OFF | ON | DEFAULT)
    PLEXCLASS= (* | plexclass, * | plexclass)
    BUFFER.SIZE= 3072-2097152|3K-2M
    ALTErnate.COMMinfo = (ALTErnate.RESTART=YES | NO,
        ALTErnate.DIRection=BALANCE | TOP,
        (ALTErnate.ADDRes= | ALTErnate.NODEDEF=, ALTErnate.PORT=,
        SOURCEIP=IP address,
        ALTErnate.TYPE=SNA | TCP | UDT33 | LU62
        ALTErnate.LOGmode=logmode entry name,
        ALTErnate.USE.OUTbound=YES | NO)
        )
    CDFTP.PLUGIN="name or location of the plugin"
    CDFTP.TEMPFILE="fully qualified file path and name"
    CONTACT.NAME="name"
    CONTACT.PHONE="phone information"
    DESCRIPTION="description information")

```

In an environment when one IBM Connect:Direct system (either a IBM Connect:Direct/Plex environment or IBM Connect:Direct/Stand-alone Server) communicates with a IBM Connect:Direct/Plex environment, add the following statement to each ADJACENT.NODE entry that defines the other IBM Connect:Direct/Manager.

```

ENVIRONMENT=ZOS

```

Note: When a Connect:Direct for z/OS system communicates with another IBM Connect:Direct system in a IBM Connect:Direct/Plex environment and ENVIRONMENT=ZOS|OS390 is not specified, Process redirection does not function correctly.

See [“TCP/IP Addressing”](#) on page 263 for network map entry requirements for TCP/IP nodes.

See [Channel-to-Channel Support](#) for a discussion of channel-to-channel support.

Positional Parameters for Adjacent Node Entries

The network map adjacent node entry contains positional and keyword parameters. Following are the positional parameters for the network map adjacent node entry:

Parameter	Description
nodename	<p>The first positional parameter is the 1–16 alphanumeric character node name. This name represents the partner IBM Connect:Direct and is used in communications with the local IBM Connect:Direct. This parameter is required. The node name is always changed to upper case in the network map, regardless of the remote platform.</p> <p>Note: Connect:Direct for z/OS does not accept the following characters for the node name:</p> <ul style="list-style-type: none"> • < less than • ¬ logical not • , comma • > greater than • = equal sign • / forward slash • * asterisk • \ backward slash • ' single quote • " double quote • (open parenthesis •) close parenthesis <p>Important: Characters used in Netmap Node Names (or Secure+ Node Names or Secure+ Alias Names) should be restricted to A-Z, a-z, 0-9 and @ # \$. _ - to ensure that the entries can be properly managed by Control Center, Connect:Direct Browser User Interface, or Sterling Connect:Direct Application Interface for Java (AIJ) programs.</p>
communications name channel- range-start-addr	<p>The second positional parameter is the 1–8 alphanumeric character communications name. It specifies the network name of the partner IBM Connect:Direct. It can be an SNA VTAM APPLID or a TCP/IP port number. This parameter is optional.</p> <p>For SNA, this field must contain the VTAM APPLID of the remote IBM Connect:Direct node that the local DTF uses for DTF-to-DTF communications with the remote node.</p> <p>This name is the same name that is defined for the communications name in the network map of the remote IBM Connect:Direct node.</p> <p>For OpenVMS and HP NonStop, use the PNODE.LUS and SNODE.LUS parameters and leave this field blank.</p> <p>For TCP/IP, this field contains the TCP/IP port number of the remote partner IBM Connect:Direct. You do not need to use this field if the partner IBM Connect:Direct is initialized using the default TCP/IP port number. This port number does not change the port number for the host IBM Connect:Direct that is defined at initialization. See “Multiple Port TCP/IP Listen” on page 265 for more information about the TCP/IP port number.</p> <p>For CTCA, this field contains the CCUU of the first CTCA address used by this node.</p>

Parameter	Description
remote library name IP address or Alias addr-count	<p>The third positional parameter is for the Host ID (DNS), or IP Address, or Library name. This is for I5/OS or TCP/IP nodes only.</p> <p>For I5/OS nodes, this parameter specifies the name of the library where the IBM® Connect:Direct for I5/OS program SMMAIN resides for the partner IBM Connect:Direct.</p> <p>For TCP/IP nodes, this parameter specifies the IPv4 address (1-15 dotted decimal characters), DNS alias name (1-16 alphanumeric characters) or the IPv6 address (1-39 colon separated hexadecimal characters) to establish the TCP/IP session.</p> <p>For DNS alias names greater than 16 characters, leave this field blank and use the LDNS parameter.</p> <p>For CTCA, this parameter specifies the number of CTCA addresses used by this node. Specify this with an even number value (a minimum of 2).</p>
session type	<p>The fourth positional parameter is the session type. It specifies the type of session communications protocol to use for communications with this adjacent node. This parameter is required for i5/OS adjacent nodes and any node using a protocol other than LU0. Valid values are:</p> <p>SNA (for LU0 protocol)</p> <p>SNUF (for LU0 protocol for the i5/OS)</p> <p>LU62 (for LU6.2 protocol)</p> <p>TCP (for TCP/IP protocol)</p> <p>UDT33 (for UDT protocol)</p> <p>CTCA (for Channel to Channel connections)</p> <p>Note: SNA is the default if this positional parameter is left blank.</p>
security node type	<p>The fifth positional parameter is the security node type. It classifies the node as an internal or external node. Specify this parameter for Trusted Node security. It is optional if you do not use Trusted Node security.</p> <p>For further information on Trusted Node security refer to Trusted Node Security .</p> <p>EXTERNAL EXT specifies an external security classification for this node.</p> <p>INTERNAL INT specifies an internal security classification for this node. This is the default.</p>

Parameter	Description
data direction restriction	<p>The sixth positional parameter is the data direction restriction. It identifies the copy initiation abilities of this adjacent node with the local node. For further information on data direction restriction, refer to Data Direction Restriction. This parameter is optional. Valid data direction values are:</p> <p>RECEIVE RECV indicates that when the adjacent node initiates a transfer, it is only allowed to receive data from this node. It is never allowed to send data to this node.</p> <p>Note: For CTCA, RECEIVE RECV indicates that the first address in channel-range-start-addr is used for inbound traffic; the next address is used for outbound traffic. For two IBM Connect:Direct systems to communicate through CTCA, one adjacent node must specify SEND and the other must specify RECEIVE RECV.</p> <p>SEND indicates that when the adjacent node initiates a transfer, it is only allowed to send data to this node. It is never allowed to receive data from this node.</p> <p>Note: For CTCA, SEND specifies that the first address specified in channel-range-start-addr is is used for outbound traffic; the next address is used for inbound traffic.</p> <p>For two IBM Connect:Direct systems to communicate through CTCA, one adjacent node must specify SEND and the other must specify RECEIVE RECV.</p> <p>BOTH indicates that when the adjacent node initiates a transfer, it is allowed to both send and receive data from this node. This value is the default.</p> <p>NONE indicates that when the adjacent node initiates a transfer, it is neither allowed to send or receive data from this node.</p>

Keyword Parameters for Adjacent Node Entries

The network map adjacent node entry contains keyword parameters. The following are the positional parameters for the network map adjacent node entry:

Parameter	Description
PARSESS=(max,default)	<p>This is an optional parameter that uses two CLASS values for each session between the PNODE and SNODE to define the total number of concurrent sessions or classes defined by the NETMAP entry. Each session is assigned a range of session class values that are obtained from the ADJACENT.NODE entry of the SNODE, which is specified by the PROCESS statement keyword, SNODE. The PARSESS parameter also defines the default Process CLASS value used by the Process when the PROCESS statement does not specify the CLASS keyword.</p> <p>max specifies the maximum number of simultaneous DTF-to-DTF sessions that the local IBM Connect:Direct node can initiate with this adjacent node. The range of this subparameter is 2–255. Each session is represented by a corresponding class value. This class value determines the execution eligibility of a Process. Leave this field blank if parallel sessions are not available.</p> <p>default specifies the class assigned to a Process if one is not specified on the Process statement or when the Process is submitted. The range of this subparameter is 1-the value coded for parallel sessions. If you do not code this parameter, the node is not parallel session-capable, and the max and default values are set to 1. This parameter is required if you do PNODE=SNODE processing.</p> <p>Note: If you do not specifically code the PARSESS parameter on an adjacent node definition, a default of PARSESS=(1,0) is set by the system, which allows only a single session. The default class 0 means no CLASS which shows up as a CLASS of NONE in an executing Process. You cannot code the default PARSESS=(1,0) on an ADJACENT.NODE definition—the minimum value that can be coded is PARSESS=(2,1).</p> <p>The following formula is a quick way to determine the number of parallel sessions available when class is not defined for a Process (the default class is used): Default Parsess value = (max – default) +1</p> <p>For CTCA connections, this value is automatically set to 1/2 of the addr-count. Do not try to reset this parameter for CTCA connections.</p> <p>For best results when using Independent LU6.2, code PARSESS the same for both the local and remote nodes.</p> <p>Selecting a Class</p> <p>IBM Connect:Direct selects a class in which a Process is to run by starting with the default class, or the coded class, and proceeds upward in class values until an available class slot is found, or until all possible class values are tested.</p> <p>For more information on Process class, see the Connect:Direct Process Language help.</p> <p>For more information about parallel sessions, refer to <i>Building, Modifying, and Submitting Processes Queueing</i> in the <i>IBM Connect:Direct for z/OS User Guide</i>.</p>

Parameter	Description
ZFBA=(addr,addm)	<p>The ZFBA parameter has no default value; if it is not specified, the node will be ineligible for ZFBA transfers.</p> <p>The range is specified as a pair of 4-hex-digit device addresses separated by a comma or space; the second value must be greater than the first value. Adjacent.node example defines 4 devices, 2000-2003 as the zFBA device range.</p>
COMPRESS.EXT=ALLOW DISALLOW FORCE	<p>This is an optional parameter that can be used to control how extended compression will be used on a node.</p> <p>ALLOW specifies that extended compression may be used. In this case, extended compression may be specified within the Process or by the other node.</p> <p>DISALLOW specifies that extended compression cannot be used.</p> <p>FORCE specifies that extended compression is required.</p>
COMPRESS.STD=ALLOW DISALLOW FORCE	<p>This is an optional parameter that can be used to control how standard compression will be used on a node.</p> <p>ALLOW specifies that standard compression may be used. Standard compression may be specified within the Process or by the other node.</p> <p>DISALLOW specifies that standard compression cannot be used.</p> <p>FORCE specifies that standard compression is required.</p>
COMPRESS.STD.PRIMECHAR=C'x' X'xx'	<p>This is an optional parameter used for text data or single-character repetitive data to specify the primary compression character. It is specified as either two hex digits preceded by "X" or a single character preceded by "C", for example, COMPRESS.STD.PRIMECHAR=C'\$', or this can also be expressed as COMPRESS.STD.PRIMECHAR=X'4B'</p> <p>Note: If COMPRESS.STD is specified and COMPRESS.STD.PRIMECHAR is not specified, then the Process default for COMPRESS.STD.PRIMECHAR, which is X'40' (an EBCDIC space), will be in effect.</p>
SOURCEIP = IP address	<p>If an IPv6 or IPv4 address is required for the network map, this parameter defines the source IP address that is bound to during outbound connection requests. If the SOURCEIP parameter is not specified, the local or default address is bound. The destination address is obtained from the network map. The source port is obtained as assigned by TCP/IP, or through the TCP.SRC.PORTS table. If the address from the SOURCEIP cannot be bound during Process execution, the Process is placed in a TI RE state (Timer Queue Retry Status).</p> <p>Note: If the network map adjacent node contains a IPv6 address, but IBM Connect:Direct is not enabled to support it, the submit of a Process for that network map entry fails. For information about enabling IBM Connect:Direct for IPv6 support, refer to TCP/IP Addressing.</p>
SESS.SNODE.MAX = nnn	<p>This is an optional parameter that can be used to control the number of concurrent sessions that an adjacent node can initiate as the SNODE. The range for SESS.SNODE.MAX is 1 to 255. The default value is 255. In a IBM Connect:Direct/Plex environment, this parameter controls the number of sessions that an adjacent node can initiate with each IBM Connect:Direct/Plex server.</p> <p>Note: For use with TCP/IP and LU6.2 only.</p>

Parameter	Description
LDNS=hostname	<p>LDNS is an optional parameter that specifies the host name in the network map adjacent node entry.</p> <p>If you use this parameter, you can leave the third positional parameter (IP address or alias) blank. You must also specify TCP as the fourth positional parameter (session type).</p> <p>hostname specifies the host name for this node. The host name length is from 1–256 characters.</p> <p>To see an example of a long DNS record, see <i>The Network Map</i> in the <i>IBM Connect:Direct for z/OS User Guide</i>.</p>
ENVIRONMENT= operating environment	<p>The ENVIRONMENT parameter identifies the adjacent node operating system environment. This parameter is required when the session type positional parameter specifies LU6.2 protocol for i5/OS systems and when one IBM Connect:Direct system (either a IBM Connect:Direct/Plex environment or IBM Connect:Direct/Stand-alone Server) communicates with a IBM Connect:Direct/Plex environment. Other protocols can use it for documentation purposes.</p> <p>Valid values are: GIS, HPNONSTOP, I5OS, LINUX, OPENVMS, OS400, SELECT, VOS, STRATUS, UNIX, VM, VMS, WINDOWS, and ZOS.</p> <p>Note: When an OS/390 or z/OS IBM Connect:Direct system communicates with another IBM Connect:Direct system in a IBM Connect:Direct/Plex environment and ENVIRONMENT=ZOS OS390 is not specified, Process redirection does not function correctly.</p>
LOGMODE=logmode entry name	<p>Identifies the VTAM logmode entry that defines the communication protocol for this node.</p> <p>This parameter is only required for LU6.2 nodes.</p> <p>It is optional for LU0 connections. If you specify this parameter for LU0 connections, the RUSIZE defined within this LOGMODE is used for any transfer with this node. For a host-to-host transfer, the LOGMODE entry in the VTAM MODETAB of the SNODE determines the RUSIZE. For a host-to-PC transfer, the LOGMODE entry in the host VTAM MODETAB is used.</p> <p>This parameter is not valid for TCP/IP nodes or CTCA connections.</p> <p>Refer to <i>Sample VTAM Definitions</i> in the <i>IBM Connect:Direct for z/OS Configuration Guide</i> for information about VTAM definitions.</p>

Parameter	Description
<p>APPLIDS=(vtam applid1 [,vtam applid2,...])</p> <p>or</p> <p>BATCHAPPLIDS=(batch.applid1 [,batch.applid2,...])</p> <p>TSO.APPLIDS=(tso.applid1 [,tso.applid2,...])</p> <p>INTERACTIVE.APPLIDS=(interactive.applid1 [,interactive.applid2,...])</p> <p>CICS.APPLIDS=(cics.applid1 [,cics.applid2,...])</p>	<p>The APPLIDS parameter specifies the VTAM APPLIDs that establish a session between the local IBM Connect:Direct SNA LU0 API and the IBM Connect:Direct DTF. APPLIDs are defined on the local nodes adjacent node record for LU0 sessions to this DTF and are defined on the remote nodes adjacent node record for SNA LU0 multi-session signon to those remote DTFs. This parameter is valid only for z/OS and VM/ESA nodes, and the actual VTAM APPLIDs must be defined and active on the VTAM system where the SNA LU0 IUI / API signon is taking place.</p> <p>To isolate or separate the VTAM pools to specific user interfaces, you can use the following parameters to define special-purpose APPLIDS or just use the APPLIDS parameter itself:</p> <ul style="list-style-type: none"> • BATCH.APPLIDS with the batch user interface • INTERACTIVE or TSO APPLIDS with the ISPF user interface • CICS.APPLIDS with the CICS user interface
<p>NETID=networkid servername</p>	<p>The NETID parameter specifies the 1–8 character network ID for this node. When a Process starts, the network ID provided at the session start is verified against the network ID in the network map for the adjacent node. If they do not match, the Process is terminated with an error.</p> <p>For multiple session signons, the network ID of the node signing on is verified against the network map network ID of the node being signed on to. If they do not match, the signon fails.</p> <p>If this keyword is not coded or the IBM Connect:Direct initialization parameter NETMAP.CHECK is set to NO, the network ID is not checked at Process start or multiple session signon.</p> <p>For a CTCA connection in a IBM Connect:Direct/Plex environment, this parameter specifies the 1–8 character name of the IBM Connect:Direct/Server.</p>
<p>PNODE.LUS=(luname1 [,luname2,...])</p>	<p>The PNODE.LUS parameter specifies the logical units used by a remote node to initiate a session with this local node. Do not specify the communications name when you use this parameter.</p> <p>Applies to OpenVMS nodes. It is not valid for CTCA connections.</p>

Parameter	Description
SNODE.LUS=(luname1 [,luname2,...]))	<p>For OpenVMS, the SNODE.LUS parameter specifies the logical unit names used by the local node to initiate a session with this remote node. For all other platforms, it specifies the logical units used for all communications with the remote node.</p> <p>Communications to nodes that cannot handle parallel sessions can require a pool of logical units. If an adjacent node is defined in its host environment to use more than one logical unit for communications, then each of the logical unit names that can communicate with the local node must be defined to the local node on the corresponding adjacent node network map entry.</p> <p>HP NonStop and i5/OS adjacent node entries use the SNODE.LUS keyword only to define the LU pool.</p> <p>OpenVMS nodes assign the logical units in the pool as either ACTIVE, session initiating, PASSIVE, or listening for session requests. This distinction in function is defined to the z/OS node by specifying the ACTIVE logical units with the PNODE.LUS keyword and the PASSIVE logical units with the SNODE.LUS keyword.</p> <p>This parameter is not valid for CTCA connections.</p>
PNODE.LAST.USED=(mm.dd.yyyy,h h:mm:ss)	<p>The date and time the Adjacent Node was last used as the PNODE that is, the Local Node was the SNODE. This parameter is only unloaded by program DGADNTLD if the usage (US) record exists for the Adjacent Node, and the PNODE last used fields in the US record are not binary zeroes.</p>
SNODE.LAST.USED=(mm.dd.yyyy,h h:mm:ss)	<p>The date and time the Adjacent Node was last used as the SNODE that is, the Local Node was the PNODE. This parameter is only unloaded by program DGADNTLD if the usage (US) record exists for the Adjacent Node, and the SNODE last used fields in the US record are not binary zeroes.</p>
USE.SERVER.NODE = YES <u>NO</u>	<p>Setting this parameter to YES tells a IBM Connect:Direct/Server to use its CDPLEX.SERVER.NODE initialization parameter as identification when communicating with this adjacent node. If this parameter is set to NO, the IBM Connect:Direct/Server identifies itself to this adjacent node as the same local node as all other members of the IBM Connect:Direct/Plex environment. (See CDPLEX.SERVER.NODE = node name for more information.)</p> <p>This parameter is required if a remote IBM Connect:Direct system, using NETMAP checking, communicates with more than one IBM Connect:Direct/Server in a IBM Connect:Direct/Plex environment.</p> <p>This parameter is ignored in a IBM Connect:Direct/Stand-alone Server.</p>

Parameter	Description
TCPAPI=(port number, IP address)	<p>This parameter defines the adjacent node communication address used by an external API that uses TCP to communicate with the node. This value must be the same as the TCP.API.PORTNUM initialization parameter of the node that you are communicating with.</p> <p>If the adjacent node is an SNA node, include both the port number and IP address.</p> <p>If the adjacent node is a TCP node, you must supply the port number, but do not need to supply the IP address. If you do not supply the IP address, you must define the IP address in the Adjacent node record. See API Signons for more information.</p>
CRC =(OFF ON DEFAULT)	<p>Determines if you will perform CRC checking for any TCP/IP Process sending to this node. If overrides are allowed, this parameter enables you to override the CRC setting in the initialization parameters for this node.</p> <p>Note: Although DEFAULT is an acceptable value, it is the equivalent of not specifying the network map parameter at all and would normally only be seen in the output of the network map file unload when no CRC specification had been supplied previously.</p>
PLEXCLASS= (* plexclass, * plexclass)	<p>This parameter specifies a default PLEXCLASS for the PNODE (the first parameter) and SNODE (the second parameter). This IBM Connect:Direct/Server checks this PLEXCLASS to determine if it can run the Process. (See the initialization parameter, “CDPLEX.PLEXCLASSES = (*,plexclass,...,plexclass)” on page 499, for more information.)</p> <p>Each PLEXCLASS name is 1–8 characters long. An asterisk (*) is also an acceptable entry, which indicates that the IBM Connect:Direct/Server supports any Process that does not specify a PLEXCLASS, or specifies a PLEXCLASS of ‘*’.</p>
BUFFER.SIZE=V2.buffer override	<p>V2.buffer override defines the buffer size for the adjacent node. It is only valid for TCP and CTCA session types. Use it to dynamically override the V2.BUFSIZE value of the local node during TCP buffer size negotiation. This parameter is generally coded to reduce the V2.BUFSIZE value so that IBM Connect:Direct provides less data on each TCP send. Valid values are 3072–2097152 3K–2M.</p>

Parameter	Description
ALternate.COMMInfo= (ALternate.RESTART=YES <u>NO</u> ALternate.DIRction= <u>BALANCE</u> TOP, (ALternate.ADDress= ALternate.NODEDEF=, ALternate.PORT=, SOURCEIP=, . . .	<p>This parameter enables you to specify multiple remote node addresses for NETMAP checking.'</p> <p>Note: The number of ALT.ADDR parameters can not exceed the maximum length of the Netmap Alternate Address Record.</p> <p>ALternate.RESTART indicates if an alternate communications path is considered when a Process must restart.</p> <p>Note: Do not specify ALT.NODEDEF if you use ALT.RESTART. All alternate communications paths must use ALT.ADDR and the same protocol (all Version 1 or Version 2). For example, if you use ALT.TYPE=SNA, specify all alternate addresses as ALT.TYPE=SNA (Version 1 protocol). Since LU62 and TCP are Version 2 protocols, you can use LU62 or TCP when specifying alternate addresses.</p> <p>ALternate.DIRction specifies the direction of the communications paths are selected as the primary path. BALANCE (default) indicates a balanced approach; that is, all current sessions are scanned for this same adjacent node and the least used path is selected as the primary path for the Process. The list is processed forward from there.</p> <p>Note: BALANCE does not guarantee a truly balancing between all alternate paths. It is simply used to select the first path attempted and the Connect:Direct Process may not necessarily execute on that path. TOP indicates the paths are processed from the top.</p> <p>ALternate.ADDress specifies either a TCP/IP address or an SNA address as appropriate.</p> <p>If ALT.TYPE=SNA or ALT.TYPE=LU62, ALT.ADDR must be a 1–8 character APPLID. If ALT.TYPE=TCP, ALT.ADDR must be a 15-character IPv4 or 39-character IPv6 TCP/IP address or a 1–256 character LDNS name. This subparameter is required if you do not specify ALternate.NODEDEF.</p> <p>ALternate.NODEDEF specifies an alternate node definition to use for NETMAP checking. This subparameter references another entry in the NETMAP. This subparameter is required if you do not specify ALternate.ADDress.</p> <p>ALternate.PORT specifies the alternate address port number if the alternate address is TCP/IP. This field is not required. This subparameter defaults to the port of the adjacent node record. If not specified in the adjacent node record, the default is 1364. You can only use this subparameter if ALT.TYPE=TCP.</p> <p>Note: If ALternate.DIRction=BALANCE is specified, the value specified for ALternate.PORT is not used.</p> <p>SourceIP, which has no short form, specifies an alternate IPv6 or IPv4 address that is bound to during outbound connection requests. The source port is obtained as assigned by TCP/IP, or through the TCP.SRC.PORTS table. If the address from the SOURCEIP cannot be bound during Process execution, the Process is placed in WA or WT state.</p> <p>If SourceIP is specified for this ADJACENT NODE but not in the ALT.COMM entry, the value specified for the ADJACENT NODE also applies to this ALT.COMM entry.</p> <p>Note: If the network map adjacent node contains a IPv6 address, but IBM Connect:Direct is not enabled to support it, the submit of a Process for that network map entry fails. For information about enabling Connect:Direct for IPv6 support, refer to TCP/IP Addressing.</p>

Parameter	Description
ALternate.COMMInfo= . . . ALternate.TYPE=SNA TCP LU62, ALternate.LOGmode= logmode entry name, ALternate.USE.OUTbound= <u>Yes</u> No) (Continued from previous page)	<p>ALternate.TYPE specifies the protocol used for the alternate address. This value defaults to that of the adjacent node record. Valid values are SNA, TCP, LU62. This subparameter is only used with ALternate.ADDress. This subparameter is required if you do not specify ALternate.NODEDEF.</p> <p>ALternate.LOGmode specifies an SNA logmode used when ALternate.TYPE=SNA or LU62. This parameter is required for LU62 if the adjacent node is not defined as LU62.</p> <p>ALternate.USE.OUTbound specifies whether the alternate communications path is used for outbound Processes, providing session failover for Processes sent to this adjacent node. Valid values are Yes (default) and No.</p> <p>For more information on this parameter, refer to Using Alternate Communications Paths.</p>
CDFTP.PLUGIN="fully qualified installation directory of the IBM Connect:Direct FTP+ Plug-in"	<p>IBM Connect:Direct FTP+ for z/OS gets its configuration information about remote systems it connects to from the Netmap.</p> <p>Configure the CDFTP.PLUGIN parameter in your Netmap to specify where the IBM Connect:Direct FTP+ Plug-in is installed on each remote system that IBM Connect:Direct FTP+ for z/OS connects to.</p> <p>This parameter is not required when the Netmap entry defines a z/OS system.</p>
CDFTP.TEMPFILE="fully qualified name of the IBM Connect:Direct FTP+ temporary file"	<p>IBM Connect:Direct FTP+ for z/OS gets its configuration information about remote systems it connects to from the Netmap.</p> <p>Configure the CDFTP.TEMPFILE parameter in your Netmap to specify the fully qualified name of the IBM Connect:Direct FTP+ temporary file for each remote system that IBM Connect:Direct FTP+ for z/OS connects to.</p> <p>The FTP+ Plug-in creates a temporary file to store the results of a directory command RUNTASK operation.</p> <p>Note: If no variables are used in the CDFTP.TEMPFILE, then IBM Connect:Direct FTP+ for z/OS appends the job userid and the ASID and CPUID to the CDFTP.TEMPFILE parameter as a further specification of uniqueness.</p> <p>Several variables are available for substitution in the CDFTP.TEMPFILE specification. These variables are:</p> <ul style="list-style-type: none"> • &userid;—The remote userid (lower case) • &USERID;—The remote userid (upper case) • &luserid;—The local userid padded to 8 characters with '\$' • &lcpuid;—The first 8 bytes of the local CPUID with "C" overlaying the first byte • &asid;—The 4 character Address Space ID prefixed with "AS" <p>For example, you can specify CDFTP.TEMPFILE= with a variable, such as "&userid;". IBM Connect:Direct FTP+ for z/OS inserts the remote userid in place of the "&userid;".</p>

Parameter	Description
CONTACT.NAME="name" CONTACT.PHONE="phone information" DESCRIPTION="description information"	These are free-form text parameters, which provide additional general information about an Adjacent Node entry. The CONTACT.NAME and CONTACT.PHONE parameters are limited to a maximum of 40 characters. The DESCRIPTION parameter is limited to a maximum of 255 characters.
FASP = (NO SSP , NO SSP)	The first positional parameter is used for when this IBM Connect:Direct z/OS Server is the PNODE and the second is for when this IBM Connect:Direct z/OS Server is the SNODE. The default of NO indicates that the FASP via SSP support is globally set to FASP=NO. IBM Connect:Direct Processes should not attempt to use the FASP transport unless overridden by the NETMAP or Process. SSP indicates that the IBM Connect:Direct Processes should request the FASP transport when in session with SSP unless override by the NETMAP or Process.
FASP.BANDWIDTH = nnn nK nM nG	This is an optional FASP configuration parameter that specifies how much bandwidth each transfer can use. The default is taken from the license and is negotiated with the remote to the smaller of the values. The value cannot exceed the value defined in the license and SSP will ensure the smaller is used. This value is represented as number of bits and 1K equals 1000.
FASP.FILESIZE.THRESHOLD = nnn nK nM nG	The default of 1G defines the threshold to limit the use of the FASP transport to specific file sizes. If the estimated file size is less than this threshold, IBM Connect:Direct for z/OS will proceed as FASP=NO for that file transfer. This value is represented as number of bytes and 1K equals 1024. The sending IBM Connect:Direct Server will apply the threshold using the PNODE's setting.
FASP.POLICY = FAIR FIXED HIGH LOW	This is an optional FASP configuration parameter that specifies the 'fairness' of each transfer. The default is FAIR.

Using Alternate Communications Paths

Use the ALT.COMM parameter to set up alternate communications paths. To see an example of using alternate communications paths in a IBM Connect:Direct/Plex environment, see [Strategies for Communicating with Non-Plex Servers](#).

Note the following:

- The protocol and number of alternate communications paths depend on the capability of the remote node.
- If you use alternate communications paths, the remote IBM Connect:Direct node must have the capability to support the ALT.COMM parameter and have a defined method of performing network map checking for these alternate paths.
- If you are using either the SSL or TLS protocol with Connect:Direct Secure Plus, all communications paths must be TCP.
- Since each ALT.COMM entry varies in size and each ALT.NODE and ALT.ADDR entry is also variable, the number of entries that may be added to a single ADJACENT.NODE varies. The limit will be determined by how many entries fit within a single Netmap record and it may vary from 20 to 30 entries, depending on parameters and values for each entry.

Alternate Communications Paths with Current Node as Primary Node

The adjacent node record defines a communications path. You can use the ALT.COMM parameter in the adjacent node record to define alternate communications paths.

When a Process is submitted, the primary communications path is selected based on the `alternate.direction` subparameter. If `alternate.direction=BAL`, all communications paths are scanned to find the least used path, including the path defined in the adjacent node record, and all paths defined in the `alternate.address` subparameters. The least used path becomes the primary path. If `alternate.direction=TOP`, the communications path defined in the adjacent node record is used as the primary path.

If the Process fails to establish a session using the primary communications path, the network map is processed to determine the next eligible communications path. This is repeated until the session is successfully established or the number of communications paths is reached. When this number is reached, the primary communications path is restored and the Process is placed in timer retry (TI RE) status. After all session retries (MAXRETRIES) are reached, the original communications path is restored and the Process is placed in hold (HO WC) status.

Each failed attempt to establish a session produces the following result:

- Message SVTM310I SESSION NOT ESTABLISHED WITH SNODE=xxxx
- Session Begin statistics record that identifies the communications path used at the time of the failure
- Diagnostics to RPLERRCK
- When the MAXRETRIES is reached and the Process is placed in the HOLD queue, the following message is issued in addition to the previous message(s):

SVTM105I PNAME=pname, PNUM=pnum MOVED TO Q=HOLD, QSTATUS=WC

When the Process establishes a session, the Session Begin statistics record identifies the communications path that was successfully used. That communications path is used for the life of the Process, unless `ALTErnate.RESTART` is coded for the node, in which case the communications path may change if the Process restarts. For more information on `ALTErnate.RESTART`, see the description for the `ALTErnate.COMMinfo` parameter.

Alternate Communications Paths with Current Node as Secondary Node

When the current IBM Connect:Direct node is the secondary node (SNODE), the `ALT.COMM` parameter is used for network map checking purposes only.

Outbound Processes

The `ALT.COMM` parameter is used for outbound Processes when following conditions are true:

- The current IBM Connect:Direct is the PNODE.
- The Process is not in restart (the default `ALT.RESTART=NO` is in effect).

Note: If you specify `ALT.RESTART=YES`, IBM Connect:Direct considers alternate communications paths if the path used to establish a session fails and the Process restarts. For an example, see [Restarting Processes using an Alternate Communications Path](#)

- The Process is not `PNODE=SNODE`.
- The Process is not `SNODE=TCPNAME=`.

PNODE=SNODE Processing

IBM Connect:Direct can initiate Processes where the local node is both the initiating and target node (PNODE and SNODE). You enable `PNODE=SNODE` processing by creating an adjacent node entry with the same node name as the local node entry in the network map.

If the `PNODE=SNODE` connection is SNA, `SNA=YES` is specified in the initialization parameters.

Observe the following rules when setting up the adjacent node:

- Define an LU0 communications name for the PNODE=SNODE network map entry. Do not specify an LU6.2 logmode entry name for the common name of the adjacent node entry.
- The communications name for the adjacent node must be different from the communications name of the local node. If the names are the same, PNODE=SNODE processing is disabled at initialization.
- Code the PARSESS=(max, default) parameter in the network map adjacent node entry to govern the number of simultaneous PNODE=SNODE connections.

TCP/IP Considerations

A network map entry is not required for every TCP/IP node. A default entry provides a standard set of parameters. If the standard set of parameters is not adequate, you can code a network map entry for that node to override the default entry. If network map entries are not used for TCP/IP, you must submit Processes by using the SNODE=TCPNAME keyword. For more information, see [Example - Defining a TCP/IP Default Entry](#).

Note: If you set the NETMAP.CHECK=TCP initialization parameter, you must define the TCP/IP node in the local network map.

The APPLIDS and LOGMODE keywords and the remote library name positional parameter are not used on any TCP/IP node and cannot be coded in the network map. A warning is generated for any unnecessary keyword or subparameter, and the coded value is ignored when the network map is loaded.

TCP/IP Addressing

Each host on a TCP/IP network is assigned a unique address known as an IP address. Applications running on a TCP/IP host that connect to the network are assigned one or more port numbers of the IP address.

IBM Connect:Direct supports both IPv4 and IPv6 protocols, where IPv6 allows a much larger address range than IPv4. During product initialization, IBM Connect:Direct determines if the TCP Stack is IPv6 or only IPv4. (IPv6 must be enabled within TCP/IP itself.) If IBM Connect:Direct initializes on a system where IPv6 is not enabled, any function involving an IPv6 address fails, including attempts to resolve an address or name using the DNS name resolution. For information about enhancing and extending IPv6 through the TCP.LISTEN parameter, see [Multiple Port TCP/IP Listen](#).

Command Syntax for IPv4 AND IPv6 Addressing

The following examples demonstrate the command syntax for IPv4 and IPv6 addressing.

Note: The address must be specified on a single line. You cannot break the address between lines.

The following example demonstrates how to break the command syntax over an additional line:

```
(SC.DUB.MWATL3,4399, -
1111:2222:3333:4444:5555:6666:7777:8888,TCP)
```

The following example demonstrates the appropriate network syntax:

```
/*          */
/* IPV6 ADDRESS */
/*          */
ADJACENT.NODE=( -
(SC.DUB.MWATL3,4399,1111:2222:3333:4444:5555:6666:7777:8888,TCP) -
PARSESS=(00000255 00000001) -
APPLIDS=(M1CDI7P6 M1CDI7P7 M1CDI7P8) )
/*          */
/* IPV4 ADDRESS */
/*          */
ADJACENT.NODE=( -
(SC.DUB.MWATL3,4399,111.222.333.444,TCP) -
PARSESS=(00000255 00000001) -
APPLIDS=(M1CDI7P6 M1CDI7P7 M1CDI7P8) )
```

The following example demonstrates the appropriate command syntax for a multiple listen for IPv6 addresses through the initialization parameter:

```
TCP.LISTEN=((10.20.201.2,4199), -  
            (fd00:0:0:20cc::2,4299), -  
            (10.20.201.2,4399), -  
            (10.20.201.2,4499) )
```

The following two examples represent a simple specification for a single listen for either an IPv4 or IPv6 address through the initialization parameters:

```
/*          */  
/* IPv4 address */  
/*          */  
TCP.LISTEN = (111.222.333.444,01364)
```

```
/*          */  
/* IPv6 address */  
/*          */  
TCP.LISTEN = (1111:2222:3333:4444:5555:6666:7777:8888,01364)
```

The following example represents a simple specification for a single listen for either an IPv4 or IPv6 address through the Process syntax:

```
/*          */  
/* IPv6 Address */  
/*          */  
label PROCESS -  
  SNODE=TCPNAME=1111:2222:3333:4444:5555:6666:7777:8888;nnnnn  
/*          */  
/* IPv4 Address */  
/*          */  
label PROCESS SNODE=TCPNAME=111.222.333.444;nnnnn  
(where nnnnn is the 5 digit port number)
```

Domain Name Resolution

Because IP addresses are difficult to remember and not descriptive of the host it is assigned to, TCP/IP enables meaningful names to map to the IP address. TCP/IP provides a function called Domain Name Resolution to map the name to the IP address. Refer to your TCP/IP implementation documentation for information on the setup and use of Domain Name Resolution.

Releasing Processes Serially from Connect:Direct for Microsoft Windows

To support dial-up connections from a Connect:Direct for Microsoft Windows node and release Processes serially, define the Connect:Direct for Microsoft Windows node in an adjacent node entry using a null IP address of 0.0.0.0. You can let the port number default to 1364 on the node where the network map resides since it will be resolved at connection time. Processes submitted to nodes defined in this manner default to HOLD=CALL status and are not executed since the connection cannot be resolved.

To release the HOLD=CALL Processes, create a NULL or ENABLE Process from the Connect:Direct for Microsoft Windows SNODE. (A NULL Process is an empty one with no steps and is only valid in Connect:Direct for Microsoft Windows.) When the Connect:Direct for Microsoft Windows node establishes a connection with Connect:Direct for z/OS and sends the NULL Process, Connect:Direct for z/OS uses the same session and runs any HOLD=CALL Processes one at a time. Checkpoint restart is supported for such nodes.

TCP/IP Port Number

The initialization parameters TCP.LISTEN and TCP.API.LISTEN provide the best method for defining a single port or multiple ports for IBM Connect:Direct to listen to and accept incoming TCP/IP connection requests. You can define a single port for establishing a listen task, or define multiple ports by establishing a list of IP address and port number combinations. Multiple port listening allows IBM Connect:Direct to accept incoming traffic from a variety of addresses.

Single Port TCP/IP Listen

IBM Connect:Direct implements a standard TCP client/server architecture. Each IBM Connect:Direct node in the TCP/IP network can function as both the client and the server simultaneously.

The server establishes a connection with the TCP/IP network and waits for session requests from other IBM Connect:Direct clients. When a session request is received and validated, the server accepts the connection to the client.

The IBM Connect:Direct client requests a session that is established through the TCP/IP network to a IBM Connect:Direct server. When the session is accepted by the server, a dynamic port is assigned by the network. This dynamically assigned port is used for the actual data transfer. When the data transfer is complete, the port is released back to the network and the session is terminated.

IBM Connect:Direct uses a predefined TCP/IP port number for both client and server. This port is defined in TCP/IP as a TCP service and requires the same port for all IBM Connect:Direct TCP/IP nodes in your network. Refer to your TCP/IP implementation documentation for how to define TCP servers. If you cannot use the predefined TCP/IP port, you can override it for both the client and server.

This initialization parameter defaults to port number 1364.

Client Override Port Number

The client override port number is coded on the adjacent node record in the second positional parameter. The IBM Connect:Direct client requests sessions to the IBM Connect:Direct server through this port. The port number coded on the adjacent node record is used by client functions only and does not change the port number used by the server functions. You must code the network map entry in your network map in order to use the override port number.

Server Override Port Number

The server override port number is coded in the TCP.LISTEN initialization parameter. The IBM Connect:Direct server waits for a IBM Connect:Direct client request on this port. The initialization port number is used by server functions only and does not change the port number used by the client functions.

Multiple Port TCP/IP Listen

IBM Connect:Direct accepts incoming traffic from a variety of addresses. The initialization parameters TCP.LISTEN and TCP.API.LISTEN support both IPv4 and IPv6 protocols. They also allow for a list of IP address and port number combinations.

The first address defined in the parameter becomes the local or default address. Up to eight different addresses/ports can be defined for each server.

For information about multiple port listening in a IBM Connect:Direct/Plex environment, refer to [A New IBM Connect:Direct/Plex Environment](#).

Viewing a TCP Listen Status Report

To view the TCP listening status through the IUI interface, follow this procedure.

Procedure

1. Access the INQUIRE TCP command by selecting option INQ from the Connect:Direct Administrative Options Menu. The Inquire DTF Internal Status screen is displayed.
2. Type ITCP and press ENTER to display the status report. The following report is an example of the output.

```
Menu Utilities Compilers Help
-----
BROWSE    SYS18242.T154700.RA000.EPETE1.NDMAPI.H0F
Command ==>
***** Top of Data *****
=====
CD.ART          * INQUIRE TCP      * DATE: 08.30.2018  TIME: 15:47:04
=====
ADDRESS                                PORT  FAMILY  TYPE  STATUS
=====
10.120.131.91      5610   IPV4    NODE  LISTEN
10.100.130.91      5610   IPV4    NODE  ERROR -  CANNOT ASSIGN ADDRESS
FD00:0:0937:6F00::0091  5610   IPV6    NODE  LISTEN
10.120.131.91      5620   IPV4    API   LISTEN
FD00:0:0937:6F00::0091  5620   IPV6    API   LISTEN
***** Bottom of Data ****
```

VTAM Independence

VTAM independence enables IBM Connect:Direct to initialize without SNA support. It also enables IBM Connect:Direct to continue functioning if VTAM is not available, and to reattach to VTAM when it is restored.

To use VTAM independence, you must specify SNA=YES in the initialization parameters. You must also specify a valid VTAM APPLID in the network map local node record.

If SNA= YES is specified in the initialization parameters and you try to start IBM Connect:Direct when SNA is not available, or if SNA becomes unavailable during a session, the system prompts the operator for action.

See “SNA = YES | NO” on page 478 for more information about the SNA initialization parameter. See [API Signons](#) for more information about operator actions.

TCP/IP API Communications

The TCP.API.LISTEN initialization parameter enables an API to communicate with a IBM Connect:Direct using TCP/IP via OE sockets. Also, you can use the TCPAPI keyword parameter in the adjacent node record to define a default protocol for API communication.

The API must also use the SIGNON command to specify a transport type and communications address. See [API Signons](#) for more information. Additionally, refer to *Managing Sessions in IBM Connect:Direct for z/OS User Guide* for more information about the SIGNON command.

Example - Defining a TCP/IP Default Entry

The following example defines a TCP/IP default record using ADJACENT.NODE entries.

```
/*                                          */
/* The following entry is for the TCP/IP default entry      */
/*                                          */
/* ADJACENT.NODE=(PARSESS=(8,1)              -              */
/* ENVIRONMENT=ZOS                          -              */
/* (TCP.IP.DEFAULT, 2048,, TCP))            */
```

Note: For a TCP/IP default entry, the nodename parameter must be TCP.IP.DEFAULT and the session type must be TCP. A TCP/IP default entry is required if any Process uses SNODE=TCPNAME=.

In the previous example, nodename=TCP.IP.DEFAULT, communications name=2048, and session type=TCP. For information about adjacent node entries, see [Adjacent Node Entry](#).

When a Process is submitted, the port number value is determined in the following order:

1. If the node name within a Process is defined in the network map, the port number associated with the node name entry is used.
2. If the node name within a Process is not defined in the network map, the port number associated with the default entry is used.
3. If no port number exists in the communications name field of the default entry, the TCP.LISTEN initialization parameter is used.
4. If the TCP.LISTEN initialization parameter is not defined, the port number defaults to 1364.

The PARSESS value for the SNODE is determined in the following order:

1. If the node name within a Process is defined in the network map, the PARSESS value associated with the nodename entry is used.
2. If the node name within a Process is not defined in the network map, the PARSESS value associated with the default entry is used.
3. If no PARSESS value is in the default entry, the PARSESS value defaults to (1,0). A PARSESS value of (1,0) means that Processes to the nodes for which the default entry is used are single-threaded.

Channel-to-Channel Support

Connect:Direct for z/OS provides channel-to-channel support for direct channel links between z/OS platforms using the IBM ESCON CTCA or IBM 3088 CTCA support.

The syntax for creating a CTCA adjacent node follows.

```
ADJACENT.NODE=(
    (nodename,                                -
      channel-range-start-addr, addr-count, CTCA,, -
      SEND| RECV))                             -
    NETID=server name                          -
    PARSESS=(max default)                      -
    ENVIRONMENT=operating environment          -
    )
```

API Signons

The communication address used to establish a connection to IBM Connect:Direct is determined by the TRANSPORT parameter defined in the SIGNON command. The default for the TRANSPORT parameter is NET (NETMAP), which means that the protocol defined in the NETMAP adjacent node entry is used.

Parameter Value	Description
TRANSPORT = NET	<p>Default. When TRANSPORT is defined as NET, the signon process retrieves the adjacent node entry to determine if the TCPAPI parameter has been defined. If TCPAPI exists, then a TCP connection is attempted using the communications address defined. The communications port number is obtained from the TCPAPI parameter, and if the IP address exists in the TCPAPI parameter, it is also used. If the IP address does not exist in the TCPAPI parameter, it must be obtained from either the adjacent node or the LDNS parameter.</p> <p>If the TCPAPI does not exist, the APPLID parameter is retrieved, and SNA is used as the protocol.</p>

Parameter Value	Description
TRANSPORT = SNA	When TRANSPORT is defined as SNA, the signon process retrieves the adjacent node entry to determine the APPLID parameter, and SNA is used as the protocol. If the APPLID parameter does not exist, then an ESF SIGNON is performed.
TRANSPORT = TCP	When the TRANSPORT is defined as TCP, the communications address must be specified on the SIGNON command. To initialize IBM Connect:Direct without SNA support, refer to Initializing IBM Connect:Direct without SNA Support in the <i>IBM Connect:Direct for z/OS Configuration Guide</i> .

Adjacent Node API Signon Definition Examples

To only allow SNA API signons:

```
/* PNODE=SNODE WITH SNA API SIGNON ONLY */
ADJACENT.NODE=(( CD.ZOS.NODE,M1DEV93C) PARSESS=(53 2) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
```

To only allow TCP API signons using the IP address:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING IP ADDRESS */
ADJACENT.NODE=(( CD.ZOS.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
)
```

To only allow TCP API signons using the LDNS parameter:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING LDNS */
ADJACENT.NODE=(( CD.ZOS.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
)
```

To allow both SNA and TCP API signons:

```
/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.ZOS.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
or
/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.ZOS.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
```

Examples of Local and Adjacent Node Records

This section contains examples of local and adjacent node records for various platforms.

Local Node and Corresponding Adjacent Node Record

The following is an example of a local node entry and its corresponding adjacent node entry. Notice that the local and adjacent node names are the same (CD.DALLAS), but the local and adjacent

communications names (NDMAPP1 and NDMAPP2) are different. The local node shows a superuser password of XYZZY.

```
LOCAL.NODE=((CD.DALLAS,NDMAPP1,,XYZZY) -
            TCQ=(DSC.DALLAS.TCX DSC.DALLAS.TCQ))
/*
/* THE FOLLOWING ENTRY IS FOR THE LOCAL NODE
/*
/* ADJACENT.NODE=(PARSESS=(12,2) (CD.DALLAS,NDMAPP2) -
            APPLIDS=(NAI01 NAI02 NAI03 CDDD12 -
                    CDDD17 CDDD18 CDDD32 CDDD41 CDDD42))
```

The following is an example of a local node entry and its corresponding adjacent node entry where TCPAPI is defined.

```
LOCAL.NODE=((CD.DALLAS,NDMAPP1,,XYZZY) -
            TCQ=(DSC.DALLAS.TCX DSC.DALLAS.TCQ))
/*
/* THE FOLLOWING ENTRY IS FOR THE LOCAL NODE WHERE TCPAPI IS DEFINED
/* FOR USE BY Connect:Direct APIs TO SIGNON USING TCP/IP
/*
/* ADJACENT.NODE=(PARSESS=(12,2) (CD.DALLAS,NDMAPP2) -
            TCPAPI=(1363,111.222.333.444) -
            APPLIDS=(NAI01 NAI02 NAI03 CDDD12 -
                    CDDD17 CDDD18 CDDD32 CDDD41 CDDD42))
```

The following is an example of an adjacent node entry that defines a pool of zFBA devices.

```
ADJACENT.NODE=((CD.UNIX , 1364 , UNIXHOST , TCP ) -
            ENVIRONMENT=UNIX PARSESS=(8,2) -
            ZFBA=(2000,2003) -
            APPLIDS=(M1CDI7P6 M1CDI7P7 M1CDI7P8) )
```

Alternate Communications Path Examples

The following examples show two different ways of using alternate communications paths.

Specifying ALT.ADDR and ALT.NODEDEF

The following is an example of an adjacent node entry that has four alternate communications paths that can be used if the main communication path specified for the adjacent node (TCP/IP address of 199.1.4.1) cannot be used. Three of the alternate communications paths are defined using the ALT.ADDR parameter while one uses the ALT.NODEDEF parameter.

```
ADJACENT.NODE=((CD.NODE,1364,199.1.4.1,TCP) -
            PARSESS=(6 2) -
            ALT.COMM=((ALT.ADDR=1.1.1.2,ALT.TYPE=TCP), -
                    (ALT.ADDR=VTAMAPL1,ALT.TYPE=SNA), -
                    (ALT.NODEDEF=TEST.NODE), -
                    (ALT.ADDR=1.1.1.3,ALT.PORT=4374,ALT.TYPE=TCP)) -
            )
```

Restarting Processes using an Alternate Communications Path

In the following example, the adjacent node in the previous example has been redefined to allow an alternate communications path which only supports Version 2 flows. In this system, IBM Connect:Direct is running on a machine with two Network Interface Cards (NICs), which means that there are two IP addresses for this machine. On the remote IBM Connect:Direct system, both of these IP addresses are included in the network map adjacent node entry for the original IBM Connect:Direct system.

In this example, assume a Process is started by the remote IBM Connect:Direct to this adjacent node with two NICs. The first path (TCP/IP address of 1364,199.1.4.1) is selected when the Process establishes its session. For some reason, the NIC fails on the machine, and the Process fails and goes into retry.

If ALT.RESTART=NO (the default), this Process can never restart successfully because it will always use the path used during initial session establishment (in this case, the failed NIC). When ALT.RESTART=YES is specified, the Process will try all ALT.COMM paths and will restart successfully because an additional path is available (ALT.ADDR=1.1.1.3,ALT.PORT=4374). ALT.NODEDEF points to another network entry and only uses the primary address for that node. Connection failure using the primary address does not result in running that node's ALT.COMM parameters (assuming it has any).

```
ADJACENT.NODE=((CD.NODE,1364,199.1.4.1,TCP) -
  PARSESS=(6 2) -
  ALT.COMM=(ALT.RESTART=YES,ALT.DIR=TOP,-
    (ALT.ADDR=1.1.1.3,ALT.PORT=4374,ALT.TYPE=TCP)) -
  )
```

Connect:Direct for z/OS Adjacent Node Examples

The following examples show how to define adjacent z/OS nodes.

SNA LUO

The following example shows an adjacent z/OS node named CD.NYCZOS, with the communications name (VTAM APPLID) of CDDD10. The APPLIDS parameter indicates nine API sessions are possible.

```
ADJACENT.NODE=(PARSESS=(4,2) (CD.NYCZOS,CDDD10) -
  APPLIDS=(CDAPI01 CDAPI02 CDAPI03 CDAPI04 -
    CDAPI05 CDAPI06 CDAPI07 CDAPI08 CDAPI09))
```

CTCA

The following example shows an adjacent z/OS node named CD.DALLAS.ZOS1 which uses channel-to-channel adapter addresses E001-E008. The first address of each pair is used for outbound traffic; the second address is used for inbound traffic.

```
ADJACENT.NODE=(PARSESS=(4,1) -
  ENVIRONMENT=ZOS -
  (CD.DALLAS.ZOS1,E001,8,CTCA,,SEND))
```

In the network map entry of the adjacent node, an adjacent node entry for the other side of the CTCA connection is in the following example. The z/OS node named CD.DALLAS.ZOS2 uses channel-to-channel adapter addresses F001-F008. The first address of each pair is used for inbound traffic. The second address of each pair is used for outbound traffic.

```
ADJACENT.NODE=(PARSESS=(4,1) -
  ENVIRONMENT=ZOS -
  NETID=SERVER1 -
  (CD.DALLAS.ZOS2,F001,8,CTCA,,RECV))
```

TCP/IP and UDT

The following example shows three adjacent node entries with session protocol types of TCP/IP and UDT:

- The first, with a TCP/IP net name of ZOS.CD.CHICAGO, specifies the default TCP/IP port number by leaving the communications name positional parameter null.
- The second, with a TCP/IP net name of ZOS.CD.DALLAS, specifies a UDT port number of 4444. The IP address will default to the node name, to be resolved by domain name resolution.

- The third, with a TCP/IP net name of ZOS.CD.AUSTIN, specifies a TCP/IP port number of 4443 and an IP address of 199.8.8.8.
- The fourth defines a TCPAPI to use port number 4442 and obtain the IP address from the adjacent node record.
- The fifth defines the LDNS parameter.

```

ADJACENT.NODE=(PARSESS=(4,2) -
                (ZOS.CD.CHICAGO,,199.1.4.51,TCP) -
                ENVIRONMENT=ZOS)

ADJACENT.NODE=(PARSESS=(4,2) -
                (ZOS.CD.DALLAS,4444,,UDT33) -
                ENVIRONMENT=ZOS)

ADJACENT.NODE=(PARSESS=(4,2) -
                (ZOS.CD.AUSTIN,4443,199.8.8.8,TCP) -
                ENVIRONMENT=ZOS)

ADJACENT.NODE=(PARSESS=(4,2) -
                (ZOS.CD.AUSTIN,4443,199.8.8.8,TCP) -
                TCPAPI=(4442,) -
                ENVIRONMENT=ZOS)

ADJACENT.NODE=(PARSESS=(4,2) -
                (ZOS.CD.AUSTIN,4443,,TCP) -
                LDNS=(TCP.AUSTIN.DOMAIN) -
                TCPAPI=(4442,) -
                ENVIRONMENT=ZOS)

```

SNA LU6.2

The following example shows an adjacent node entry for a node named CD.LAZOS with a communications name (APPLID) of APPLLAI and a session protocol type of LU6.2. The operating environment of this adjacent node is z/OS, and the VTAM logmode entry which defines the session protocol used when communicating with this node is LU62MOD4. The LOGMODE parameter is required for LU6.2.

```

ADJACENT.NODE=(PARSESS=(4,2) -
                (CD.LAZOS,APPLLAI,,LU62) -
                ENVIRONMENT=ZOS LOGMODE=LU62MOD4 -
                APPLIDS=(CDDD2,CDDD3,CDDD4))

```

Trusted Node

The following example shows an adjacent node entry for a node named SC.NODE.A with a security type of external (EXT) and data direction restriction of SEND.

```

ADJACENT.NODE=(PARSESS=(4,2) -
                (SC.NODE.A,NZOSD20,,,EXT,SEND) -
                APPLIDS=(NZOSA36,NZOSA37,NZOSA38))

```

VM/ESA SNA LU0 Adjacent Node Example

The following example shows an adjacent node named CD.BOSTON.VM, with a communications name (APPLID) of CDDD16.

```

ADJACENT.NODE=(PARSESS=(4,2) (CD.BOSTON.VM,CDDD16) -
                APPLIDS=(CDAPI01 CDAPI02 CDAPI03 CDAPI04 -
                CDAPI05 CDAPI06 CDAPI07 CDAPI08 CDAPI09))

```

OpenVMS Adjacent Node Example

The following example shows an adjacent node named CD.DALLAS.VMS. The SNODE.LUS parameter specifies the logical unit names used by the local node to initiate a session with this remote node.

```

ADJACENT.NODE=((CD.DALLAS.VMS) -
PARSESS=(8,1) -
PNODE.LUS=(N91LU09 N91LU0A N91LU0B N91LU0C -
N91LU0D N91LU0E N91LU0F N91LU10) -
SNODE.LUS=(N91LU07 N91LU08))

```

Microsoft Windows Adjacent Node Example

The following example show how to define adjacent Microsoft Windows nodes.

```

ADJACENT.NODE=(
(WIN.TCPIP.NODE,1364,123.4.5.67,TCP) -
PARSESS=(20,1) -
ENVIRONMENT=WINDOWS -
)

```

UNIX Adjacent Node Examples

The following examples show how to define adjacent UNIX nodes.

TCP/IP Adjacent Node Example

The following example shows two adjacent node entries with session protocol types of TCP/IP:

- The first entry specifies the default TCP/IP port number by leaving the communications name positional parameter null. The IP address will default to the node name to be resolved by domain name resolution.
- The second entry specifies a TCP/IP port number of 5555 and an IP address of 199.5.5.5.

```

ADJACENT.NODE=(PARSESS=(6,2) -
(UNIX.CD.CHICAGO,,,TCP) -
ENVIRONMENT=UNIX)
ADJACENT.NODE=(PARSESS=(6,2) -
(UNIX.CD.DALLAS,5555,199.5.5.5,TCP) -
ENVIRONMENT=UNIX)

```

Notice that no APPLID or LOGMODE keywords are used for any TCP/IP node. A warning is generated for any unneeded keyword or subparameter, and the coded value is ignored.

LU6.2 Adjacent Node Example

The following example shows an adjacent UNIX node with a communications name (APPLID) of D1UNIX and a session protocol type of LU6.2. The logmode entry name is LU62MODE. The LOGMODE parameter is required for LU6.2.

```

ADJACENT.NODE=(PARSESS=(6,2) -
(UNIX.LU62.DALLAS,D1UNIX,,LU62) -
LOGMODE=LU62MODE -
ENVIRONMENT=UNIX)

```

Stratus VOS Adjacent Node Examples

The following examples show how to define adjacent Stratus VOS nodes.

TCP/IP

The following example shows an adjacent node entry with a session protocol type of TCP/IP, a TCP net name of CD.STRAT, a TCP port number of 3333, and an IP address of 199.1.1.11.

```
ADJACENT.NODE=(PARSESS=(12,1) -  
  (CD.STRAT,3333,199.1.1.11,TCP) -  
  ENVIRONMENT=STRATUS)
```

LU0

The following example shows an adjacent Stratus VOS node with a communications name (APPLID) of M1T20404 and a session protocol type of LU0. The logmode entry name is CDPCLU0. The LOGMODE parameter is optional for LU0.

```
ADJACENT.NODE=(( CD.STRAT,M1T20404,,SNA) -  
  LOGMODE=CDPCLU0)
```

i5/OS Adjacent Node Examples

The following examples show how to define adjacent i5/OS nodes.

i5/OS SNUF

The following example shows an adjacent node named AS400.CD.TX with a remote library name of LBNAME and session protocol type of LU0 (SNUF). The SNODE.LUS parameter defines the dependent LU pool.

```
ADJACENT.NODE=(PARSESS=(4,2) -  
  (AS400.CD.TX,,LBNAME,SNUF) -  
  SNODE.LUS=(N11LU01,N11LU02,N11LU03,N11LU04))
```

LU6.2 with Independent LU

The following example shows an adjacent node named AS400.CD.LA with an independent LU communications name of APPLLA1, a remote library name of CDLIB1, a session protocol type of LU6.2, and a logmode entry name of LU62MOD2. The ENVIRONMENT=OS400 parameter is required for i5/OS nodes using the LU6.2 protocol. The LOGMODE parameter is required for the LU6.2 protocol.

```
ADJACENT.NODE=(PARSESS=(6,2) -  
  (AS400.CD.LA,APPLLA1,CDLIB1,LU62) -  
  ENVIRONMENT=OS400 LOGMODE=LU62MOD2)
```

LU6.2 with Dependent LU

The following example shows an adjacent i5/OS node named AS400.CD.NY with a remote library name of CDLIB1, a session protocol type of LU6.2, and a logmode entry name of LU62MOD3. The SNODE=LUS parameter defines the dependent LU pool. The ENVIRONMENT=OS400 parameter is required for i5/OS nodes using the LU6.2 protocol. The LOGMODE parameter is required for the LU6.2 protocol.

```
ADJACENT.NODE=(PARSESS=(4,2) -  
  (AS400.CD.NY,,CDLIB1,LU62) -  
  ENVIRONMENT=OS400 LOGMODE=LU62MOD3 -  
  SNODE.LUS=(NYLU01,NYLU02,NYLU03,NYLU04))
```

TCP/IP

The following example shows an adjacent node entry with a node name of OS400.TCP.NODE, session protocol type of TCP, a TCP port number of 1364, and an IP address of 199.1.1.11.

```
ADJACENT.NODE=( -  
  (OS400.TCP.NODE,1364,199.1.1.11,TCP,INT,BOTH) -  
  CRC=DEFAULT -  
  PARSESS=(20,2))
```

How to Update the Network Map

The network map is created during installation, when the network map source is input to the network map load program.

The network map source contains one local node entry and multiple adjacent node entries. It can contain \$\$ACTION verbs added during previous maintenance.

You can update the network map source while IBM Connect:Direct is not executing or dynamically while IBM Connect:Direct is executing.

Note: To test connectivity to an ADJACENT.NODE, use the DGAPHTBT Process (see \$CD.SDGASAMP). This Process makes a connection to the ADJACENT.NODE and executes a RUN TASK at the local node. Be aware that the connection causes both IBM Connect:Direct nodes to search their TCQ files for Processes destined for the other node.

Updating the Network Map while IBM Connect:Direct is Not Executing

About this task

You can update the network map using the network map source and the DGAJLOAD JCL which loaded the source at initialization.

You can only update the local node by performing the following steps:

Procedure

1. Change the network map source. The network map source is loaded at installation into \$CD.SDGACNTL(NETMAP01)
2. Stop IBM Connect:Direct.
3. Delete and redefine the network map. Refer to the JCL in \$CD.SDGAJCL(DGAJJDEF).
4. Reload the network map. Refer to the JCL in \$CD.SDGAJCL(DGAJLOAD).
5. Restart IBM Connect:Direct.

Updating the Network Map while IBM Connect:Direct is Running

You can update the network map without deleting and redefining it. In addition, you can update the network map source without stopping IBM Connect:Direct, by using the UPDATE NETMAP command.

After updating the network map with UPDATE NETMAP, you can refresh the network map for any Processes in the Wait queue with the Change Process command. See the *IBM Connect:Direct for z/OS User Guide* for a description of the Change Process command.

Note: Any changes to the CTCA definition do not take effect until IBM Connect:Direct is reinitialized.

As with most commands, you can execute the command through a batch job or through the IUI. Both methods use \$\$ACTION verbs as part of the network map source.

Note: This method of updating the network map is only available for adjacent nodes.

The format of the UPDATE NETMAP command follows.

Label	Command	Parameters
(optional)	UPDATE NETMAP	WHERE (

Label	Command	Parameters
		NETINput = filename(member name)
		NETLOG =[ddname NONE]
)
		DIS PRT

Note: Reinitialize IBM Connect:Direct before CTCA definition changes or additions are effective.

Required Parameter

WHERE is the only required parameter for the UPDATE NETMAP command.

Parameter	Description
WHERE (NETINput = filename (member name) NETLOG = [ddname NONE])	<p>Network map source file and where the update activity is reported.</p> <p>NETINput = filename (member name) specifies the name of the network map source file. This file can be sequential or a PDS member. The network map source can contain multiple basic action verbs, multiple special purpose action verbs, or a combination of both. The source that updates the network map can be the entire network map source or a subset of it.</p> <p>NETLOG = [ddname NONE] specifies where the update activity is reported.</p> <p>ddname specifies the data definition name allocated to the IBM Connect:Direct DTF where the update activity is reported. The first two characters of the ddname must be CD.</p> <p>NONE specifies that no update activity is reported.</p> <p>If the parameter is left blank the update activity is reported to the CDLOG data set. Regardless of which option is selected the activity is recorded in the Statistics file as WTO records.</p>

Optional Parameter

Parameter	Description
DIS PRT	<p>Use the DIS or the PRT optional parameter to specify the output destination.</p> <p>DIS indicates that the activity is reported in display format, either to the screen for IUI requests or to the DDNAME for batch requests.</p> <p>PRT indicates that the output is routed to SYSOUT for batch requests and to the print destination specified by the PRINT FILE DESTINATION parameter in the signon defaults.</p>

Using \$\$ACTION VERBS in the Network Map

Add \$\$ACTION verbs to the network map source as described in the description of the NETINPUT parameter of the UPDATE NETMAP command. For more information on the required parameter, NETINPUT, refer to “Updating the Network Map while IBM Connect:Direct is Running” on page 274. Each verb defines the action to take for the node entry immediately following the action verb. Three basic action verbs and three special purpose action verb pairs exist. The following table describes the action verbs:

Action Verb	Description
\$\$INSERT	Inserts the following node entry into the network map.

Action Verb	Description
\$\$UPDATE	<p>Updates the following existing network map node entry. Node entry updates are performed as a replacement at the keyword level. Therefore, updates of list-type keywords, like APPLIDS, require that you specify the entire list.</p> <p>Do not use \$\$UPDATE to update network map node entries which already contain the LDNS parameter. Use \$\$DELETE and \$\$ INSERT to change nodes which contain an LDNS parameter.</p> <p>Note: You can use \$\$UPDATE to add the LDNS parameter to a node which did not previously contain one.</p>
\$\$REPLACE	Replaces the present node entry with the one following the \$\$REPLACE verb. This verb first does a \$\$DELETE and then a \$\$INSERT. It can be used in place of \$ \$UPDATE.
\$\$DELETE	Deletes the following existing network map node entry.
\$\$SYNTAX and \$ \$ENDSYNTAX	Performs a syntax check of the network map control statements following this verb.
\$\$VERIFY and \$ \$ENDVERIFY	Verifies that the node definitions following this verb match those in the network map.
\$\$BLKxxxxxx and \$ \$ENDxxxxxx	Performs the basic action verb defined by xxxxxx for the block of node entries following this verb. Replace xxxxxx with either INSERT, UPDATE, or DELETE.

Updating the Netmap through the Batch Interface

About this task

To issue the UPDATE NETMAP command through the IBM Connect:Direct batch utility, follow this procedure.

Procedure

1. Change the network map source using the \$\$ACTION verbs.
2. Place the UPDATE NETMAP commands in the batch job stream.
3. Ensure that IBM Connect:Direct is running.
4. Submit the job.
5. Correct any errors identified on the activity report and resubmit if necessary.
6. Verify the results.

Updating the Netmap through the IUI Interface

About this task

To issue the UPDATE NETMAP command through the IBM Connect:Direct IUI, perform the following steps.

Note: Updating the network map through the IUI can take significant time. For mass updates, consider batch processing.

Procedure

1. To change or create a new member with your updates, change the network map source to use the \$ \$ACTION verbs.
2. Ensure that IBM Connect:Direct is running.

3. Access the Update network map screen by selecting option UNM from the Administrative Options Menu.

```
node.name          UPDATE NETWORK MAP          hh:mm
CMD ==>

      WARNING: THIS COMMAND CAN TAKE SIGNIFICANT TIME. FOR MASS
              UPDATES, BATCH PROCESSING SHOULD BE CONSIDERED!

ENTER NETMAP INPUT FILE NAME:
==> -----

ENTER MEMBER NAME:      (OPTIONAL)
==> -----

ENTER DDNAME FOR LOG FILE OR "NONE":      (DEFAULTS TO "CDLOG",
==> -----                                FIRST TWO CHARACTERS MUST BE "CD")

      OUTPUT DESTINATION ==> DIS  (DIS-DISPLAY,PRT-PRINT)
```

4. Type the network map source file name and appropriate optional parameters and press ENTER. Unless you select PRINT on the Update NETMAP screen, the report routes to your terminal.
5. Verify the results on the activity report.
6. Correct any errors identified on the activity report and re-type if necessary.

\$\$ACTION Verb Examples

The following are examples of updating the network map through the use of action verbs.

\$\$INSERT Example

The following \$\$INSERT command inserts an adjacent node into the network map.

```
$$INSERT
  ADJACENT.NODE=((CD.NODE2  APPLID2  ) -
                PARSESS=(5,2) -
                APPLIDS=(RAPPL1))
```

The output follows.

```
= = > * * *   START NETMAP UPDATE           * * *
= = >   DATE:  02/27/2003 TIME=14:59:26
= = >   SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
*INSERT THE FOLLOWING NODE DEFINITION
$$INSERT
  ADJACENT.NODE=(( CD.NODE2  APPLID2 )  -
                PARSESS=(5,2)          -
                APPLIDS=(RAPPL1 ))
= = >   SMUP032I APPLIDS RECORD INSERTED
= = >   SMUP034I ADJACENT.NODE RECORD INSERTED
= = >   SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE2
=====
```

The first message, SMUPNLGI, shows that logging is not requested; therefore, IBM Connect:Direct does not keep a record of the transaction (except in the statistics file). The last messages indicate that the information for the specified adjacent node was successfully inserted.

\$\$UPDATE Example

The following \$\$UPDATE command updates an adjacent node in the network map by adding the RAPPL2 APPLID and changing the maximum parallel sessions to four.

```

$$$UPDATE
  ADJACENT.NODE=(( CD.NODE2  APPLID2 )  -
                PARSESS=(4,2)          -
                APPLIDS=(RAPPL1 RAPPL2 ))

```

The output follows.

```

== > * * *  START NETMAP UPDATE      * * *
== >  DATE: 02/27/2003 TIME=14:59:26
== >  SMUPNLGI LOGGING ACTIVE - LOG DDNAME=CDLOG
=====

*UPDATE THE FOLLOWING NODE DEFINITION ADDING RAPPL2
*CHANGING MAXIMUM PARALLEL SESSIONS TO FOUR (4).
$$$UPDATE
  ADJACENT.NODE=(( CD.NODE2  APPLID2 )  -
                PARSESS=(4,2)          -
                APPLIDS=(RAPPL1 RAPPL2))

== >  SMUP032I APPLIDS RECORD UPDATED
== >  SMUP034I ADJACENT.NODE RECORD UPDATED
== >  SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE2
=====

```

The first message, SMUPNLGI, shows that logging is requested and that a record of the transaction is recorded in CDLOG. The last messages indicate that the adjacent node information was successfully updated.

\$\$\$REPLACE Example

The following \$\$\$REPLACE command deletes then inserts an adjacent node in the network map.

```

$$$REPLACE
ADJACENT.NODE=((CD.NODE2 APPLID2) -
PARSESS=(4,2) -
APPLIDS=(RAPPL1 RAPPL2))

```

The output follows.

```

==> * * *  S T A R T   N E T M A P   U P D A T E   * * *
==>  DATE: 04.02.2010  TIME=14:18:47
==>  SMUPLOGI LOGGING ACTIVE - LOG DDNAME=CDLOG
=====

$$$REPLACE
  ADJACENT.NODE=(  -
                  (CD.NODE2,APPLID2)  -
                  PARSESS=(6 2)  -
                  APPLIDS=(RAPPL1 RAPPL2)  -
                  )

==>  SMUP034I ADJACENT.NODE record "DELETED ".
==>  SMUP032I APPLIDS record "INSERTED".
==>  SMUP034I ADJACENT.NODE record "INSERTED".
==>  SMUP008I Request successful for Node=CD.NODE2
=====

==>  SMUP000I C:D Network Map processing completed.

==>  DATE: 04.02.2010  TIME=14:18:47
==> * * *  E N D   N E T M A P   U P D A T E   * * *

```


\$\$DELETE Example

The following \$\$DELETE command deletes an adjacent node from the network map.

```
$$DELETE
  ADJACENT.NODE=((CD.NODE2  APPLID2) -
    PARSESS=(4,2) -
    APPLIDS=(RAPPL1 RAPPL2))
```

The output follows.

```
= = > * * *   START NETMAP UPDATE           * * *
= = >   DATE:  02/27/2003 TIME=15:09:36
= = >   SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
$$DELETE
  ADJACENT.NODE=((CD.NODE2  APPLID2) -
    PARSESS=(4,2) -
    APPLIDS=(RAPPL1 RAPPL2 ))
= = >   SMUP032I APPLIDS RECORD DELETED
= = >   SMUP034I ADJACENT.NODE RECORD DELETED
= = >   SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE1
=====
```

The first message, SMUPNLGI, indicates that logging is not requested, so IBM Connect:Direct does not keep a record of the transaction. The last messages indicate that the APPLIDs and adjacent node records are successfully deleted.

\$\$SYNTAX Example

The following \$\$SYNTAX command performs a syntax check on the specified nodes.

```
$$SYNTAX
  LOCAL.NODE=((CD.NODE1  APPLID1 , , SUPERUSR) -
    TCQ=( TCQ TCX ))
  ADJACENT.NODE=((CD.NODE1  APPLID1) -
    PARSESS=(5,2) -
    APPLIDS=(LAPPL1  LAPPL2  LAPPL3))
  ADJACENT.NODE=((CD.NODE2  APPLID2) -
    PARSESS=(5,2) -
    APPLIDS=(RAPPL1 ))
$$ENDSYNTAX
```

The output follows. The messages are numbered in the example for clarification; they are not numbered on the actual output.

```

      = = > * * *          START NETMAP UPDATE          * * *
      = = > DATE:  02/27/2003 TIME=13:49:16
(1)    = = > SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
      $$SYNTAX
(2)    = = > SMUP011I 'SYNTAX    ' ACTION STARTED
=====
      LOCAL.NODE=(( CD.NODE1  APPLID1  ,, SUPERUSR) -
              TCQ=( TCQ TCX  ))
(3)    = = > SMUP005I LOCAL.NODE RECORD PROCESSING NOT ALLOWED
              BYPASSED
=====
      ADJACENT.NODE=(( CD.NODE1  APPLID1  ) -
              PARSESS=(5,2) -
              APPLIDS=(LAPPL1  LAPPL2  LAPPL3))
(4)    = = > SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE1
=====
      ADJACENT.NODE=(( CD.NODE2  APPLID2  ) -
              PARSESS=(5,2) -
              APPLIDS=(RAPPL1))
(4)    = = > SMUP008I REQUEST SUCCESSFUL FOR NODE=CD.NODE2
=====
      $$ENDSYNTAX
(5)    = = > SMUP012I 'SYNTAX    ' ACTION STOPPED
=====

```

The numbers in parentheses indicate:

1. Logging is not requested; therefore, no record is kept of the transaction.
2. Syntax check of network map control statements starts.
3. No processing is allowed against the local node record.
4. Requests for syntax checking on nodes are successful.
5. Syntax checking completes.

\$\$VERIFY Example

The following \$\$VERIFY command verifies the definition of the specified adjacent node record prior to updating the network map.

```

$$VERIFY
ADJACENT.NODE=((CD.NODE2 APPLID2) -
              PARSESS=(5,2) -
              APPLIDS=(RAPPL1))
$$ENDVERIFY

```

The output follows. The messages are numbered in the example for clarification; they are not numbered on the actual output.

```

      = = > * * *      START NETMAP UPDATE      * * *
      = = > DATE: 02/27/2003 TIME=15:35:16
(1)    = = > SMUPNLGI NETLOG=NONE REQUIRED, LOGGING INACTIVE
=====
      $$VERIFY

(2)    = = > SMUP011I 'VERIFY  ' ACTION STARTED
=====

      ADJACENT.NODE=(( CD.NODE2  APPLID2 ) -
      PARSESS=(5,2) -
      APPLIDS=(RAPPL1 ))
(3)    = = > SMUP092I  APPLIDS RECORD DID NOT MATCH
      = = > SMUP094I ADJACENT.NODE RECORD DID NOT MATCH
      = = > SMUP096I RECORDS DO NOT MATCH - VERIFICATION FAILED
              FOR NODE
              =CD.NODE2
=====

      $$ENDVERIFY

(4)    = = > SMUP012I 'VERIFY  ' ACTION STOPPED
=====

```

The number in parentheses indicate the following steps:

1. Logging is not requested; therefore, no record is kept of the transaction.
2. Verification of the node definition to the network map file has started.
3. The APPLIDs and adjacent node records did not match the network map file definitions.
4. Verification is complete.

Viewing the Network Map

You can view the network map online through the IUI interface or view the contents of the network map by unloading the network map VSAM file source.

Viewing the Netmap through the IUI Interface

To view the network map using the IBM Connect:Direct IUI, select option **NM** from the Primary Options Menu to display the SELECT NETMAP OR TCP INFORMATION screen. For more information, see *The Network Map* in the *IBM Connect:Direct for z/OS User Guide*.

Unloading the Network Map to the Source Format

IBM Connect:Direct provides a utility to unload the network map to its source format. You can then view the source format to see network map settings. This utility is provided in the DGAJUNLD member in the \$CD.SDGAJCL data set.

To unload the network map, submit the DGAJUNLD member. You can unload the network map while IBM Connect:Direct is running.

An example of the JCL follows.

```

//STEP      EXEC      PGM=DGADNTLD,PARM='UNLOAD'
//NETMAP DD      DSN=NETMAP.DATASET,DISP=SHR
//UNLOAD DD      DSN=NETMAP.UNLOAD,DISP=(NEW,CATLG),
//              DCB=(DSORG=PS,RECFM=FB,BLKSIZE=0,LRECL=80),
//              UNIT=SYSDA,SPACE=(TRK,(4,2,))

```

The network map source is unloaded to the data set specified in the JCL.

Configuring a IBM Connect:Direct/Plex Environment

IBM Connect:Direct runs in two configurations:

- IBM Connect:Direct Stand-alone Server, which is a single IBM Connect:Direct system operating within an IBM z/OS environment.
- IBM Connect:Direct/Plex, which is a IBM Connect:Direct system operating in an IBM z/OS sysplex or parallel sysplex environment consisting of a IBM Connect:Direct Manager and one or more IBM Connect:Direct Servers.

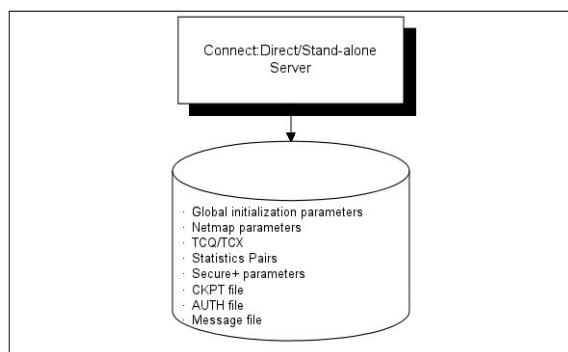
Differences Between Stand-Alone and Plex Environments

A IBM Connect:Direct Stand-alone Server and a IBM Connect:Direct/Plex environment have the following configuration differences:

- Initialization parameters

The two sets of initialization parameters in IBM Connect:Direct are global and local.

A IBM Connect:Direct Stand-alone Server uses only global initialization parameters to set system-wide values, as shown in the following illustration.



A IBM Connect:Direct/Plex environment uses both global and local initialization parameters. Global initialization parameters apply to each member of the IBM Connect:Direct/Plex environment. Local initialization parameters apply to specific members of the IBM Connect:Direct/Plex environment and can override some global initialization parameters affecting that member. Each IBM Connect:Direct/Plex member must have its own local initialization parameter member with the parameter, `CDPLEX.MANAGER = NO | YES` on page 499, as the first statement in the member.

Note: You can save a copy of a local initialization parameters member to have as a backup in case IBM Connect:Direct cannot start up successfully after initparm updates have been applied. Specify the name of the backup file using the local initialization parameter, `CDPLEX.INITPARM.BACKUP = member`. Be sure to specify this parameter for each IBM Connect:Direct/Plex member.

If an update of the global and local initialization parameter files is performed using Control Center and fails during the process, IBM Connect:Direct will use the backup members to restore the parameters. IBM Connect:Direct will never use the backup members to initialize the DTF unless you explicitly instruct the system to do so by updating the JCL to use the backup members as the initparm members.

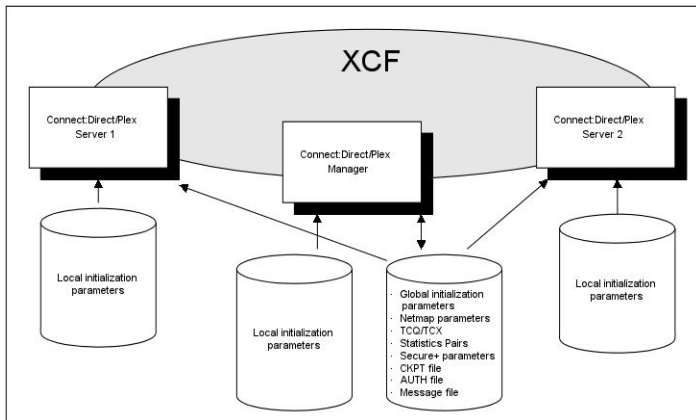
Global initialization parameters are stored in a file shared by all IBM Connect:Direct/Plex members. In the EXEC statement (IBM Connect:Direct Stand-alone Server and IBM Connect:Direct/Plex), the `PARM=` keyword specifies the name and location of the global initialization parameters file.

The local initialization parameters of each IBM Connect:Direct/Plex member are stored in a unique PDS member for each system (one for the Plex Manager and one for each server). The location of the local initialization parameters file is specified by the `//CDPLEX DD` in the startup JCL of each member.

In a IBM Connect:Direct/Plex environment, you can override only the initialization parameters allowed in the local initialization parameters member by using the `PARM=` keyword in the EXEC statement at system startup.

In a IBM Connect:Direct Stand-alone Server environment, however, you can override global initialization parameters with the PARM= keyword in the EXEC statement.

The following illustration shows how global and local initialization parameters are used in a IBM Connect:Direct/Plex environment.



- VTAM APPLIDs

A IBM Connect:Direct Stand-alone Server obtains its VTAM APPLIDs from the network map.

In a IBM Connect:Direct/Plex environment, the IBM Connect:Direct/Manager obtains its VTAM APPLIDs from the network map, but each IBM Connect:Direct Server obtains its VTAM APPLIDs from its local initialization parameters file.

- TCP/IP addresses and ports

A IBM Connect:Direct Stand-alone Server obtains its TCP/IP listen ports from the global initialization parameters file.

In a IBM Connect:Direct/Plex environment, the IBM Connect:Direct Manager and IBM Connect:Direct Servers obtain their TCP/IP addresses and listen port numbers from their local initialization parameters members. However, if the TCP port number is not specified in the local initialization parameters member of the IBM Connect:Direct Manager, the IBM Connect:Direct Manager obtains its listen port number from the global initialization parameters member.

Each server overrides the global initialization parameters by specifying those parameters in that server's local initialization parameters. The first address defined in the parameter becomes the local or default address. For more information about defining TCP/IP listening tasks, see [TCP/IP Port Number](#).

The CDPLEX.REDIRECT local initialization parameter is used by the IBM Connect:Direct/Plex Manager in the IBM Connect:Direct/Plex environment to determine the redirection address that is presented to the remote node. This parameter allows you to specify redirection addresses based on the security node type (internal or external) and session type (TCP/IP or UDT) of the adjacent node in the network map.

When an address is specified, an internal address and external address is defined and each can have a specified redirection port defined. When the adjacent node entry is defined with the INT flag, the appropriate internal address is returned. Conversely, when the EXT flag is defined, the appropriate external address is returned.

Note: If your environment requires additional external redirection addresses to enable a remote IBM Connect:Direct server to run Processes, see the local initialization parameter, [CDPLEX.REDIRECT.EXCEPTION = \(\(Mgr-IP, Ext_Svr-IP, Ext_Svr-port, Exception-IP, Exception-port\),...\)](#).

Up to eight different addresses or ports can be defined for each server. However, in the IBM Connect:Direct/Plex server that defines CDPLEX.REDIRECT only two are effectively used when Process redirection occurs. To use the additional port in the IBM Connect:Direct/Plex servers, those servers must be contacted directly by the remote node.

Note: A special consideration exists if the IBM Connect:Direct/Plex Manager is initialized on a system that is not IPv6 enabled, and one or more of the servers supports IPv6. When the IBM Connect:Direct/

Plex is the SNODE, the IBM Connect:Direct/Plex Manager accepts connection requests for IPv4 only. However, if the IBM Connect:Direct/Plex is the PNODE, the IBM Connect:Direct/Plex Manager can assign outbound processes to a IBM Connect:Direct Server that supports IPv6.

- System files

In a IBM Connect:Direct Stand-alone Server, the system files (network map, Statistics Pairs, CKPT, AUTH, Message, TCQ and TCX files) are stored in one location and apply to the entire DTF. If two IBM Connect:Direct Stand-alone Server systems operate in a sysplex environment, each system must have its own system files.

In a IBM Connect:Direct/Plex environment, the system files are also stored in one location and are shared by all IBM Connect:Direct/Plex members, as in the earlier illustration. Only one set of system files is needed for all IBM Connect:Direct/Plex members.

A New IBM Connect:Direct/Plex Environment

Use this setup for a new installation. The setup in this section assumes the following:

- You have installed and started a single IBM Connect:Direct DTF.
- You are changing a IBM Connect:Direct Stand-alone Server into a IBM Connect:Direct/Plex environment with three members: the IBM Connect:Direct Manager and two IBM Connect:Direct Servers. The IBM Connect:Direct Servers are named SERVER1 and SERVER2. SERVER1 has tape drive access for copy Processes requiring tapes.
- The global and local initialization parameters files are located in \$CD.PLEX.INITPARM (a PDS, not a PDSE, that you have built). The JCL to bring up the IBM Connect:Direct/Plex environment is located in \$CD.PLEX.JCL. You can either allocate these data sets or use existing data sets in their place. Do not use IBM Connect:Direct's SMP/E target PDSEs for these data sets.

Setting Up a New IBM Connect:Direct/Plex Environment

To set up a IBM Connect:Direct/Plex environment:

Procedure

1. Create a PDS (which will be referred to as \$CD.PLEX.INITPARM in this procedure) to hold the initialization parameters for the IBM Connect:Direct Plex environment, with sufficient directory space to handle ISPF Statistics. Create a PDS (which will be referred to as \$CD.PLEX.JCL) to hold the JCL.
2. Copy your current DTF's INITPARM member into \$CD.PLEX.INITPARM as member CDPLX.
3. Copy your current IBM Connect:Direct Stand-alone Server startup JCL into \$CD.PLEX.JCL as member CDMGR.
4. Add the following initialization parameters to the CDPLX member in \$CD.PLEX.INITPARM. This member becomes the IBM Connect:Direct/Plex global initialization parameters file.

```
CDPLEX=YES  
CDPLEX.MAXSERVER = number of servers | 4  
XCF.NAME=8-char-name
```

The CDPLEX=YES parameter indicates a IBM Connect:Direct/Plex environment. It also directs the DTF to read its local initialization parameters from the file specified in the //CDPLEX DD statement in the startup JCL.

The CDPLEX.MAXSERVER parameter specifies the maximum number of servers the Connect:Direct/Plex Manager will manage. For more information, see *Storage Requirements in a IBM Connect:Direct Environment* in the *IBM Connect:Direct for z/OS Configuration Guide*.

The XCF.NAME parameter specifies a unique name used by the z/OS Cross Systems Communication Facility (XCF) to assist communications among IBM Connect:Direct/Plex members. This name

indicates that the IBM Connect:Direct Manager and IBM Connect:Direct Servers are part of the same XCF group.

5. Add the following optional parameters to the CDPLX member in \$CD.PLEX.INITPARM.
 - CDPLEX.TIMER specifies the time-out value for XCF communications in minutes.
 - CDPLEX.WLM.GOAL specifies whether IBM Workload Manager (WLM) Goal Mode queries are made. This parameter is optional.
6. Create a local initialization parameters file for the IBM Connect:Direct Manager:
 - a) Copy the MANAGER sample local initialization parameters member from the IBM Connect:Direct installation \$CD.SDGAPARM(DGAIPMGR) into \$CD.PLEX.INITPARM.
 - b) Change the TCP.LISTEN parameter of the MANAGER member to specify the TCP/IP stack address used by the IBM Connect:Direct Manager.

```
CDPLEX.MANAGER=YES
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
CDPLEX.SERVER.JOBDSN=$CD.PLEX.JCL
CDPLEX.SERVER.JOBMEM=((CDSRV1,SERVER1), -
                      (CDSRV2,SERVER2))
```

7. Create a local initialization parameters file for IBM Connect:Direct SERVER1:
 - a) Copy the sample local initialization member \$CD.SDGAPARM(DGAISRV1), into \$CD.PLEX.INITPARM as member SERVER1.
 - b) Change the CDPLEX.VTAM parameter in the SERVER1 member as follows:
 - Replace the applid11 value with the VTAM APPLID used by this IBM Connect:Direct Server for SNA (Node to Node/PROCESS use, as opposed to API use).
 - Replace the applid12 value with the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex environment and cannot be the same as those specified in the network map.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER1
CDPLEX.VTAM=(applid11,applid12)
CDPLEX.PLEXCLASSES=(TAPE,*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
UDT33.LISTEN=(nnn.nnn.nnn.nnn,port)
```

The TCP.LISTEN and UDT33.LISTEN initialization parameters specify the TCP and UDT listen address and port number combinations. Use a different listen port number than the one used in the existing initialization parameters file.

Note: The CDPLEX.PLEXCLASSES parameter in SERVER1 specifies a 'TAPE' PLEXCLASS. For Processes that require tape drives, specify the 'TAPE' PLEXCLASS in their Process definitions. These Processes run on SERVER1. (See the chapters about building Processes and controlling Processes in the TCQ in *IBM Connect:Direct for z/OS User Guide* for more information on using PLEXCLASS in a Process.)

8. Create a local initialization parameters file for IBM Connect:Direct SERVER2:
 - a) Copy the sample local initialization member \$CD.SDGAPARM(DGAISRV2), into \$CD.PLEX.INITPARM as member SERVER2.
 - b) Change the CDPLEX.VTAM parameter in the SERVER2 member as follows:
 - Replace the applid21 value with the VTAM APPLID used by this IBM Connect:Direct Server for SNA copy Processes.
 - Replace the applid22 value with the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex environment and cannot be the same as those specified in the network map.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER2
CDPLEX.VTAM=(applid21,applid22)
CDPLEX.PLEXCLASSES=(*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
UDT33.LISTEN=(nnn.nnn.nnn.nnn,port)
```

9. Add the CDPLEX DD statement in the following example to the CDMGR member in \$CD.PLEX.JCL.

This statement directs the startup JCL to the global initialization parameters file.

```
//DTF EXEC DGADINIT,
// PARM='$CD.PLEX.INITPARM(CDPLX)'
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(MANAGER)
```

10. Create CDSRV1 in \$CD.PLEX.JCL and copy the CDMGR member into it.

11. Make the following changes to the CDSRV1 member:

- Change the job name so that this job can run simultaneously with the CDMGR JCL.
- Change the member name in the CDPLEX DD statement to SERVER1, as follows. This change directs the CDSRV1 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER1)
```

12. Create a CDSRV2 member and copy CDSRV1 into it.

13. Make the following changes to the CDSRV2 member:

- Change the job name so that this job can run simultaneously with the CDMGR JCL and CDSRV1 JCL.
- Change the member name in the CDPLEX DD statement to SERVER2, as follows. This change directs the CDSRV2 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER2)
```

Note: Route jobs to a different z/OS image by specifying the local node name of the other system in an XEQ statement in the IBM Connect:Direct Manager or Server startup JCL, as follows:

```
/*XEQ njenode
[JES2]
/*ROUTE XEQ njenode [JES3]
```

This example routes the job to the z/OS image identified by the local node name NJENODE.

14. Submit the CDMGR JCL to bring up the IBM Connect:Direct/Plex server.

After the IBM Connect:Direct Manager initializes, it submits the CDSRV1 JCL and CDSRV2 JCL to bring up the two IBM Connect:Direct Servers.

15. After IBM Connect:Direct Manager initializes, use the IUI to signon.

You can then submit Processes and perform other functions through the IUI.

Advanced IBM Connect:Direct:/Plex Configuration Considerations

Examples of complex configurations include:

- [Converting an Existing Stand-Alone Server to a Plex Environment](#)
- [Converting Two Existing Stand-Alone Server Systems to a Plex Environment](#)

Before attempting a complex configuration, be aware of the following issues.

IBM Connect:Direct/Plex System File Considerations

All IBM Connect:Direct/Plex members share a single set of IBM Connect:Direct system files. If you combine multiple existing IBM Connect:Direct systems into one IBM Connect:Direct/Plex environment, you may need to merge some IBM Connect:Direct system files from the individual systems.

Do not merge system files that are listed in the following table.

File	Comment
CKPT file	You cannot merge the CKPT files from multiple IBM Connect:Direct images. You must either: <ul style="list-style-type: none">• Define a new CKPT file using the DGAXCKPD JCL in the \$CD.SDGASAMP library, or• Use the CKPT file from one of the existing IBM Connect:Direct systems. However, the existing CKPT file size may not be sufficient for a IBM Connect:Direct/Plex environment.
Message file	You can use any existing IBM Connect:Direct Message file. You do not need to combine Message files from the individual IBM Connect:Direct systems.
Statistics files	You cannot merge statistics files from multiple IBM Connect:Direct images. You can reference them using a IBM Connect:Direct/Plex environment as archived statistic files. You can create new statistics file pairs by using the DGAXSTAD JCL in the \$CD.SDGASAMP library.
TCQ and TCX files	You cannot merge TCQ and TCX files from multiple IBM Connect:Direct images. You must either: <ul style="list-style-type: none">• Define new TCQ and TCX files using the DGAXTCQD JCL in the \$CD.SDGASAMP library, or• Use the TCQ and TCX files from one of the existing IBM Connect:Direct systems. However, the existing TCQ and TCX file sizes may not be sufficient for a IBM Connect:Direct/Plex environment.

The following table lists system files that you need to merge.

File	Comment
AUTH file	If unique entries exist in the existing IBM Connect:Direct systems' AUTH files: <ol style="list-style-type: none">1. Define a new AUTH file using the DGAXAUTD JCL found in the \$CD.SDGASAMP library.2. Copy the existing AUTH files into the new AUTH file using the DGAXAUTC JCL found in the \$CD.SDGASAMP library. PLEXAUTC is an IDCAMS REPRO that specifies NOREPLACE. If any duplicate records exist, only the first one is saved. If unique entries do not exist, use one of the existing AUTH files.

File	Comment
NETMAP file	<p>You need to create a new network map source file.</p> <p>The new network map source file uses information from the existing systems' network maps. (If the existing IBM Connect:Direct systems' network map source is not available, create the source files by performing network map unloads for the existing systems' network map files. See Unloading the Network Map to the Source Format for more information.)</p> <p>To create a new network map source file:</p> <ol style="list-style-type: none"> 1. Copy the network map source from an existing IBM Connect:Direct system as NETMAPLX. 2. Copy the remote definitions from all other existing network map source files into NETMAPLX. 3. Remove all duplicate entries. 4. Define the new network map file using the DGAXNETD JCL found in the \$CD.SDGASAMP library. 5. Load the new network map file using the DGAXNETL JCL found in the \$CD.SDGASAMP library. 6. Check the output from network map load and correct any errors. 7. Rerun the network map load if necessary.
TYPE file	<p>If unique entries exist in the existing IBM Connect:Direct systems' TYPE files:</p> <ol style="list-style-type: none"> 1. Define a new TYPE file using the DGAXTYPD JCL found in the \$CD.SDGASAMP library. 2. Copy the existing TYPE files into the new TYPE file using the DGAXTYPC JCL found in the \$CD.SDGASAMP library. <p>The PLEXTYPC is an IDCAMS REPRO that specifies NOREPLACE. If any duplicate records exist, only the first one is saved.</p> <p>If unique entries do not exist, use one of the existing TYPE files.</p>

Local Node Naming Considerations

The network map contains the local node name for the IBM Connect:Direct/Plex environment. The node name used in the network map varies according to the type of configuration:

- Installing a new IBM Connect:Direct/Plex environment

If you are installing a new IBM Connect:Direct/Plex environment, you must create a new local node name. You must provide the new node name, along with the APPLID and/or TCP/IP address and port number to all IBM Connect:Direct partner nodes. The partner Nodes must provide you the same information for use in your local network map.

- Replacing an existing IBM Connect:Direct system with a IBM Connect:Direct/Plex environment

If you are replacing an existing IBM Connect:Direct system with a IBM Connect:Direct/Plex environment, use the existing system node name as the IBM Connect:Direct/Plex local node name, you must provide new APPLIDs for the IBM Connect:Direct Manager. One IBM Connect:Direct Server uses the existing APPLIDs in its local initialization parameters. Any additional servers require new APPLIDs.

- Replacing multiple existing IBM Connect:Direct systems with a IBM Connect:Direct/Plex environment

If you are replacing multiple IBM Connect:Direct systems with a IBM Connect:Direct/Plex environment, create a new local node name for the IBM Connect:Direct/Plex. Use the existing node names in the CDPLEX.SERVER.NODE initialization parameter of the IBM Connect:Direct Server. All adjacent node entries in the network map must include USE.SERVER.NODE=YES. You must provide new APPLIDs for

the IBM Connect:Direct Manager. Each IBM Connect:Direct Server uses the existing APPLID from its corresponding IBM Connect:Direct Stand-alone Server image in its local initialization parameters.

Refer to the setup procedure in “[Converting Two Existing Stand-Alone Server Systems to a Plex Environment](#)” on page 293 for more details.

Strategies for Communicating with Non-Plex Servers

A IBM Connect:Direct/Plex environment can perform workload balancing among the IBM Connect:Direct Servers. However, if the IBM Connect:Direct/Plex environment communicates with an external non-IBM Connect:Direct/Plex system, the other IBM Connect:Direct system may have problems with Processes from the same IBM Connect:Direct adjacent node, but with a different VTAM APPLID or TCP/IP address than specified in their network map.

IBM Connect:Direct/Plex offers three ways of avoiding this problem.

Using Alternate Communication Paths to Define the IBM Connect:Direct/Plex

The advantages to this approach are you only have to define one entity, the IBM Connect:Direct/Plex, and then copy that same definition to the network maps of the non-Plex Servers and you can still use the NETMAP-checking feature.

About this task

To define a IBM Connect:Direct/Plex that has servers running on several hosts, you can use the ALT.COMM parameter in the network map of each non-Plex Server that will communicate with the IBM Connect:Direct/Plex.

Procedure

1. Specify USE.SERVER.NODE=NO in the IBM Connect:Direct/Plex network map entry of each non-Plex Server so that all servers in the IBM Connect:Direct/Plex environment appear as one node.
2. Define the IBM Connect:Direct/Plex node as an adjacent node with all possible IP addresses of the hosts that the Manager can run on specified using the ALT.COMM definition.

The following is an example of the ALT.COMM parameter:

```
ADJACENT.NODE=( -  
  (CDMGR,1366,10.1.1.1,TCP,EXT,BOTH) -  
  ENVIRONMENT=ZOS -  
  PARSESS=(00000010 00000002) -  
  ALT.COMM=(ALT.DIR=TOP  
    (ALT.ADDR=10.1.1.2,ALT.PORT=1366,-  
      ALT.TYPE=TCP , ALT.USE.OUT=NO )  
    (ALT.ADDR=10.1.1.3,ALT.PORT=1366,-  
      ALT.TYPE=TCP , ALT.USE.OUT=NO )  
    (ALT.ADDR=10.1.1.4,ALT.PORT=1366,-  
      ALT.TYPE=TCP , ALT.USE.OUT=NO )) -
```

3. Copy this ALT.COMM definition and put it in the network map of each non-Plex Server that will communicate with the IBM Connect:Direct/Plex.

Forcing All Processes to One IBM Connect:Direct Server

This approach does not take advantage of IBM Connect:Direct/Plex workload balancing. To direct all Processes between the IBM Connect:Direct/Plex environment and the external IBM Connect:Direct to one IBM Connect:Direct Server, follow this procedure.

Procedure

1. Specify a default PLEXCLASS parameter in the IBM Connect:Direct/Plex network map adjacent node entry for the external IBM Connect:Direct system.

2. Specify that PLEXCLASS parameter in only one local initialization parameter of the IBM Connect:Direct Server.
3. Specify the VTAM APPLID or TCP/IP address of the specific IBM Connect:Direct Server all Processes are being forced to in the network map of the external non-Plex IBM Connect:Direct system.

Define a Unique Node Name for Each IBM Connect:Direct Server

By defining a node name for each IBM Connect:Direct Server, the IBM Connect:Direct/Plex environment can initiate Processes to the external IBM Connect:Direct system through any available IBM Connect:Direct/Plex server. For each IBM Connect:Direct server, you must define a USE.SERVER.NODE network map parameter and a CDPLEX.SERVER.NODE initialization parameter.

To avoid making your IBM Connect:Direct/Plex configuration more complex than necessary, use the USE.SERVER.NODE and CDPLEX.SERVER.NODE parameters only if your system meets all of the following conditions:

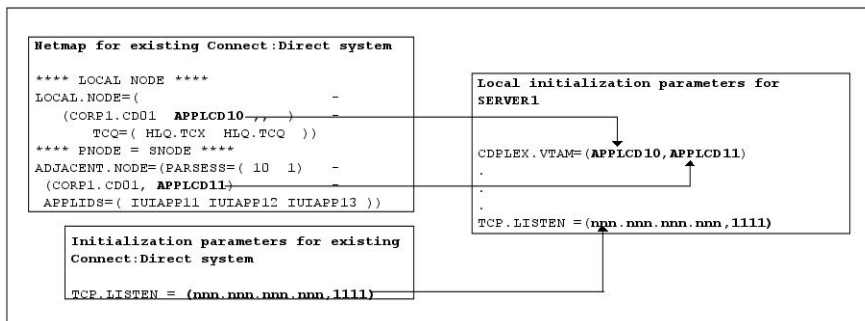
- The external IBM Connect:Direct system must connect to two or more IBM Connect:Direct Servers in the IBM Connect:Direct/Plex environment.
- The external IBM Connect:Direct system uses network map checking.
- The external IBM Connect:Direct system has non-Plex servers, which cannot communicate directly with the IBM Connect:Direct Manager.

The disadvantages of this approach are you must manually direct Processes initiated by the external IBM Connect:Direct system to each IBM Connect:Direct Server and you may need to create additional network map entries for remote systems.

Converting an Existing Stand-Alone Server to a Plex Environment

You can convert an existing production IBM Connect:Direct Stand-alone Server into a IBM Connect:Direct/Plex environment with two servers. This configuration takes advantage of the IBM Connect:Direct/Plex single image and workload balancing capability for Processes initiated by this IBM Connect:Direct/Plex environment. This configuration also supports external IBM Connect:Direct systems without requiring any changes to the external systems.

The following illustration shows how the network map and initialization parameter values from the existing stand-alone IBM Connect:Direct system map to the new IBM Connect:Direct/Plex environment for this configuration.



The setup example assumes the following:

- You are currently running a production IBM Connect:Direct Stand-alone Server.
- The global and local initialization parameter files for the IBM Connect:Direct/Plex environment are located in \$CD.PLEX.INITPARM (a PDS that you set up for this purpose). The JCL to bring up the IBM Connect:Direct/Plex environment is located in \$CD.PLEX.JCL (a PDS that you set up for this purpose).

You can either allocate these data sets or use existing data sets in their place.

- The IBM Connect:Direct/Plex environment identifies itself to external systems with the same node name as the production IBM Connect:Direct Stand-alone Server.

Converting an Existing Production Server to a Plex Environment

To convert an existing production IBM Connect:Direct Stand-alone Server into a IBM Connect:Direct/Plex environment:

Procedure

1. Copy the existing INITPARMs member into \$CD.PLEX.INITPARM as member CDPLX.
2. Copy the existing IBM Connect:Direct Stand-alone Server startup JCL into \$CD.PLEX.JCL as member CDMGR.
3. Change the network map source to specify new APPLIDs for the LOCAL.NODE and the PNODE/SNODE ADJACENT.NODE.

The existing APPLIDs are used for the IBM Connect:Direct SERVER1, which means that you do not need to change the connections to external IBM Connect:Direct systems.

4. Add the following initialization parameters to the CDPLX member in \$CD.PLEX.INITPARM. This member becomes the IBM Connect:Direct/Plex global initialization parameters file.

```
CDPLEX=YES  
CDPLEX.MAXSERVER = number of servers | 4  
XCF.NAME=8-char-name
```

The CDPLEX=YES parameter indicates a IBM Connect:Direct/Plex environment. It also directs the DTF to read its local initialization parameters from the file specified in the //CDPLEX DD statement in the startup JCL.

The CDPLEX.MAXSERVER parameter specifies the maximum number of servers the Connect:Direct/Plex Manager will manage. For more information, see *Storage Requirements in a IBM Connect:Direct Environment* in the *IBM Connect:Direct for z/OS Configuration Guide*.

The XCF.NAME parameter specifies a unique name used by the z/OS Cross Systems Communication Facility (XCF) to assist communications among IBM Connect:Direct/Plex members. This name indicates that the IBM Connect:Direct Manager and IBM Connect:Direct Servers are part of the same XCF group.

5. You can add the following optional parameters to the CDPLX member in \$CD.PLEX.INITPARM.
 - “CDPLEX.TIMER = 5 | number of minutes” on page 449 specifies the time-out value for XCF communications in minutes.
 - “CDPLEX.WLM.GOAL = (NO | YES, exitname)” on page 449 specifies whether IBM Workload Manager (WLM) Goal Mode queries are made. This parameter is optional.
6. Create the local initialization parameter files for each IBM Connect:Direct/Plex member:
 - a) Copy the DGAITMGR, DGAISRV1, and DGAISRV2 sample local initialization parameters members from the IBM Connect:Direct \$CD.SDGAPARM library into \$CD.PLEX.INITPARM as members MANAGER, SERVER1, and SERVER2.
 - b) Change the TCP.LISTEN parameter (following in bold) to specify the TCP/IP stack address that is used by the IBM Connect:Direct Manager.

You do not need to change any other parameters in the MANAGER member.

```
CDPLEX.MANAGER=YES  
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)  
UDT33.LISTEN=(nnn.nnn.nnn.nnn,port)  
CDPLEX.SERVER.JOBDSN=$CD.PLEX.JCL  
CDPLEX.SERVER.JOBMEM=( (CDSRV1,SERVER1), -  
                        (CDSRV2,SERVER2))
```

- c) Change the CDPLEX.VTAM parameter in the SERVER1 member as follows:

- Replace the applid11 value with the VTAM APPLID that is defined in the existing network map for SNA copy Processes.
- Replace the applid12 value with the VTAM APPLID that is defined in the existing network map for the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex environment and cannot be the same as specified in the new network map.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER1
CDPLEX.VTAM=(applid11,applid12)
CDPLEX.PLEXCLASSES=(TAPE,*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
UDT33.LISTEN=(nnn.nnn.nnn.nnn,port)
```

- d) The TCP.LISTEN and UDT33.LISTEN initialization parameters specify the TCP and UDT listen address and port number combinations. Use a different listen port number than the one used in the existing initialization parameters file.

Note: The CDPLEX.PLEXCLASSES parameter in SERVER1 specifies a 'TAPE' PLEXCLASS. For Processes that require tape drives, specify the 'TAPE' PLEXCLASS in their Process definitions. These Processes run on SERVER1. (See *Building, Modifying, and Submitting Processes* in the *IBM Connect:Direct for z/OS User Guide* for more information on using PLEXCLASS in a Process.)

- e) Change the CDPLEX.VTAM parameter in the SERVER2 member as follows:

- Replace the applid21 value with a new VTAM APPLID you have defined for SNA copy Processes.
- Replace the applid22 value with a new VTAM APPLID you have defined for the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex environment and cannot be the same as specified in the new network map or used for SERVER1.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER1
CDPLEX.VTAM=(applid21,applid22)
CDPLEX.PLEXCLASSES=(*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
UDT33.LISTEN=(nnn.nnn.nnn.nnn,port)
```

- f) The TCP.LISTEN and UDT33.LISTEN initialization parameters specify the TCP and UDT listen address and port number combinations. Use a different listen port number than the one used in the existing initialization parameters file.

7. Add the CDPLEX DD statement in the following example to the CDMGR member in \$CD.PLEX.JCL (this JCL is the startup JCL copied earlier).

```
//DTF EXEC DGADINIT,
// PARM='$CD.PLEX.INITPARM(CDPLX)'
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(MANAGER)
```

8. Create the CDSRV1 member and copy CDMGR into it.

9. Make the following changes to the CDSRV1 member:

- Change the job name so that this job can run simultaneously with the CDMGR JCL.
- Change the member name in the CDPLEX DD statement to SERVER1, as follows. This change directs the CDSRV1 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER1)
```

10. Create the CDSRV2 member and copy CDSRV1 into it.

11. Make the following changes to the CDSRV2 member:

- a) Change the job name so that this job can run simultaneously with the CDMGR JCL and CDSRV1 JCL.
- b) Change the member name in the CDPLEX DD statement to SERVER2, as follows. This change directs the CDSRV2 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER2)
```

Note: You can route jobs to a different z/OS image by specifying the local node name of the other system name in an XEQ statement in the IBM Connect:Direct Manager or Server JCL as follows:

```
/*XEQ njenode
[JES2]
//*ROUTE XEQ njenode [JES3]
```

This example routes the job to the z/OS image identified by the local node name NJENODE.

12. Submit the CDMGR JCL to bring up IBM Connect:Direct/Plex.

After the IBM Connect:Direct Manager initializes, it submits the CDSRV1 JCL and CDSRV2 JCL to bring up the two IBM Connect:Direct Servers.

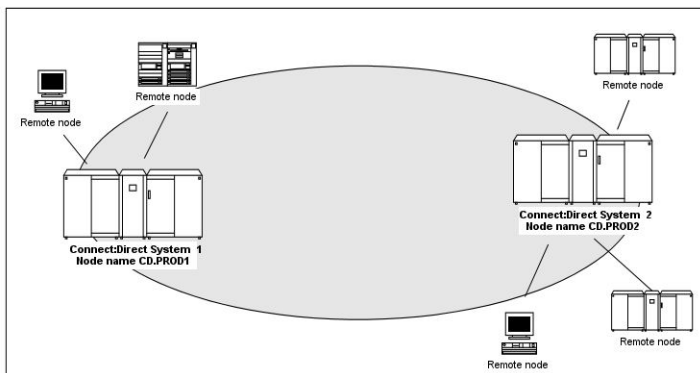
13. After the IBM Connect:Direct Manager initializes, use the IUI to signon to the IBM Connect:Direct Manager.

You can then submit Processes and perform other functions through the IUI.

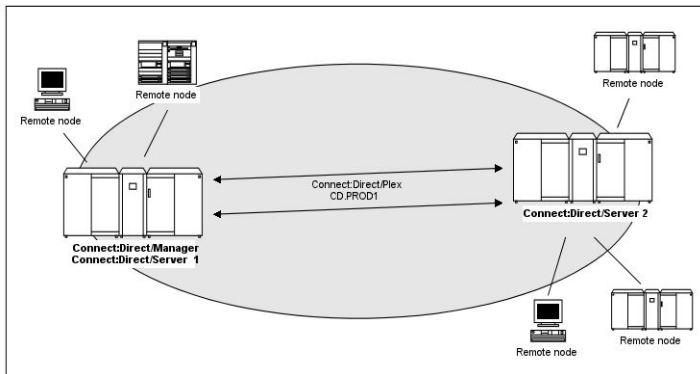
Converting Two Existing Stand-Alone Server Systems to a Plex Environment

This section illustrates how to convert existing production IBM Connect:Direct Stand-alone Server systems into a IBM Connect:Direct/Plex environment with two servers. This configuration takes advantage of the IBM Connect:Direct/Plex single image and workload balancing capability for Processes initiated by this IBM Connect:Direct/Plex environment. This configuration also supports external IBM Connect:Direct systems without requiring any changes to the external systems.

In the following illustration, two separate IBM Connect:Direct Stand-alone Server systems run in a z/OS sysplex environment.



The procedure in this section combines the separate systems into the single IBM Connect:Direct/Plex environment as follows.

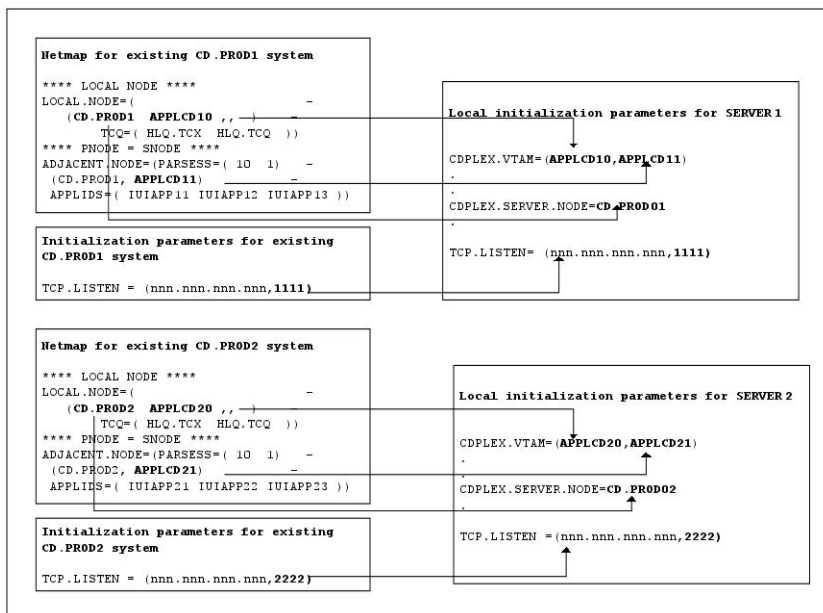


In this configuration, the original CD.PROD1 system becomes the IBM Connect:Direct Manager and IBM Connect:Direct Server1. The CD.PROD2 system becomes the IBM Connect:Direct Server2. The IBM Connect:Direct/Plex environment is given the node name CD.PROD1. No changes are made to the remote nodes' network maps. The remote nodes communicate with the IBM Connect:Direct/Plex environment as if they communicated with a single IBM Connect:Direct image.

To create this configuration you must:

- Define new APPLIDs for the IBM Connect:Direct/Plex environment
- Use the APPLIDs from the existing CD.PROD1 system in the Server1 local initialization parameters
- Use the APPLIDs from the existing CD.PROD2 system in the Server2 local initialization parameters

The following illustration shows how the network map and initialization parameter values from the existing IBM Connect:Direct systems map to the IBM Connect:Direct/Plex environment.



This example assumes the following:

- You are running two production IBM Connect:Direct Stand-alone Server systems: CD.PROD1 and CD.PROD2.
- The global and local initialization parameter files are located in \$CD.PLEX.INITPARM. The JCL that brings up the IBM Connect:Direct/Plex environment is located in \$CD.PLEX.JCL. You can either allocate these data sets or use existing data sets in their place.
- The IBM Connect:Direct/Plex environment identifies itself to external systems with a new node name. Each IBM Connect:Direct Server identifies itself to external systems with the same node name it used as a IBM Connect:Direct Stand-alone Server.

Converting a Standalone Server System into One IBM Connect:Direct/Plex Setup

To convert two existing IBM Connect:Direct Stand-alone Server systems into one IBM Connect:Direct/Plex environment:

Procedure

1. Copy the the INITPARMs member for CD.PROD1 into \$CD.PLEX.INITPARM as member CDPLX.
2. Resolve any differences (other than data set names) between the new INITPARMS member and the CD.PROD2 INITPARMS member.
3. Copy the CD.PROD1 IBM Connect:Direct/Plex startup JCL into \$CD.PLEX.JCL as member CDMGR.
4. Resolve any differences, such as trace DDs, with the CD.PROD2 startup JCL.
5. Merge the existing AUTH and TYPE files from both systems as described in [IBM Connect:Direct/Plex System File Considerations](#), using the same files names used for CD.PROD1.
6. Create new TCQ/TCX, CKPT, and statistics file pairs as discussed in [CKPT file](#), using the same file names used for CD.PROD1.
7. Merge the source from the individual network map files as described in [NETMAP file](#).
8. Change the new network map source (created from the merged network map files) as follows:
 - a) Specify new APPLIDs and a new LOCAL.NODE name for the LOCAL.NODE and the PNODE/SNODE ADJACENT.NODE.

Because the existing APPLIDs are used for SERVER1, you do not need to change the external IBM Connect:Direct connections.
 - b) Specify USE.SERVER.NODE=YES on all ADJACENT.NODE records.
 - c) Use the same local node name that is used for CD.PROD1.
 - d) Load the network map.
9. Add the following initialization parameters to the CDPLX member in \$CD.PLEX.INITPARM. This member becomes the IBM Connect:Direct/Plex global initialization parameters file.

```
CDPLEX=YES  
CDPLEX.MAXSERVER = number of servers | 4  
XCF.NAME=8-char-name
```

The CDPLEX=YES parameter indicates a IBM Connect:Direct/Plex environment. It also directs the DTF to read its local initialization parameters from the file specified in the //CDPLEX DD statement in the startup JCL.

The CDPLEX.MAXSERVER parameter specifies the maximum number of servers the Connect:Direct/Plex Manager will manage. For more information, see *Storage Requirements in a IBM Connect:Direct Environment* in the *IBM Connect:Direct for z/OS Configuration Guide*.

The XCF.NAME parameter specifies a unique name used by the z/OS XCF to assist communications among IBM Connect:Direct/Plex members. This name indicates that the IBM Connect:Direct Manager and IBM Connect:Direct Servers are part of the same XCF group.

10. You can add the following optional parameters to the CDPLX member in \$CD.PLEX.INITPARM:
 - “CDPLEX.TIMER = 5 | number of minutes” on page 449 specifies the time-out value for XCF communications in minutes.
 - “CDPLEX.WLM.GOAL = (NO | YES, exitname)” on page 449 specifies whether IBM Workload Manager (WLM) Goal Mode queries are made. This parameter is optional.
11. Create the local initialization parameters files for each IBM Connect:Direct/Plex member:
 - a) Copy the DGAITMGR, DGAISRV1, and DGAISRV2 sample local initialization parameters members from the IBM Connect:Direct \$CD.SDGAPARM library into \$CD.PLEX.SDGAPARM as members MANAGER, SERVER1, and SERVER2.

- b) Change the TCP.LISTEN parameter of the MANAGER member to specify the TCP/IP stack address used by the IBM Connect:Direct/Plex Manager.

You need not change any other parameters need in the MANAGER member.

```
CDPLEX.MANAGER=YES
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
CDPLEX.SERVER.JOBDSN=$CD.PLEX.JCL
CDPLEX.SERVER.JOBMEM=((CDSRV1,SERVER1), -
                      (CDSRV2,SERVER2))
```

- c) Change the CDPLEX.VTAM parameter in the SERVER1 member as follows:

- Replace the applid11 value with the VTAM APPLID from CD.PROD1 for SNA copy Processes.
- Replace the applid12 value with the VTAM APPLID from CD.PROD1 for the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex and cannot be the same as those specified in the new network map.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER1
CDPLEX.VTAM=(applid11,applid12,applid13)
CDPLEX.PLEXCLASSES=(TAPE,*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
```

- d) Change the TCP.LISTEN parameter in the SERVER1 member to specify the TCP/IP stack address used by this IBM Connect:Direct/Plex server.
- e) Change the TCP.LISTEN parameter in SERVER1 member to the TCP.LISTEN value from CD.PROD1.

Note: The CDPLEX.PLEXCLASSES parameter in SERVER1 specifies a 'TAPE' PLEXCLASS. For Processes that require tape drives, specify the 'TAPE' PLEXCLASS in their Process definitions. These Processes run on SERVER1. (See *Building, Modifying, and Submitting Processes in IBM Connect:Direct for z/OS User Guide* for more information on using PLEXCLASS in a Process.)

- f) Add the following statement to the SERVER1 member.

```
CDPLEX.SERVER.NODE=CD.PROD1
```

- g) Change the CDPLEX.VTAM parameter in the SERVER2 member as follows:

- Replace the applid21 value with the VTAM APPLID from CD.PROD2 for SNA copy Processes.
- Replace the applid22 value with the VTAM APPLID from CD.PROD2 for the PNODE-SNODE APPLID.

These APPLIDs must be unique across the IBM Connect:Direct/Plex and cannot be the same as those specified in the new network map or used for SERVER1.

```
CDPLEX.MANAGER=NO
CDPLEX.SERVER=SERVER2
CDPLEX.VTAM=(applid21,applid22)
CDPLEX.PLEXCLASSES=(*)
TCP.LISTEN=(nnn.nnn.nnn.nnn,port)
```

- h) Change the TCP.LISTEN parameter in the SERVER2 member to specify the TCP/IP stack address used by this IBM Connect:Direct/Plex server.
- i) Change the TCP.LISTEN parameter in the SERVER2 member to the TCP.LISTEN value from CD.PROD2.
- j) Add the following statement to the SERVER2 member.

```
CDPLEX.SERVER.NODE=CD.PROD2
```

12. Add the CDPLEX DD statement in the following example to the CDMGR member in \$CD.PLEX.JCL. (This JCL is the startup JCL copied earlier.)

```
//DTF EXEC DGADINIT,  
// PARM='$CD.PLEX.INITPARM(CDPLX) '  
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(MANAGER)
```

13. Create the CDSRV1 member and copy CDMGR into it.
14. Make the following changes to the CDSRV1 member:
- a) Change the job name so that this job can run simultaneously with the CDMGR JCL.
 - b) Change the member name in the CDPLEX DD statement to SERVER1 as follows. This directs the CDSRV1 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER1)
```

15. Create the CDSRV2 member and copy CDSRV1 into it.
16. Make the following changes to the CDSRV2 member:
- a) Change the job name so that this job can run simultaneously with the CDMGR JCL and CDSRV1 JCL.
 - b) Change the member name in the CDPLEX DD statement to SERVER2 as follows. This directs the CDSRV2 JCL to its local initialization parameters file.

```
//CDPLEX DD DISP=SHR,DSN=$CD.PLEX.INITPARM(SERVER2)
```

Note: You can route jobs to a different z/OS image by specifying the local node name of the other system name in an XEQ statement in the IBM Connect:Direct Manager or Server JCL as follows:

```
/*XEQ njenode  
[JES2]  
/*ROUTE XEQ njenode [JES3]
```

This example routes the job to the z/OS image identified by the local node name NJENODE.

17. Submit the CDMGR JCL to bring up the IBM Connect:Direct/Plex.

After the IBM Connect:Direct Manager initializes, it submits the CDSRV1 JCL and CDSRV2 JCL to bring up the two IBM Connect:Direct Servers.

18. After the IBM Connect:Direct Manager initializes, use the IUI to signon to the IBM Connect:Direct Manager.

You can then submit Processes and perform other functions through the IUI.

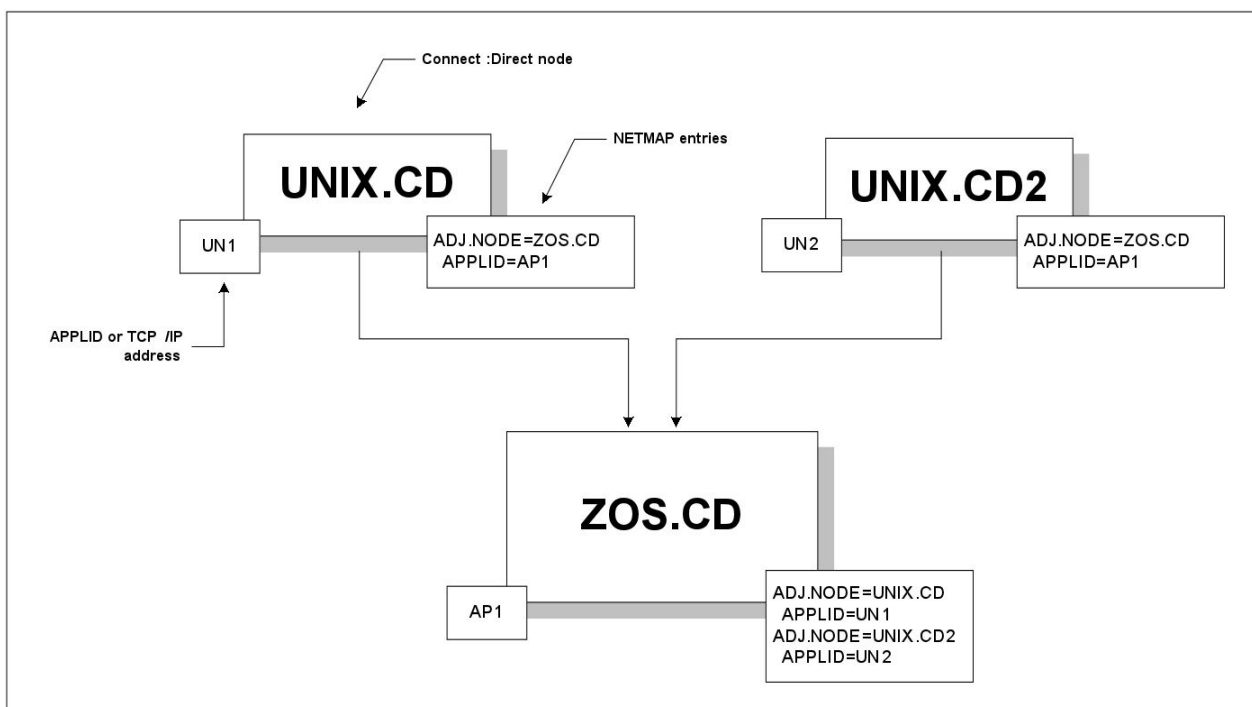
Additional IBM Connect:Direct/Plex Configuration Examples

This topic illustrates additional IBM Connect:Direct/Plex configuration examples. While your site configuration can vary due to the number of external nodes, use this topic as a guide in determining the best way to configure a IBM Connect:Direct/Plex.

Note: The examples in this section are high-level descriptions for use as a configuration model. They do not describe all configuration changes to set up a IBM Connect:Direct/Plex. See [A New IBM Connect:Direct/Plex Environment](#) for detailed information.

Configuration Examples Using One Connect:Direct for z/OS System

This section assumes the following IBM Connect:Direct environment exists.



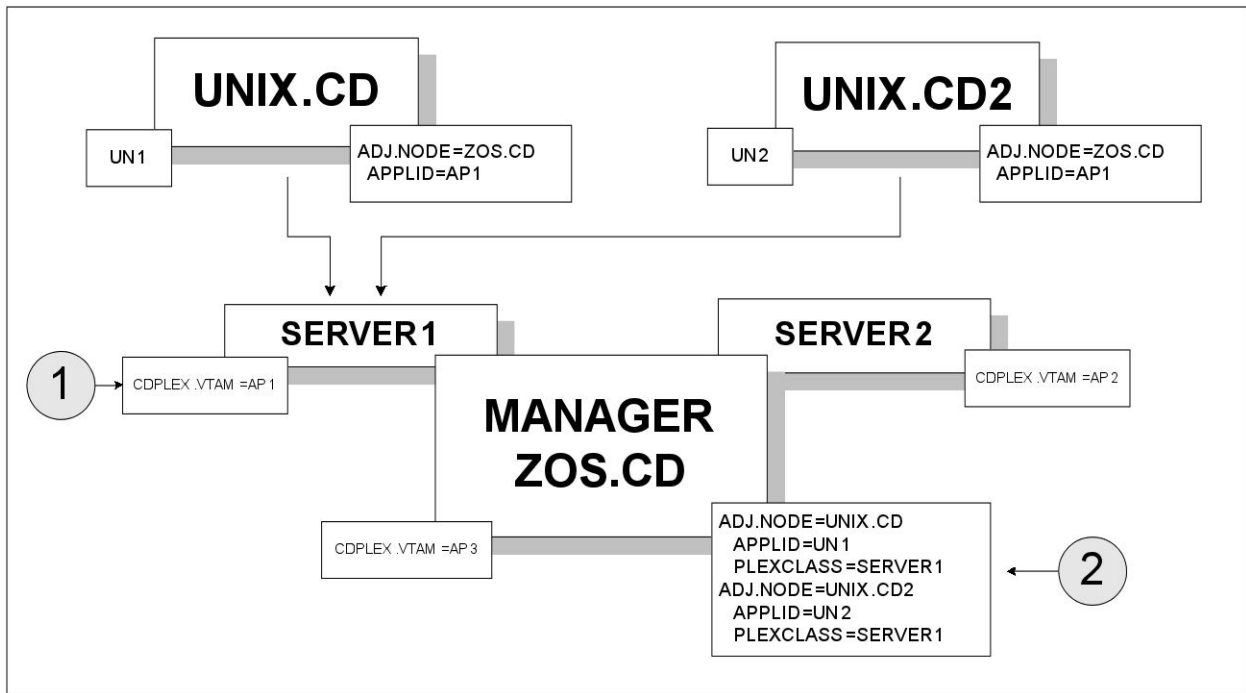
In this environment, two Connect:Direct for UNIX nodes (UNIX.CD and UNIX.CD2) communicate with a Connect:Direct for z/OS system (ZOS.CD). The ZOS.CD system uses the APPLID “AP1.” The UNIX.CD system uses the APPLID “UN1”, while the UNIX.CD2 system uses the APPLID “UN2.” The network map entries define the adjacent nodes.

Although this example uses Connect:Direct for UNIX as the external nodes, the relationship is the same when any IBM Connect:Direct platform is an external node.

Scenario 1 – External Nodes Communicate with One IBM Connect:Direct Server

This section describes the simplest IBM Connect:Direct/Plex configuration – both external nodes communicate with the same IBM Connect:Direct Server.

In the following illustration, the ZOS.CD system is configured as a IBM Connect:Direct/Plex consisting of a IBM Connect:Direct Manager and two IBM Connect:Direct Servers. Both external Connect:Direct for UNIX systems communicate only with SERVER1.



To accomplish this setup, assign the APPLID from the original IBM Connect:Direct system (AP1) to SERVER1 through the local initialization parameters of SERVER1 (callout 1 in the preceding illustration). Note that you must create new APPLIDs for the SERVER2 (AP2) and the IBM Connect:Direct Manager (AP3).

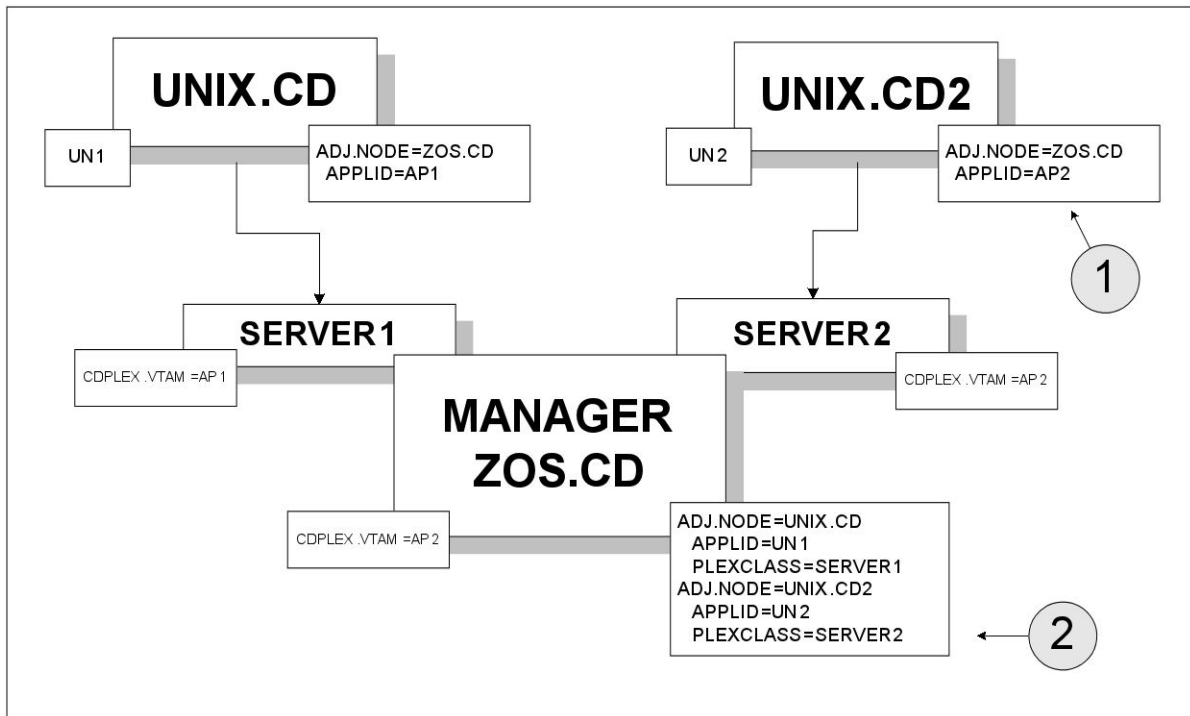
Then, specify SERVER1 as the default PLEXCLASS in the IBM Connect:Direct/Plex network map (callout 2). This routes all work and communication among the nodes through SERVER1.

The advantage of this approach is that the Connect:Direct for UNIX nodes do not need to change any initialization parameter or network map definitions. They do not have any knowledge of the change to the z/OS node.

The disadvantage of this approach is that you cannot use IBM Connect:Direct/Plex workload balancing to its full potential. You cannot perform workload balancing on Processes received from or sent to the external nodes. However, work originating and done entirely on the Connect:Direct for z/OS system can use IBM Connect:Direct/Plex workload balancing.

Scenario 2 – External Nodes Communicate with Individual IBM Connect:Direct Servers

In this configuration, each external IBM Connect:Direct node communicates with a specific IBM Connect:Direct Server. This configuration makes better use of the IBM Connect:Direct/Plex environment by spreading the work from the external nodes between the IBM Connect:Direct Servers.



To accomplish this configuration, change the UNIX.CD2 network map to point to the APPLID for SERVER2 (callout 1).

Then, make the following changes to the IBM Connect:Direct/Plex network map (callout 2):

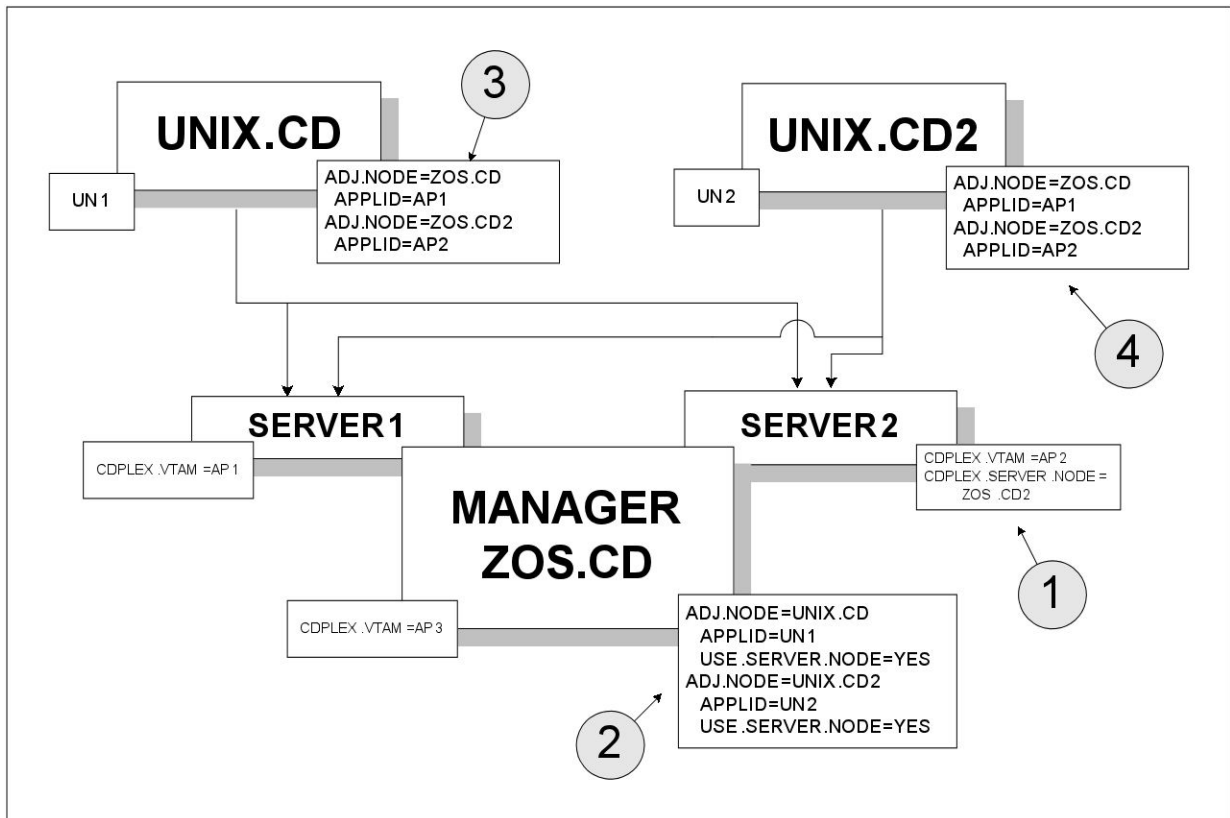
- Route all Processes from UNIX.CD to SERVER1 by defining SERVER1 as the default PLEXCLASS in the adjacent node definition for UNIX.CD.
- Route all Processes from UNIX.CD2 to SERVER2 by defining SERVER2 as the default PLEXCLASS in the adjacent node definition for UNIX.CD2.

The advantage of this approach is that work from each UNIX node runs on a different IBM Connect:Direct Server, so work from one node does not interfere with work from the other. You do not need to change Processes submitted from either node to run on the specified servers (unless the Process itself specifies a TCP/IP address).

The disadvantage of this approach is that you still cannot use the IBM Connect:Direct/Plex workload balancing to its full potential. You cannot perform workload balancing on Processes received from or sent to the external nodes. Work originating and done entirely on the Connect:Direct for z/OS system can use IBM Connect:Direct/Plex workload balancing.

Scenario 3 – External Nodes Communicate with Both IBM Connect:Direct Servers

This configuration uses the IBM Connect:Direct/Plex workload balancing capability. In this environment, both external nodes can communicate with either IBM Connect:Direct Server.



In this configuration, the network map of each Connect:Direct for UNIX node is changed to point to both IBM Connect:Direct Servers. However, because the IBM Connect:Direct/Plex normally is displayed as a single node to external systems, you must first create a unique node name for SERVER2. To create a unique node name, specify:

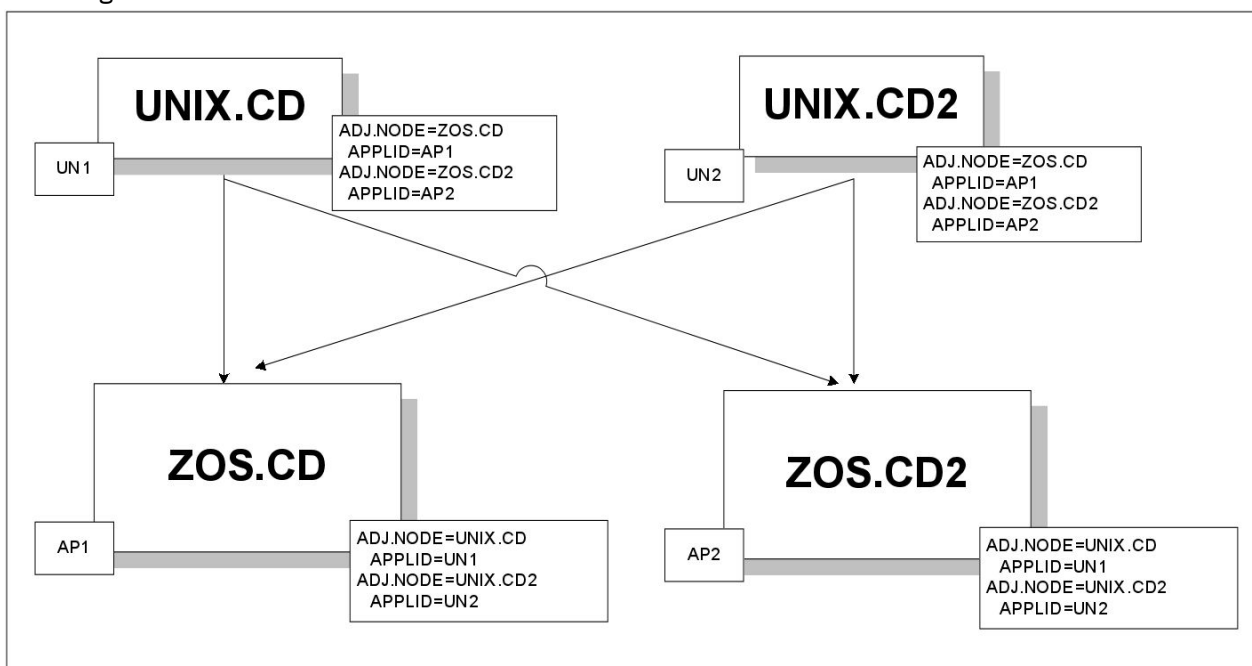
- CDPLEX.SERVER.NODE = ZOS.CD2 in the SERVER2 local initialization parameters (callout 1)
You do not need to specify the CDPLEX.SERVER.NODE initialization parameter for SERVER1 because it uses the local node (ZOS.CD).
- USE.SERVER.NODE=YES in the IBM Connect:Direct/Plex network map adjacent node definitions (callout 2)

The ZOS.CD2 node name is then added to the external nodes' network maps (callouts 3 and 4).

The advantage of this configuration is that you can perform workload balancing on outgoing Processes from the IBM Connect:Direct/Plex. However, you cannot perform automatic workload balancing on Processes received from the external nodes; you must manually balance them by changing the SNODE.

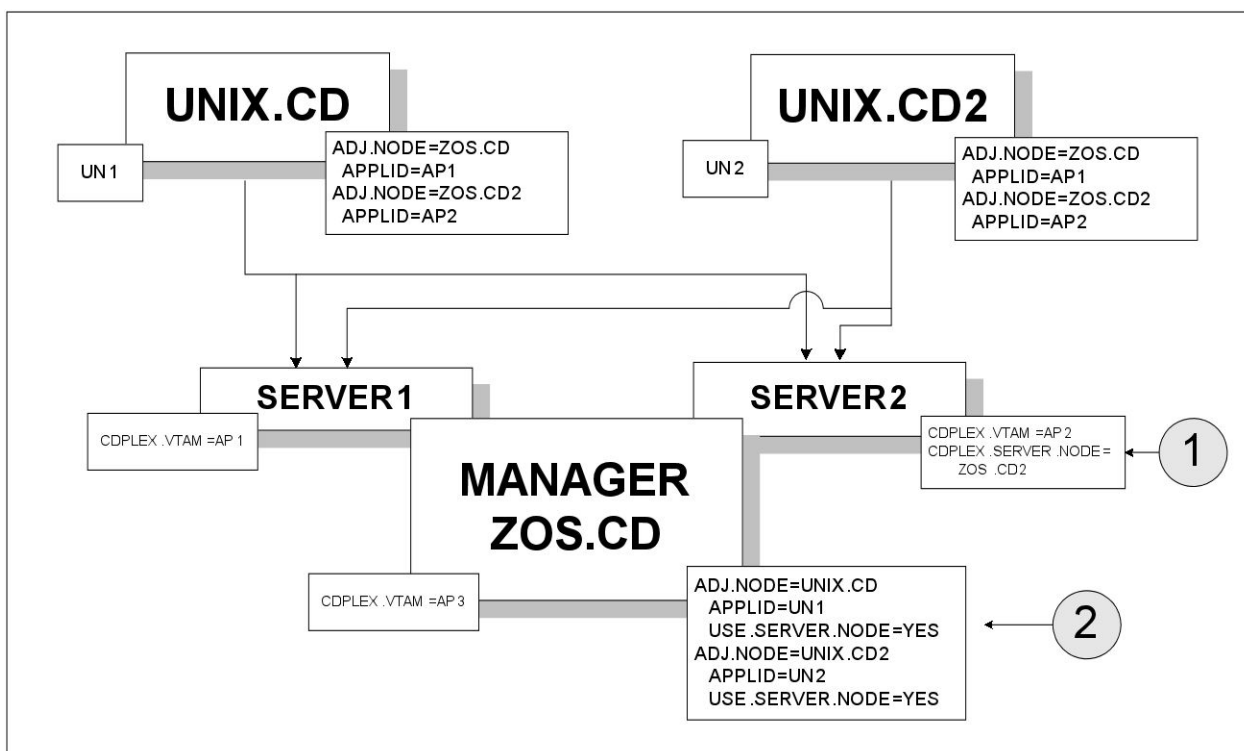
Configuration Example Using Two Connect:Direct for z/OS Systems

This configuration describes a more complex IBM Connect:Direct/Plex configuration. It assumes that the following IBM Connect:Direct environment exists.



In this environment, two Connect:Direct for UNIX nodes (UNIX.CD and UNIX.CD2) communicate with two different Connect:Direct for z/OS systems (ZOS.CD and ZOS.CD2).

To change this setup to a IBM Connect:Direct/Plex, the ZOS.CD and ZOS.CD2 systems are merged into a single IBM Connect:Direct/Plex. ZOS.CD is designated as the IBM Connect:Direct Manager and SERVER1, while ZOS.CD2 is designated as the IBM Connect:Direct Server SERVER2. (See [Converting Two Existing Stand-Alone Server Systems to a Plex Environment](#) for more information.)



The initialization parameter `CDPLEX.SERVER.NODE=ZOS.CD2` is added to the `SERVER2` local initialization parameter (callout 1). `USE.SERVER.NODE=YES` is added to the IBM Connect:Direct/Plex network map adjacent node definitions (callout 2).

The advantages of this approach are:

- Changes to the z/OS nodes have no affect on the Connect:Direct for UNIX nodes. Therefore, no changes are required to the Connect:Direct for UNIX nodes.
- You can perform workload balancing on Processes sent from `ZOS.CD` to the Connect:Direct for UNIX nodes.
- The IBM Connect:Direct/Plex provides a single administrative and operating environment.

The disadvantage of this approach is that you must manually balance Processes coming from the Connect:Direct for UNIX nodes.

Configuring Extended Recovery

IBM Connect:Direct uses the IBM Extended Recovery Facility (XRF) to quickly recover and resume processing after an abnormal termination. This recovery is accomplished by using a standby IBM Connect:Direct system that waits to resume processing if the active IBM Connect:Direct system fails.

Both the IBM Connect:Direct/Stand-alone Server and IBM Connect:Direct/Plex configurations support extended recovery. This chapter describes how to set up extended recovery in either environment.

Setting Up Extended Recovery for a IBM Connect:Direct/Stand-Alone Server

About this task

To configure a IBM Connect:Direct/Stand-alone Server to use extended recovery, follow this procedure.

Procedure

1. Specify the `XCF.NAME` in the initialization parameters file. The `XCF.NAME` is a unique 8-character string that identifies a IBM Connect:Direct/Stand-alone Server using extended recovery.

The following example shows a IBM Connect:Direct/Stand-alone Server assigned the `XCF.NAME` of `MNPLS`.

```
XCF.NAME=MNPLS
```

Note: XCF group names cannot begin with the letters A through J or with the letters SYS because these are reserved by IBM.

2. Add the following parameter to the initialization parameters file.

```
EXTENDED.RECOVERY=YES
```

3. Add the `XRFJOB` DD statement to the IBM Connect:Direct startup JCL as follows.

```
//XRFJOB DD DISP=SHR,DSN=$CD.SDGAJCL(CDJOBX)
```

`$CD.SDGAJCL` is the PDS where the active (current) IBM Connect:Direct startup JCL is located. The `CDJOBX` member is the standby IBM Connect:Direct startup JCL or startup command to run as a started task, which will be created later.

4. To run as a started task, use one of the following, or skip to [step 5](#) if you are not running as a started task.

- To run the IBM Connect:Direct standby image as a started task on the same z/OS image as the IBM Connect:Direct active image, use the following as the first statement in the JCL.

```
START=membername,parms
```

In this example, START= indicates to issue a START command. When the START command is issued, the equal sign (=) is replaced with a blank, and the entire statement is passed to z/OS as a command.

For example, if START=HOSTJCL,X is the first statement, then START HOSTJCL,X is issued to z/OS.

- To run the IBM Connect:Direct standby image as a started task on an z/OS image in the sysplex that is not the IBM Connect:Direct active image, use the following as the first statement in the JCL.

```
/*$VS, 'command'
```

Where command is the command you want the Job Entry Subsystem (in this case, a JES2 environment) to send to z/OS. Because the statement does not begin with START=, IBM Connect:Direct submits the statement to JES. JES identifies the /*\$VS and issues the command to z/OS rather than placing it in the job queue.

For example, if /*\$VS,'RO CSGB,S CDICOMB' is submitted in the JCL, the

RO CSGB,S CDICOMB command is issued rather than placed in the job queue.

5. Copy the IBM Connect:Direct startup JCL to the CDJOBX member if you are submitting the JCL.
6. Change the job name in CDJOBX so that it runs simultaneously with the active IBM Connect:Direct image. (Both the active IBM Connect:Direct image and the standby IBM Connect:Direct image run at the same time.)
7. Change the XRFJOB DD statement in the standby IBM Connect:Direct startup JCL to reference the IBM Connect:Direct startup JCL or command to run as a started task, as in the following example (where CDJOB is replaced by the member name that actually has your IBM Connect:Direct JCL).

```
//XRFJOB DD DISP=SHR,DSN=$CD.SDGACNTL(CDJOB)
```

8. Submit the CDJOB JCL to bring up IBM Connect:Direct using extended recovery.

After IBM Connect:Direct initializes, the JCL specified in the startup JCL initializes the standby IBM Connect:Direct system. The standby IBM Connect:Direct image partially initializes, then begins monitoring the active IBM Connect:Direct image.

If the active IBM Connect:Direct image terminates abnormally, the standby IBM Connect:Direct image resumes initialization, becomes the active IBM Connect:Direct image, and submits the JCL in its XRFJOB DD statement. This JCL initializes the original active IBM Connect:Direct image, which now becomes the standby system.

If the active IBM Connect:Direct image shuts down normally, the standby IBM Connect:Direct image also terminates normally.

Setting Up Extended Recovery for a IBM Connect:Direct/Plex Environment

About this task

This section describes how to configure the IBM Connect:Direct/Plex environment to use extended recovery.

Procedure

1. Add the following parameter to the initialization parameters file.

```
EXTENDED.RECOVERY=YES
```

2. Add the XRFJOB DD statement to the IBM Connect:Direct startup JCL as follows:

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDMGRX)
```

\$CD.PLEX.JCL is the PDS where the active (current) IBM Connect:Direct startup JCL is located. The CDMGRX member is the standby IBM Connect:Direct startup JCL or command to run as a started task, which will be created later.

3. To run as a started task, use one of the following, or skip to [step 4](#) if you are not running as a started task.
 - To run the IBM Connect:Direct standby image as a started task on the same z/OS image as the IBM Connect:Direct active image, use the following as the first statement in the JCL.

```
START=membername,parms
```

In this example, START= indicates to issue a START command. When the START command is issued, the equal sign (=) is replaced with a blank, and the entire statement is passed to z/OS as a command.

For example, if START=HOSTJCL,X is the first statement, then START HOSTJCL,X is issued to Connect:Direct for z/OS.

- To run the IBM Connect:Direct standby image as a started task on a different z/OS image in the sysplex as the IBM Connect:Direct active image, use the following as the first statement in the JCL.

```
/*$VS, 'command'
```

Where command is the command you want JES to send to z/OS (in this case, a JES2 environment). Because the statement does not begin with "START=", IBM Connect:Direct submits the statement to JES. JES identifies the /*\$VS and issues the command to z/OS rather than placing it in the job queue.

For example, if /*\$VS,'RO CSGB,S CDICOMB' is submitted as JCL, the

RO CSGB,S CDICOMB command is issued rather than placed in the job queue.

4. Create the CDMGRX member from the member that contains your MANAGER's JCL..
5. Change the job name in CDMGRX so that it runs simultaneously with other IBM Connect:Direct/Plex members.
6. Change the XRFJOB DD statement in CDMGRX to point to the CDMGR member, as in the following example.

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDMGR)
```

7. Add the following DD statement to the CDSRV1 member (the SERVER1 startup JCL) in \$CD.PLEX.JCL.

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDSRV1X)
```

8. Copy the changed CDSRV1 JCL member to CDSRV1X.
9. Change the job name in CDSRV1X so that it runs simultaneously with other IBM Connect:Direct/Plex members.
10. Change the XRFJOB DD statement in CDSRV1X to point to CDSRV1, as in the following example.

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDSRV1)
```

11. Add the following DD statement to the CDSRV2 member (the SERVER2 startup JCL) in \$CD.PLEX.JCL.

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDSRV2X)
```

12. Copy the changed CDSRV2 member to CDSRV2X.
13. Change the job name in CDSRV2X so it can run simultaneously with other IBM Connect:Direct/Plex members.
14. Change the XRFJOB DD statement in CDSRV2X to point to CDSRV2, as in the following example.

```
//XRFJOB DD DISP=SHR,DSN=$CD.PLEX.JCL(CDSRV2)
```

15. Submit the CDMGR JCL to bring up the IBM Connect:Direct image using extended recovery.

Each IBM Connect:Direct member initializes using extended recovery and submits the JCL specified in its XRFJOB DD statement. This JCL starts the standby IBM Connect:Direct members. Each standby IBM Connect:Direct member partially initializes, then begins monitoring its active IBM Connect:Direct member.

If the active member terminates abnormally, or is shut down with a STOP CD CDPLEX RECOVER command, the standby member resumes initialization, becomes the active member, and submits the JCL in its XRFJOB DD statement. This JCL initializes the original active IBM Connect:Direct/Plex member, which now becomes the standby member.

Note: If you want the standby member to run on a different z/OS image in the sysplex, you must define the VTAM APPLID as dynamic in both z/OS images, and you must define TCP/IP addresses as dynamic VIPA addresses.

Configuring SNMP Support

The Simple Network Management Protocol (SNMP) defines a set of protocols that describe management data and the protocols for exchanging that data between systems. This management data is a set of defined variables called the Management Information Base (MIB).

Three primary functional entities are defined for SNMP: managers, agents, and subagents. A manager is a network management application, like Netview or HP OpenView. The agent is a server that responds to request for management data from a network manager. Subagents provide support for particular MIBs to the agent.

The primary function of an SNMP environment and the communication between these functional entities is to enable the network manager to poll a device or application to inquire about specific management data that the device or application is monitoring. The device or application can alert the network manager of certain conditions by sending traps to reflect the status of that condition. Traps are asynchronous, unsolicited messages sent to the network manager, when the agent and/or subagent detect certain conditions.

IBM Connect:Direct provides support for an SNMP agent to send SNMP traps to alert a network manager of certain events. An event is any IBM Connect:Direct message that is written to the console using IBM Connect:Direct members. Each event is triggered by the IBM Connect:Direct message ID and the trap text (short message text of that IBM Connect:Direct message). The IBM Connect:Direct events generated are defined by category and type.

The IBM Connect:Direct Trap Table can hold up to 127 entries. It is built using the input from the data set specified by the SNMP.DSN initialization parameter. One entry is generated for each message that can trigger a SNMP trap. The predefined IBM Connect:Direct traps are triggered by 61 messages. A maximum of 66 additional user messages can be used to trigger SNMP traps. To see information about each SNMP trap defined in the table along with its status, see [INQUIRE SNMP Command Displays the SNMP Trap Table](#).

Define message traps using the SNMP.DSN initialization parameter and a data set that contains the variables associated with traps.

Identifying Trap Variables

Traps are defined as alarm or status alerts which enable the network manager to display the trap in the appropriate color on the network manager console. Alarm trap variables signal events that are critical to the operation of IBM Connect:Direct. Status trap variables signal events that are not critical to the operation of IBM Connect:Direct, but show valuable information. The tables in the following sections describe the predefined traps, the message that triggers the trap, and a description of the trap and associated text.

Following are the six categories for trap variables:

- Type events
- Initialization events
- Shutdown events
- API events
- Execution events
- STATS events
- Miscellaneous events

Valid values for all events is YES to enable and NO to disable.

Type Events

Use the events in the following table to enable or disable all alarm events or all status events.

Trap Event	Description	Event
sendAlarmTraps	NO disables all Alarm Trap Variables regardless of settings. YES enables all Alarm Trap Variables, allowing you to disable individual Alarm Trap Variables	Alarm
sendStatusTraps	NO disables all Status Trap Variables regardless of settings. YES enables all Status Trap Variables, allowing you to disable individual Status Trap Variables	Status

Initialization Events

The following table details alarm and status events that occur at initialization.

Trap Event	Description	Trap Trigger	Short Message Text	Event
Initializationcomplete	The initialization of this node completed successfully. In a IBM Connect:Direct Plex environment, the member is named in the message text.	SITA036I	IBM Connect:Direct rel-level for z/OS Initialization Complete.	Status

Trap Event	Description	Trap Trigger	Short Message Text	Event
snaNotAvailable	The SNA support is temporarily unavailable, because the SNA=NO initialization parameter is specified, the VTAM ACB is inactive and could not be opened, the VTAM ACB is disabled during IBM Connect:Direct processing or the IBM Connect:Direct VTAM APPLID is already used.	SVTJ018I	SNA Support is Not Available.	Status
snaNowAvailable	The VTAM ACB is successfully opened and IBM Connect:Direct now supports all SNA functions.	SVTJ019I	SNA Support is Now Available.	Status
tcpNotAvailable	The TCP support is temporarily unavailable either because the TCP=NO initialization parameter is specified, the TCP/IP connection cannot be established, or the connection to TCP/IP is terminated.	STCP103I	TCP Support is Not Available.	Status
tcpNowAvailable	The connection to TCP/IP is successful and all TCP functions are now supported.	STCP104I	TCP Support is Now Available.	Status

Shutdown Events

The following table details alarm and status events that occur at shutdown.

Trap Event	Description	Trap Trigger	Short Message Text	Event
abnormalShutdown	An abnormal termination occurred.	SSHA021I	Abnormal termination of IBM Connect:Direct.	Alarm
shutdownRequest	A IBM Connect:Direct Stop command issued.	SSHA002I	IBM Connect:Direct QUIESCE shutdown begun.	Status
		SSHA003I	IBM Connect:Direct Run Task IMMEDIATE shutdown begun.	
		SSHA004I	IBM Connect:Direct IMMEDIATE shutdown begun.	

Trap Event	Description	Trap Trigger	Short Message Text	Event
		SSHA019I	IBM Connect:Direct STEP shutdown begun.	
normalShutdownComplete	IBM Connect:Direct normal shutdown completed successfully.	SITB001I	IBM Connect:Direct Termination Complete.	Status

API Events

The following table details alarm and status events that occur from the API.

Trap Event	Description	Trap Trigger	Short Message Text	Event
maxUserReached	MAXUSER is reached.	STAA004I	Task not created, MAX IUI/API task count reached.	Status

Execution Events

The following table details alarm and status events that occur when a Process executes.

Trap Event	Description	Trap Trigger	Short Message Text	Event
processFailure	A IBM Connect:Direct Process failed with a return code greater than 0, due to abnormal session termination, NETMAP check failure, or FM72 Security failure.	SVTM024I	&var1 EXECPROC: FMH-72 -RECEIVED;	Alarm
		SVTM026I	SESSION (&class) NOT ESTABLISHED WITH &node=&snode	
		SVTM030I	&var1 FMH-74 -RECEIVED AFTER STEP ERROR:	
		SVTM050I	&var1 PROCESS INTERRUPTED: RECOVERY INITIATED	
		SVTM052I	&stpnm &func &pname(&pnum) &node=&snode &var1	

Trap Event	Description	Trap Trigger	Short Message Text	Event
		SVTM054I	&var1 SNODE REQUESTING SESSION SHUTDOWN F/ END_OF_STEP	
		SVTM063I	PASSWORD NOT MATCHED IN C:D AUTH FILE-- MSG=SAFB005I	
		SVTM102I	MSGID=&msgid &msgtext,NODE =&snode SENSE=&sense LUNAME=&slu	
sessionRetryExceeded	The IBM Connect:Direct Process exceeded the session retry threshold and is placed in the Hold queue.	SVTM505I	Session Retry exceeded for &pname &pnum	Alarm
processRetryExceeded	The IBM Connect:Direct Process exceeded the process retry threshold and is placed in the Hold queue.	SVTM506I	Process Retry exceeded for &pname &pnum	Alarm
maxProcess	MAXPROCESS value is reached.	STAA010I	Task not created. Max Process count reached.	Alarm
maxPnode	The maximum number of PNODE Processes is reached.	STAA006I	Task not created. Max primary task count reached.	Alarm
ProcessNotStarted	Process was not started because IBM Connect:Direct quiesced and a task could not be created.	STAA011I	SNODE task not created, Session Quiesce in progress	Alarm
		STAA012I	PNODE task not created, Session Quiesce in progress	
		STAA008I	Task not created. No free TCA available.	
		STAA003I	Task not created. IBM Connect:Direct is quiescing.	
maxSnode	The maximum number of SNODE Processes is reached.	STAA005I	Task not created. Max secondary task count reached.	Alarm

Trap Event	Description	Trap Trigger	Short Message Text	Event
tcpCloseFailure	A TCP Close failed leaving the TCP/IP socket in use and unavailable.	STCP105I	TCP Close Socket Failure	Alarm
userMessageAlarm	A user-defined IBM Connect:Direct message is issued.	User-defined	message text	Alarm
tcqMovement	A IBM Connect:Direct Process is moved to the Hold queue due to errors during Process execution.	SVTM105I	PNAME=&pname, PNUM=&pnum MOVED TO Q=&q, QSTATUS=&qstat	Status
tcqMovement	A IBM Connect:Direct Process is moved to the Process Retain queue.	SVTM105I	PNAME=&pname, PNUM=&pnum MOVED TO Q=&q, QSTATUS=&qstat	Status
processFlushed	A IBM Connect:Direct Process is flushed.	SOPD049I	IBM Connect:Direct Process,&pname &pnum, flushed by &userid.	Status
userMessageStatus	A user-defined IBM Connect:Direct message is issued.	User-defined	message text	Status

STATS Events

The following table details alarm and status events that occur due to the STATS queue.

Trap Event	Description	Trap Trigger	Short Message Text	Event
statsDisabled	An error occurred that caused the STATS logging to be disabled.	SSTL001I	Statistics logging function is disabled.	Alarm
statsStress	IBM Connect:Direct STATS queue is under stress.	SSTL041I	Statistics facility under stress, waiting on queue elements.	Alarm
statsStressResolved	The STATS Queue stress is resolved.	SSTL042I	Statistics facility stress resolved.	Alarm
statsSwitchOccurred	A IBM Connect:Direct STATS file switch has occurred.	SSTL013I	Statistics file pair switch from &a to &b	Status

Miscellaneous Events

The following table details other alarm and status events.

Trap Event	Description	Trap Trigger	Short Message Text	Type
tracesEnabled	A IBM Connect:Direct MODIFY DEBUG command is issued.	STRA028I	IBM Connect:Direct Traces enabled.	Status
netmapUpdate	Dynamic update of the IBM Connect:Direct NETMAP occurred.	SMUP191I	IBM Connect:Direct NETMAP file updated.	Status
authUpdate	Dynamic update of the IBM Connect:Direct AUTH file occurred	SAFC005I	IBM Connect:Direct AUTH file updated.	Status
typeUpdate	Dynamic update of the IBM Connect:Direct TYPE file occurred	SAFI013I	IBM Connect:Direct TYPE file updated.	Status
initparmRefresh	Dynamic update of the IBM Connect:Direct INITPARM file occurred	SITA992I	INITPARM Refresh by &userid completed	Status
changeProcess	A CHANGE PROCESS command occurred.	SOPB017I	Change Process command by &userid completed.	Status
deleteprocess	A DELETE PROCESS command occurred.	SOPC011I	Delete Process command by &userid completed	Status
tcqFull	TCQ file is full.	SPQL001I	TCQ File Full	Alarm
tcqThreshold	TCQ file becoming full.	SPQL002I	TCQ Full &VAR1% Full. Max.# CI: &VAR2 # Used CI: &VAR3	Alarm
tcqThresholdResolved	TCQ is now below the defined threshold.	SPQL003I	TCQ File is now below the user defined Threshold of% %.	Status

Setting Up SNMP

About this task

Use the following procedure to set up SNMP Support.

Note: Before performing this procedure, migrate the CDMIB and IBM Connect:Direct Trap Configuration files to HP OpenView. Refer to *Customizing SNMP in Installing IBM Connect:Direct* in the *IBM Connect:Direct for z/OS Configuration Guide* for more information.

Procedure

1. Specify SNMP=YES in the initialization parameters file.
2. If you want to exclude or disable any trap event or define additional trap triggers described in [Identifying Trap Variables](#), create a data set containing all the trap events that you want to disable. Following is an example.

Note: All traps are enabled by default.

```
sendStatusTraps = N
statsDisabled = N
statsStress = N
statsStressResolved = N
userMessageAlarm = ( SVTM100I , SVTM101I )
```

A sample SNMP.DSN file is in the \$CD.SDGASAMP data set, member DGAXSNMP.

If you do not want to exclude any trap events, go to [step 4](#).

3. Set the SNMP.DSN initialization parameter to the data set name created in [step 2](#).

```
SNMP.DSN=data set name
```

4. Set the SNMP.MANAGER.ADDR initialization parameter. This parameter is the TCP/IP address or hostname of the host where the SNMP Network Manager is initialized. By default, this address is the same as the TCP/IP address that IBM Connect:Direct is using or the local hostname. In a IBM Connect:Direct/Plex environment, the default is the TCP/IP address of the IBM Connect:Direct Manager. This parameter is required if the network manager resides on a different host or is required to use a different TCP/IP address. Following is an example.

```
SNMP.MANAGER.ADDR=123.4.5.6
```

5. Set the SNMP.MANAGER.PORTNUM initialization parameter. This port is the TCP/IP port that is defined for UDP traffic to the network manager. The default is 162. This parameter is required if the defined UDP port number is something other than 162. Following is an example.

```
SNMP.MANAGER.PORTNUM=163
```

After SNMP Setup

At IBM Connect:Direct installation, the SNMP trap table is initialized and whenever any event occurs after the SITA628I message is issued, IBM Connect:Direct determines if the event is a trap trigger and issues the appropriate trap to a network manager. The following messages are common at initialization:

- SITA001I Connect:Direct for z/OS initialization has begun.
- SITA002I Connect:Direct parameter file allocated and open.
- SITA022I Loading Connect:Direct modules.
- SITA601I The TCP server modules are loaded.
- SITA067I MESSAGE file is open.
- SITA628I SNMP Trap Agent Initialization Complete.

If any error occurs during initialization of SNMP, the appropriate message is issued to indicate that the SNMP Trap Agent is disabled or that the initialization will terminate.

You can refresh and modify the SNMP initialization parameters after initialization completes by using the MODIFY INITPARM command.

INQUIRE SNMP Command Displays the SNMP Trap Table

The INQUIRE SNMP command displays the contents of the SNMP trap table. The INQUIRE SNMP command has the following format.

Label	Command	Parameter
(optional)	INQUIRE SNMP	

Issuing the INQUIRE SNMP Command through the IUI

About this task

To display the contents of the SNMP trap table from the IUI, follow this procedure.

Procedure

1. Select option INQ from the Connect:Direct Administrative Options Menu.
The Inquire DTF Internal Status screen is displayed.
2. Select the ITRP option.
3. Press **ENTER**.

The contents of the SNMP trap table are displayed, as in the following sample.

```
*****
BROWSE      SYS11069.T114309.RA000.MWATL1.NDMAPI.H01          Line 00000056 Col 001 132
Command
====>

Scroll ==> CSR
=====
SC.DUB.MWATL3      *SNMP TRAP TABLE*    DATE: 2011.03.10  TIME: 11:43:22
=====
TRAP TRIGGER      TRAP NAME                      TRAP STATUS      TRAP TYPE
=====
SVTM506I          PROCESSRETRYEXCEEDED    ENABLED           STATUS
SVTM105I          TCQMOVEMENT            ENABLED           STATUS
SOPD049I          PROCESSFLUSHED          ENABLED           STATUS
SSTL013I          STATSSWITCHOCCURRED    ENABLED           STATUS
STRA028I          TRACEENABLED            ENABLED           STATUS
SMUP191I          NETMAPUPDATE            ENABLED           STATUS
SAFC005I          AUTHUPDATE              ENABLED           STATUS
SAFI013I          TYPEUPDATE              ENABLED           STATUS
SITA992I          INITPARMREFRESH         ENABLED           STATUS
SOPB017I          CHANGEPROCESS           ENABLED           STATUS
SOPC011I          DELETEPROCESS           ENABLED           STATUS
SPQL003I          TCQTHRESHOLDRESOLVED    ENABLED           STATUS
***** Bottom of Data *****
```

Using the INQUIRE SNMP Command from the Batch Interface

About this task

To use the INQUIRE SNMP command from the Batch interface, follow this procedure.

Procedure

1. Place your commands in a batch job stream as demonstrated in the *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running. The settings are displayed.

Note: You must set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

IBM Connect:Direct Exits

Note: APARs and PTFs from IBM have "HOLD for ACTION" directives that identify if reassembly of exits is required. Reassemble exits accordingly. If you upgrade, reassemble your exits to pick up any macro or control block changes.

IBM Connect:Direct provides several sample exits some of which can be used to customize the online execution of IBM Connect:Direct. One of these exits, the Stage 2 Security exit, can be used to test new applications and customer connections. For more information, see [Process Exit for Testing \(DGAXPRCT\)](#).

Before coding or using an exit, read the Special Considerations section below to ensure that the exit executes properly.

Note: All sample exits provided in IBM Connect:Direct define the proper AMODE and RMODE settings within the source member themselves. All user exits should be link-edited with AMODE=ANY and capable of executing in 31-bit mode (calls to user code will have the parameter list passed using 31-bit addresses). Each user exit should preserve the mode in which it was invoked and return to the caller in the proper mode. Modules written to execute in 31-bit mode can be link-edited with RMODE=ANY or RMODE=24. Check the source for the sample exits to see how IBM Connect:Direct defines the proper AMODE and RMODE settings.

To notate modifications by user in the the module maintenance history section of NDMLOG, include the local user ID by specifying the &UID SETC 'xxxx' local identifier in IBM Connect:Direct exits as part of the SCENTER macro. This value can contain up to 8 characters.

Sample JCL for assembling user exits is provided in \$CD.SDGASAMP library members DGAXSTG1 and DGAXSTG2.

Special Considerations

The following special considerations apply to Connect:Direct for z/OS exits:

- Avoiding Out-of-Storage ABENDS
 - To avoid out-of-storage ABENDS in IBM Connect:Direct, examine all user exits to verify that all obtained storage either via GETMAIN or STORAGE OBTAIN is freed. For each GETMAIN/STORAGE OBTAIN that an exit issues, the exit must also issue a corresponding FREEMAIN/STORAGE RELEASE to avoid accumulating storage. If an exit opens a file, you may need to issue a FREEPOOL after the file is closed.
- Using Exits in 31-Bit Addressing Environments
 - Because information passed to the exit by IBM Connect:Direct is located above the 16 megabyte line, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode. Refer to the section in this chapter describing the particular exit to see if this requirement applies.
 - IBM Connect:Direct honors the addressing mode (AMODE) and residence mode (RMODE) attributes of user exits. The exit modules are loaded based on the RMODE specification and given control in the addressing mode specified in the AMODE attribute. Link exits that run above 16 megabytes in 31-bit mode to AMODE=ANY, RMODE=ANY.
 - Verify that your exits are coded to receive control and execute in the AMODE with which they are linked.
 - Exits must return control to IBM Connect:Direct in the AMODE in effect when IBM Connect:Direct invokes the exit. IBM Connect:Direct calls your exit through Branch and Save and Set Mode (BASSM), and you return to IBM Connect:Direct through Branch and Set Mode (BSM).

Note: For security exit links, you may need to provide access to a load library containing the modules for the security system in use. [Linkage Editor Attribute Requirements](#)

- Linkage Editor Attribute Requirements
 - You must create all exits that execute in the DTF address space and link-edit them as RENT and REUS. Only the Stage 1 exits (DGACXSIG and DMCXSUBM) do not require re-entrancy.

Statistics Exit

IBM Connect:Direct generates and logs statistics to an online journal, then writes the information to the IBM Connect:Direct statistics log as individual records.

Each record contains information about a single event, and is identified by a 2-character record type. For example, type CT is a copy termination record and FP is a flush Process record.

IBM Connect:Direct provides a statistics exit that gives a user-written program access to the statistics records as they are generated. This exit can:

- Output the records or data generated from the records to a user-defined journal, including an SMF log
- Include or exclude the logging of any record to IBM Connect:Direct by return codes



CAUTION: Statistics records are often essential in debugging IBM Connect:Direct problems. Excluding records from the statistics log makes problem determination more difficult or even impossible.

The statistics exit runs as a subtask in the IBM Connect:Direct DTF address space. IBM Connect:Direct uses the STATISTICS.EXIT initialization parameter to specify the exit module name. You define this name, but it cannot conflict with the name of any IBM Connect:Direct module. If a user-defined journal is required, you must add the necessary data definition (DD) statements to the IBM Connect:Direct startup job stream.

Note: In a IBM Connect:Direct /Plex environment, the statistics exit, by default, only runs on the IBM Connect:Direct /Manager. For more information on how to cause the statistics exit to run on the IBM Connect:Direct /Plex server, see [“STATISTICS.EXIT = modname | \(modname\[,MANAGER | SERVER | BOTH\]\)” on page 484.](#)

The statistics exit indicates if a record is logged by setting a return code set in the SQUSE field of the SQCB. The return code is initialized to zero before the exit is invoked.

You do not have to rewrite existing exits if you do not want to exclude records. Existing exits do not alter the SQUSE field and operate the same as before.

Sample Statistics Exits

IBM Connect:Direct provides the following sample statistics exits in \$CD.SDGASAMP library:

Exits	Description
DGAXSMF	This sample exit logs IBM Connect:Direct statistics records to the SMF log. It logs records to the SMF log by prefixing each statistics record with an SMF record header, and then uses the SMFWTM macro to write to the SMF log. The SMF record is type 132.
DGAXEV01	<p>This sample exit provides a means for an application to access event data. For more information on how to use the Event Services Support (ESS) facility with this exit, see the <i>IBM Connect:Direct for z/OS Facilities Guide</i>.</p> <p>This exit writes each event record to a predefined data set. You must modify DGAXEV01 to specify the name of your event exit data set, and define the data set to accommodate records up to 2048 bytes in length.</p>
DGAXSACC	<p>This exit documents the path to log user accounting data from the copy termination (CT) records.</p> <p>This exit is invoked for every IBM Connect:Direct statistics record written to the statistics file, but only processes CT records.</p>
DGAXSXIT	This sample exit simply checks for copy termination records. When IBM Connect:Direct encounters a copy termination, the system issues a WTO.

Exits	Description
DGAXSXMC	This sample exit precludes the logging of PDS member copy (MC) records that have good return codes. It enables the logging of records of this type only when they have non-zero return codes. It also enables the logging of all other record types regardless of their return codes.
DGAXSCPU	<p>Beginning with Release 5.1, CPU time fields are present in record extensions addressed by offsets from the beginning of the Statistics record for Copy Termination (DGA\$CTR), Run Task Termination (DGA\$RTTR) and Process Termination (DGA\$PTR). Run Job Termination (DGA\$RJTR) does not contain CPU time fields.</p> <p>DGAXSCPU now shows how to access those CPU time fields as well as other fields in the record extension areas.</p> <p>For demonstration purposes, this code produces diagnostic output for each Copy Step showing Process Name, Number and SNODE Name along with counts for records and blocks read / written and finally the time on CP plus time on ZIIP for both sending and receiving as captured in the CT record. This output goes to DDNAME=SCDIAG. DGAXSCPU also shows how to interpret the CTFXNODE flag and report the SNODE name for all releases correctly.</p> <p>Note: The sole purpose of this sample exit is to demonstrate accessing fields in Stat Record extensions and report PNODE/SNODE fields. It is not recommended to run it in production environments.</p>

Statistics Exit Calling Conventions

IBM Connect:Direct calls the statistics exit once for each statistics record generated. Standard linkage conventions apply.

The exit is given control with register 1 pointing to a list of two parameters. They are:

- The first parameter is a pointer to the statistics record.
- The second parameter is a pointer to an SQCB that you need for setting a return code if record exclusion is appropriate.

The first 2 bytes of the record contain the record length in binary format. The third and fourth bytes of the record contain the 2-character record identifier. Refer to [“Statistics Records” on page 318](#) to see a table containing a list of the record type identifiers.

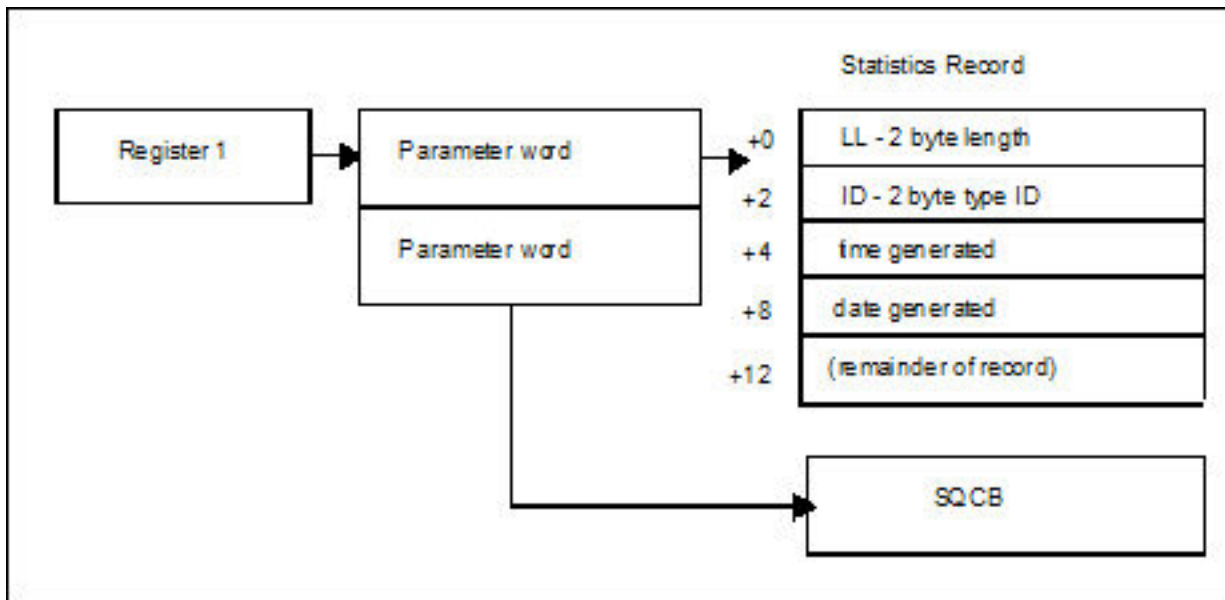
The second word of the record contains the time of day that the record is generated. The third word contains the date the record is generated.

Additional information in the records depends on the record type.

Assembler macros are provided in \$CD.SDGAMAC library to generate dummy sections (DSECTS) to map all the record types. The exit program includes the DSECTS that map the record types to the exit processes.

A return code of 0 indicates that the record is logged. A return code of 4 indicates that the record is not logged.

The following figure depicts the information passed to the exit.



Statistics Records

IBM Connect:Direct calls the statistics exit once for each statistics record generated in the DTF. The input to the exit is a pointer to the statistics record that is ready to be logged and a pointer to an SQCB. The record can be any record type. The exit must examine the record type identifier at a displacement of X'02' bytes from the beginning of the record to determine the record type and the DSECT that describes its contents.

Record Types

The following table lists the statistics record types, their corresponding record type identifiers, and the name of the assembler macro in \$CD.SDGAMAC library that generates the DSECT describing the record contents. For information about selecting, displaying, and printing statistics information for IBM Connect:Direct activities, see *IBM Connect:Direct for z/OS User Guide*.

The statistics records in this section also apply to Event Services Support.

Record ID	Description	Macro
CE	Copy I/O Start	DGA\$STEP
CH	Change Process	DGA\$CPTR
CI	Copy Step Start	DGA\$STEP
CS	Statistics Command	DGA\$SCMD
CT	Copy Termination	DGA\$CTR
CX	Check Certificate Validity	DGA\$XCR
DC	Directory Commands	DGA\$DTR
DP	Delete Process	DGA\$DPTR
DT	Select Task	DGA\$DTR
DU	Delete User	DGA\$AER
EI	Event Services Start Command	DGA\$EVR
ET	Event Services Stop Command	DGA\$EVR
EV	Event Services Command	DGA\$EVR

Record ID	Description	Macro
FA	IGWFAMS Message	DGA\$FAMS
FI	Long File Name Record	DGA\$FIR
FP	Flush Process	DGA\$FPTR
FS	Suspend Process	DGA\$FPTR
FT	Flush Task	DGA\$DTR
GO	Process Modal - GOTO, ELSE, or EXIT Statement	DGA\$MODL
H2	High Concurrent Session Count	DGA\$H2R
HW	High Concurrent Session Count	DGA\$HWR
IA	Inquire Statistics	DGA\$DTR
IB	Inquire Debug	DGA\$DTR
ID	Inquire STATDIR	DGA\$DTR
IF	Process Modal - IF Statement	DGA\$MODL
IK	Inquire APKey File	DGA\$DTR
IP	Inquire Initialization parameters	DGA\$DTR
IT	Inquire SNMP Trap Table	DGA\$DTR
IU	Insert User	DGA\$AER
IX	Inquire IBM Connect:Direct/Plex	DGA\$DTR
JI	Run Job Start	DGA\$STEP
LF	ICO Log File Record	DGA\$FRER
M2	Multiple Copy Record	DGA\$MCR
MC	PDS Member Copy	DGA\$MCR
NL	Process modal - EIF or PEND statement	DGA\$MODL
NM	NETMAP Updated	DGA\$NMR
OP	Operator Clist	DGA\$OPR
PE	IBM Connect:Direct/Plex Error Record	DGA\$PER
PI	Process Start	DGA\$PIR
PR	Performance Measurement Record	DGA\$PRRB
PS	Process Submit	DGA\$PSSR
PT	Process Termination	DGA\$PTR
PX	IBM Connect:Direct/Plex Activity (Leave or Join IBM Connect:Direct/Plex)	DGA\$PXR
QE	Queue Change to EXEC Queue	DGA\$QCR
QH	Queue Change to HOLD Queue	DGA\$QCR
QT	Queue Change to TIMER Queue	DGA\$QCR
QW	Queue Change to WAIT Queue	DGA\$QCR
RE	ICO Report Record	DGA\$FRER

Record ID	Description	Macro
RF	Refresh/Update initialization parameters	DGA\$AER
RJ	Run Job	DGA\$RJTR
RO	ICO Event Record	DGA\$OEVT
RT	Run Task	DGA\$RTTR
S2	Statistics Logging Statistics	DGA\$S2R
SB	Session Begin	DGA\$SSCR
SE	Session End	DGA\$SECR
SC	Statistics Control Record	DGA\$SCR
SD	Start IBM Connect:Direct	DGA\$SDC
SF	Statistics Format	DGA\$SFRC
SI	Signon	DGA\$SFR
SN	Select Netmap	DGA\$DTR
SO	Signoff	DGA\$SFR
SP	Select Process	DGA\$DTR
SS	Select Statistics	DGA\$DTR
ST	Stop IBM Connect:Direct	DGA\$STDC
SU	Select User	DGA\$AER
SW	Submit within a Process	DGA\$PSSR
SY	SYSOPTS	DGA\$SYR
TF	TCQ Threshold Full	DGA\$TXR
TI	Run Task Start	DGA\$STEP
TL	TCQ Threshold Low	DGA\$TXR
TP	Throughput Record Statistics	DGA\$TPR
TR	Trap Event Record	DGA\$TRP
TS	Suspend Task	DGA\$FPTR
TW	TCQ Threshold Warning	DGA\$TXR
UM	Update Network map	DGA\$AER
UU	Update User	DGA\$AER
VP	View Process	DGA\$DTR
WO	WTO	DGA\$FWTO
WS	Select Stat Command	DGA\$DTR
XO	Trace On/Off	DGA\$XOR
ZI	SNODE Process Start	DGA\$PIR
ZT	SNODE Process Terminated	DGA\$PTR

Record Control Block Maps

The following table lists the statistics record control block maps.

Macro Name	Description
DGA\$AER	Authorization Event Statistics Record
DGA\$CPTR	Change Process Statistics Record
DGA\$CTR	Copy Termination Statistics Record
DGA\$DPTR	Delete Process Statistics Record
DGA\$DTR	Display Termination Record
DGA\$EVR	Event Services Command Statistics Record
DGA\$FAMS	IGWFAMS (File and Attribute Management Services) Macro Statistics Record
DGA\$FIR	Long File Name Statistics Record
DGA\$FPTR	Flush and Suspend Process Statistics Record
DGA\$FRER	InterConnect Report Record containing text line from SYSPRINT (Record Type is RE)
DGA\$FRER	InterConnect Log File Records produced if LOG=YES is specified for ADD and EXTRACT operations (record type is LF)
DGA\$FWTO	WTO Statistics Record
DGA\$H2R	High Concurrent Session Count
DGA\$HWR	High Concurrent Session Count Statistics Record
DGA\$LSR	Log Swap Statistics Record
DGA\$MCR	PDS Member Copy Record
DGA\$NMR	NETMAP Updated
DGA\$OEVT	InterConnect Report Event Record containing one record per report written (Record Type is RO)
DGA\$OPR	Operator Clist Record
DGA\$PER	XCF Error Message Statistics Record (from Sysplex)
DGA\$PIR	Process Initiation Statistics Record
DGA\$PRRB	Performance Measurement Statistics Record
DGA\$PSSR	Submit Process Statistics Record
DGA\$PTR	Process Termination Record
DGA\$PXR	Sysplex (IBM Connect:Direct /Plex) Statistics Record
DGA\$QCR	Process Queue Change Statistics Record
DGA\$RJCB	Run Job exit
DGA\$RJTR	Run Job Termination Record
DGA\$RTTR	Run Task Termination Record
DGA\$S2R	Statistics Logging Record
DGA\$SCMD	Statistics Command Record
DGA\$SCR	Statistics ESDS Control Record

Macro Name	Description
DGA\$SDC	Start IBM Connect:Direct Statistics Record
DGA\$SFR	Signon/Signoff Statistics Record
DGA\$SFRC	Statistics Format Record
DGA\$SGNB	Connect:Direct Secure Plus Statistics Record
DGA\$SSCR	Session Begin Record
DGA\$SECR	Session End Record
DGA\$STDC	Stop IBM Connect:Direct Statistics Record
DGA\$STEP	Step Start/Copy Start Statistics Record
DGA\$SYR	SYSOPTS Record
DGA\$TPR	Throughput Record Statistics
DGA\$TQGT	GOTO Statement
DGA\$TQIF	If Statement
DGA\$TQNL	NULL Statement
DGA\$TRP	Trap Event Record
DGA\$TXR	TCQ Threshold Warning TCQ Threshold Full TCQ Threshold Low
DGA\$WRP	Statistics File Wrap Record
DGA\$XOR	TRACE On/Off Statistics Record

Submit Exit

The Submit exit provides an interface to a user-written program when you submit a Process. With this interface, the user program can change Process information, such as Process name, priority, class, and secondary node, and copy step information such as data set name.

Sample Submit Exits

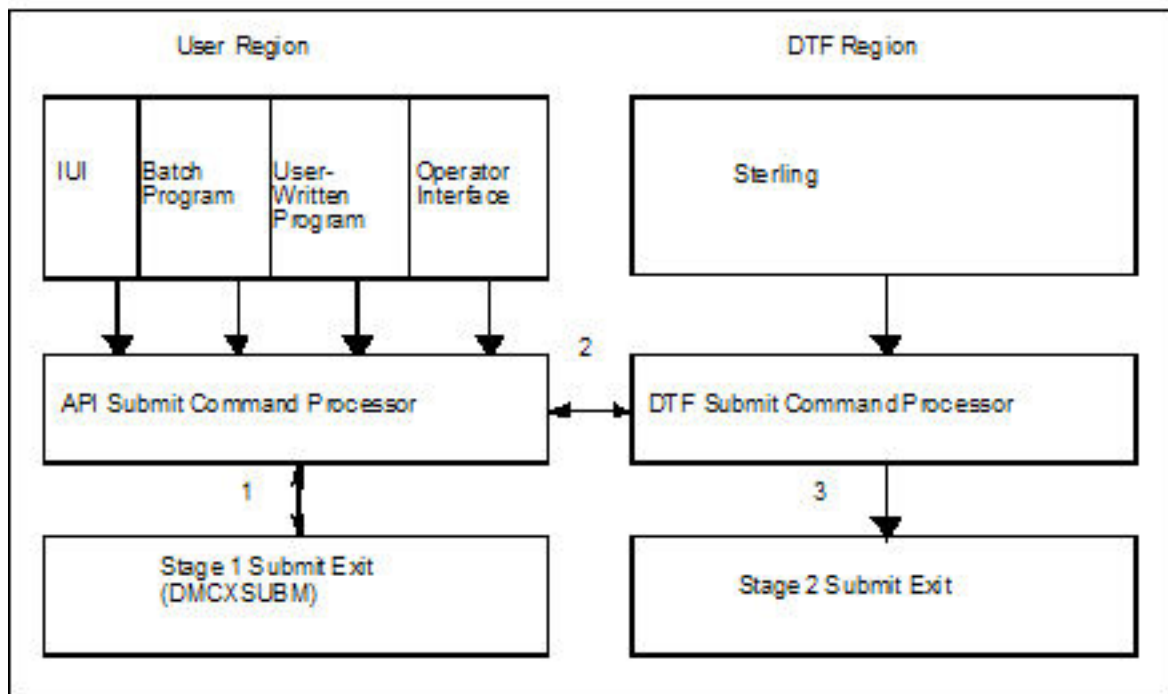
IBM Connect:Direct provides the following sample submit exits in \$CD.SDGASAMP library.

Exits	Description
DGAXSUBC	This exit sets a 0 return code to pass back to the calling program. It also locates and increments the Process class by 1. It runs either as a stage 1 or stage 2 exit.
DGAXSUBD	This exit documents the control block path to locate the source and destination data set names specified in the Process. It sets a 0 return code to pass back to the calling program. It also locates and increments the Process class by 1. It runs either as a stage 1 or stage 2 exit.
DGAXSUBX	This exit sets a return code of 0 to pass back to the calling program. It runs either as a stage 1 or stage 2 exit.

Exits	Description
DGAXSUBN	This exit turns off all compression flags in the Process for a node defined as EXTERNAL, which is a function of the Stage2 Security exit and indicated in the NETMAP. If it is named DMCXSUBM, it is invoked as a Stage1 submit exit for normal submits as well as a Stage2 submit exit for submits within a Process.
DGAXSUBP	This exit determines if the submitted Process contains a RUN JOB statement. If the Process contains a RUN JOB statement and the exit encounters a stage 1 password, the exit returns an error. You must then provide either a PNODEID or an SNODEID, depending on which node the Process is executing. Use the PNODEID or SNODEID to validate the RUN JOB submitted job stream through the user=uid, password=pwd parameter built by the RUN JOB exit.
DGAXSUBR	This exit rejects all submits by setting TARTNCD as 16, TQMSGID of SCBI159I and setting specific text in UIERRM1 and UIERRM2.
DGAXORR	This exit documents the control block path to determine whether the Process is performing a send or receive. It sets a return code of 0 to pass back to the calling program. It runs either as a stage 1 or stage 2 exit.
DGAXACCT	This exit is a Stage 1 SUBMIT exit example that shows how to update the Primary and Secondary Node accounting (PACCT and SACCT) information in the submitted Process. Note: To update accounting information, you must also update your SUBMIT exit.

Submit Exit Processing Flow

The following figure illustrates the execution order of the SUBMIT command:



The Submit exit processing flow is:

1. After you issue a IBM Connect:Direct SUBMIT command or SUBMIT statement, the API SUBMIT command processor calls the stage 1 Submit exit.
2. If the submit is successful, the API SUBMIT command processor calls the DTF SUBMIT command processor.

3. The DTF SUBMIT command processor calls the stage 2 Submit exit.

IBM Connect:Direct provides a sample Submit exit in \$CD.SDGASAMP library, called DGAXSUBX, which you can use as a model for either the stage 1 (DMCXSUBM) or stage 2 (SUBMIT.EXIT = modname) exit. In most cases, you only need to run the stage 1 Submit exit.

Stage 1 Submit Exit

The stage 1 Submit exit control point executes in the API address space when a SUBMIT command is processed and in the DTF address space when a SUBMIT statement is encountered in a Process. Observe the following restrictions and requirements:

- The IBM Connect:Direct stage 1 Submit exit is implemented as an executable load module.
- You must name the load module DMCXSUBM.
- You must link-edit the module as NORENT and NOREUS.
- You must link-edit the module with an authorization code of 1.
- The module must come from an authorized library.
- For the TSO IUI, the module must come from a library in the LNKLIST or ISPLLIB. Do not put the module in a STEPLIB. The only time a STEPLIB works under ISPF is when ISPLLIB is not allocated.
- For DGADBATC and DGADCHLA, retrieve the module from a library in the LNKLIST or STEPLIB.
- Because information passed to the exit by IBM Connect:Direct is located above the 16 megabyte line, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.

Stage 2 Submit Exit

The stage 2 Submit exit control point executes in the DTF address space when a SUBMIT command or a SUBMIT statement is encountered. Observe the following restrictions and requirements:

- The stage 2 Submit exit is implemented as an executable load module.
- The name of the load module is user-defined, but cannot conflict with any IBM Connect:Direct load module names.
- Activate the stage 2 Submit exit by specifying SUBMIT.EXIT=(modname) in the IBM Connect:Direct initialization parameters.
- You must link-edit the module as re-entrant and place it in a load library that the IBM Connect:Direct DTF can access.
- The module must come from an authorized library.
- Because information passed to the exit by IBM Connect:Direct is located above the 16 megabyte line, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.

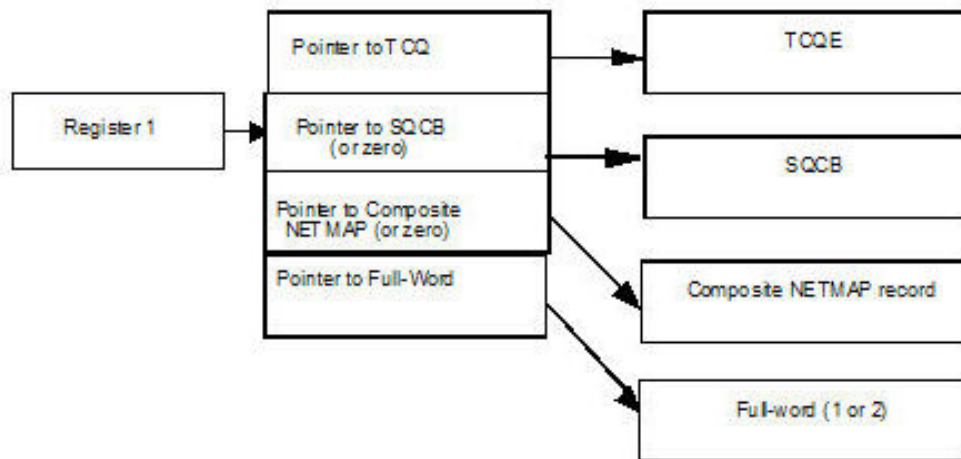
Control Block Format

Because Submit exits are invoked before a Process is submitted, some control block fields may not be filled in. This section presents control blocks used with Process statements and NETMAP entries.

Upon entry to the Submit exit, Register 1 points to a parameter list (PLIST) as shown in the following figure. This list contains the addresses of the following components:

- TCQE
- SQCB
- Composite NETMAP record
- A full-word that contains 1 or 2 indicating whether this is a Stage 1 or Stage 2 exit
- UICB (User Interface Control Block)

Note: The SQCB and composite NETMAP record addresses may be zero.



If the Process submission is rejected, the exit must set a positive non-zero value in Register 15 and the return code field (TQRTNCD) of the TCQE. A message ID in the TQMSGID field in the TCQE should also be set.

The only modifiable field in the UICB are UIERRM1 and UIERRM2 for which you can return your own message text to Connect:Direct, the exit must put up to 64 bytes of text into UIERRM1. If desired, up to 64 more bytes can be put into UIERRM2. If nothing is returned in UIERRM1, then both UIERRM1 and UIERRM2 will be ignored.

The following figure illustrates the layout of the TCQE. In the Process contained in the Statement Control Block, DMxxxxxx represents the macro name for the statement (COPY, RUN JOB, RUN TASK, SUBMIT, etc.).

TCQ Header	macro= DMTCQE DSECT= TCQE
Command Statement Header Section	macro= DGA\$TQSH DSECT= TCQSH
Statement Control Block	macro= DMxxxxxx

Displacement values found in the TCQE and pointers in the TCQSH to the next or previous TCQSH are from the top of the TCQE and may need to be multiplied by 16 (if the Process is larger than 64K). Displacement values found in the statement control blocks are from the top of the TCQSH associated with that statement control block.

The following figure illustrates the layout of the composite NETMAP record.

COMPOSITE NETMAP RECORD \$\$\$NN - displacement to ADJACENT NODE Record \$\$\$AA - displacement to ALT.COMM Record \$\$\$BA - displacement to BATCH.APPLI Record \$\$\$CA - displacement to CICS.APPLID \$\$\$DN - displacement to LDNS Record \$\$\$NA - displacement to APPLIDS Record \$\$\$ND - displacement to LUPOOL Record \$\$\$NT - displacement to TCP.API Record \$\$\$TA - displacement to TSO.APPLIDS Record \$\$\$N6 - displacement to IPv6 Record \$\$\$NU - displacement to UDT Record \$\$\$CM - displacement to COMMENT 13 Record	macro= DGA\$NETE DSECT= \$\$\$REC
ADJACENT NODE Record	macro= DGA\$NETD DSECT= NNODEREC
ALT.COMM Record	macro= DGA\$NETL DSECT= ALTADDRH
LDNS Record	macro= DGA\$NET\$ DSECT= DNREC
APPLIDS Record	macro= DGA\$NETA DSECT= NAAPLREC
LUPOOL Record	macro= DGA\$NET@ DSECT= NDLUPREC
TCP.API Record	macro= DGA\$NETT DSECT= NTAPIREC
IPv6 Record	macro= DGA\$NET6 DSECT= N6TCPREC

Displacement values found in the composite network map record are from the top of the composite network map.

Note: Modifying fields in the composite network map record is prohibited.

Example of Created Control Block

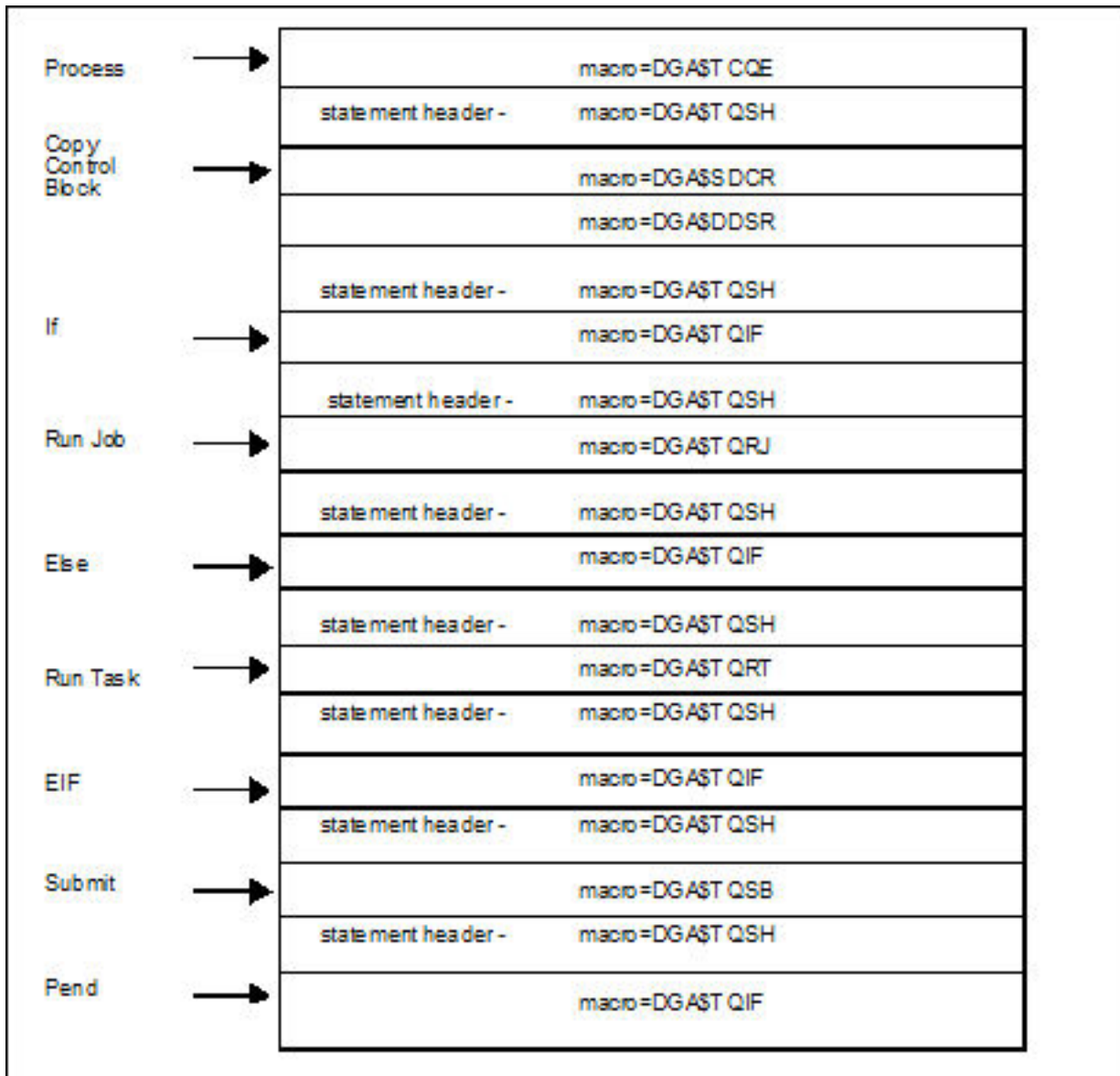
This sample section shows how a Process is submitted and the control block that is created when the Submit exit is invoked. The following figure shows the submitted Process.


```

TEST01 PROCESS SNODE=THERE
STEP01 COPY FROM (DSN=THIS.DATA.SET) -
      TO (DSN=THAT.DATA.SET DISP=OLD)
IF01 IF (STEP01=0) THEN
STEP02 RUN JOB (DSN=Z99.CONTROL(RUNJ))PNODE
      ELSE
STEP03 RUN TASK -
      (PGM=RTEXAMPL,PARM=(CL44' THIS.DATA.SET')) -
      PNODE
      EIF
STEP04 SUBMIT DSN=Z99.PROCLIB(TEST02) HOLD=Y

```

The following figure illustrates the resulting layout of the Process control block after submitting the Process named TEST01.



Modifiable TCQE Fields

The following table describes TCQE fields that you can examine or modify using the Submit exit.

TCQE Field	Content
TQCBHLNG	contains the length of the entire TCQE header. This length added to the address of the TCQE gives the address of the TCQSH.
TQSTMTN	contains the number of statements in this Process.
TQUNODE	contains the symbolic node name for the submitter of this Process.
TQUID	contains the user ID for the submitter of this Process.
TQUPAS	contains the password for the submitter of this Process.
TQPUID	contains the security user ID at the primary node.
TQOPPAS	contains the old security password at the primary node.
TQNPPAS	contains the new security password at the primary node.
TQSUID	contains the security user ID at the secondary node.
TQOSPAS	contains the old security password at the secondary node.
TQNSPAS	contains the new security password at the secondary node.
TQRTNCD	contains the Process completion code. The user exit changes this field when an error is encountered in the exit or if the Process is no longer submitted upon return from the exit.
TQMSGID	contains the Process message ID. The user exit includes a message ID related to any return codes set in the exit.
TQCSPRD	contains the displacement to the first Process statement from the TCQE. This length added to the address of the TCQE gives the address of the TCQSH. Note: If TQGT64K in TQFLAGA is 1, this displacement must be multiplied by 16.
TQPARSES	contains the value of the maximum number of parallel sessions allowed for the SNODE when submitting a Process.
TQPRSBYT	contains parallel session class. See the following section for details.
TQPRSBIT	contains parallel session class. See the following section for details.
TQPROCNM	contains the name of the submitted Process.
TQSCHDTE	contains the Julian date the Process is scheduled to submit.
TQSCHTME	contains the time of day the Process is scheduled to submit.
TQSCHDAY	contains the day of the week that the Process is scheduled to submit.
TQPRTY	contains the priority for Process selection.
TQRETAIN	contains the retain status for the Process.
TQTODFLG	contains the following interval control flags: <ul style="list-style-type: none"> - If TQTODTD is on, a Process has a scheduled time and date it is submitted. - If TQTOTME is on, a Process has a scheduled time it is submitted. - If TQTODDAY is on, a Process has a scheduled day of the week it is submitted. - If TQTODINT is on, a Process is scheduled to run when a specified interval expires.
TQPNODE	contains the symbolic node ID of the primary node.
TQSNODE	contains the symbolic node ID of the secondary node.
TQSTATUS	contains the Process status.

An exception to the table entry TQPARSES occurs in the stage 1 Submit exit. The stage 1 exit runs in the user address space (API) and the network map associated with that address space is where this information is retrieved. The network map used by the API may not be the same network map used by the DTF. The stage 2 Submit exit runs in the DTF address space and is invoked for every submit that takes place; therefore, the stage 2 Submit exit is more reliable.

Conversion of Parallel Session Values

The session class value is stored in two bytes (TQPRSBYT and TQPRSBIT) in the TCQE. The specified class can be derived from these values. The following table shows a sample of the two bytes for the first 16 classes (maximum is 256).

TQPRSBYT	TQPRSBIT	CLASS
00	80	1
00	40	2
00	20	3
00	10	4
00	08	5
00	04	6
00	02	7
00	01	8
01	80	9
01	40	10
01	20	11
01	10	12
01	08	13
01	04	14
01	02	15

ALLOCATION EXIT

The IBM Connect:Direct allocation exit provides an interface to a user-written program. If you supply a user exit in the initialization parameters, IBM Connect:Direct invokes the exit prior to any allocation activity by the receiving IBM Connect:Direct. Through the exit, you can change information that IBM Connect:Direct uses during the allocation Process. You can examine or modify information such as data set name (DSN) and type record name or set fields to terminate the copy step prior to allocation.

Sample Allocation Exits

IBM Connect:Direct provides the following sample allocation exits in the \$CD.SDGASAMP library.

Exits	Description
DGAXADSN	This exit documents the path of the data set name and unit that receives data during a COPY. It runs just prior to data set allocation. It is invoked for every COPY step on the receiving end of a transfer.

Exits	Description
DGAXUNIQ	<p>This exit creates a unique z/OS PDS member name, if the data set name specified in the COPY TO statement is found in the PDS directory.</p> <p>Each request can only specify one member name. The COPY TO statement must specify the member name. The COPY TO statement must also specify SYSOPTS="UNIQUE=YES".</p> <p>The exit only supports copying sequential files to a PDS member.</p>
DGAXALOX	<p>This exit enables a Data exit to be invoked for any or every copy performed by IBM Connect:Direct.</p>
DGAXA390	<p>This exit converts UNIT=3390 to UNIT=SYSDA.</p>
DGAXALEX	<p>This exit shows how to access the VSAMPL and the TCQSH, and both the source and destination description in the TCQSH. It shows how to change a value in the Data Set Description Control Blocks (DGA\$DDSR or DGA\$SDCR) and set a return code and message ID before return.</p>

Exits	Description
DGAXARCL	<p>In a COPY Process, this exit checks if a data set is archived or migrated. If the data set is archived, the exit requests retrieval and tells the COPY Process to go to the Timer Retry queue (TI RE), from which it is retried based on the ALLOC.WAIT and ALLOC.RETRIES initialization parameters. To ensure that the serially reusable operating system SYSZTIOT resource is freed up for subsequent users and does not cause the IBM Connect:Direct region to hang in a wait condition, use the DMGALRCL exit and specify ARCH as the value for the ALLOC.CODES initialization parameter. Change the setting of the INVOKE.ALLOC.EXIT global initialization parameter to BOTH if using this exit. If running with the CA-DMS product, you must modify the sample exit code as described in the comments at the beginning of the sample code. If you use this exit, add the following DD Statement to the DTF JCL:</p> <pre>//READER DD SYSOUT=(A,INTRDR)</pre> <p>Note: This exit is not necessary to process migrated or archived data sets. IBM Connect:Direct processes recalled data sets synchronously and the COPY Process remains in the Execution queue instead of being diverted to the Timer Retry queue and potentially to the Hold queue should retry limits be exceeded.</p> <p>In releases prior to Version 4.7, when the DGAXARCL (alias of DMGALRCL) allocation exit attempted to recall a migrated or archived data set offline, the Process went into fail state and was taken out of the Execution queue and put into the Timer Retry (TI RE) queue. As IBM Connect:Direct waited for the allocation and recall to be performed asynchronously, it would retry the Process based on the ALLOC.WAIT and ALLOC.RETRIES initialization parameters. If the Process exceeded the maximum time limit specified for retrying it (number of retries as specified by the ALLOC.RETRIES initialization parameter multiplied by the amount of time that IBM Connect:Direct waits between retries as specified by the ALLOC.WAIT initialization parameter), the Process was put into the Hold queue requiring manual intervention.</p> <p>In Version 4.7 and later, when IBM Connect:Direct executes a COPY Process without the DMGALRCL exit being present, it will use the ARCHRCAL macro synchronously, which means that the Process stays in the Execution queue not having to loop between the Execution, Timer, and Hold queues. The Process does not terminate while the recall operation is being performed but if the recall is unsuccessful, the COPY step produces a return code indicating the unsuccessful data set recall and instructs the user to correct the error and resubmit the Process.</p> <p>Although it is not needed for data recall, the DGAXARCL exit will continue to be supported and invoked prior to any allocation activity if the ALLOCATION.EXIT=DGAXARCL initialization parameter is specified. However, this exit is no longer required to process migrated or archived data sets.</p> <p>Caution: If the sample exit is not being used and a migration/recall product is not active, installed, or is temporarily down, the following messages are displayed:</p> <pre>ARC0050A DFSMSHSM IS NOT ACTIVE - START DFSMSHSM ARC0051A JOB xxxxxxxx WAITING FOR DFSMSHSM TO RECALL DSN=dsname *73 ARC0055A REPLY 'GO' OR 'CANCEL'</pre> <p>To proceed with the allocation, you must reply. Please refer to IBM documentation regarding the ARCxxxxA messages. If you reply CANCEL to the ARC0055A message, the Process completes with MSGID=SDE021CI. If 21C is in the ALLOC.CODES= list, the Process retries; otherwise, the Process terminates.</p>
DGAXPALL	This exit provides %PNUM substitution in a Process.

Restrictions and Requirements

Observe the following restrictions and requirements:

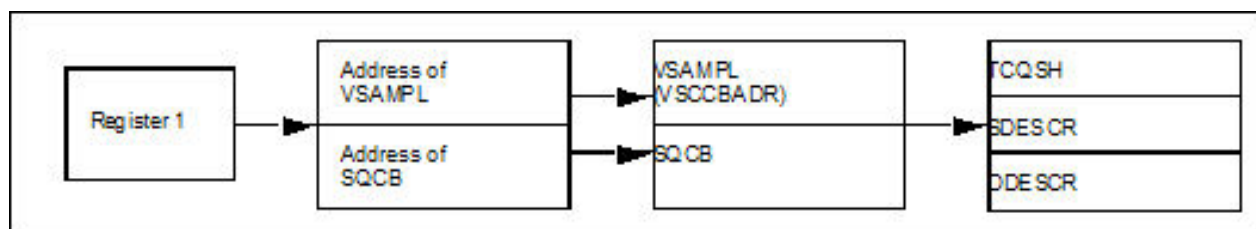
- The name of the allocation exit load module is user-defined, but it must not conflict with any other IBM Connect:Direct load module names.
- Because the control blocks provided by IBM Connect:Direct that the exit must access are located in storage requiring 31-bit addressability, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.
- To activate the exit, specify ALLOCATION.EXIT=modname in the IBM Connect:Direct initialization parameters file. You must link-edit the allocation exit as re-entrant and place it in a load library that the IBM Connect:Direct DTF can access.
- If an exit is not working, check the setting of the INVOKE.ALLOC.EXIT global initialization parameter, which should be one of the following:
 - RECV - Invokes the allocation exit when receiving a file.
 - SEND - Invokes the allocation exit when sending a file.
 - BOTH - Invokes the allocation exit both when sending and receiving a file. This is the default setting.

How the Allocation Exit Executes

The allocation exit executes in the DTF address space when the following conditions exist:

- The allocation exit is specified in the initialization parameters.
- A file is being received, and the Process step that initiated the copy is not in restart mode.

The following figure illustrates the structure of the parameter list for the allocation exit.



The following table is a list of the allocation exit parameters.

Parameter	Explanation
R1	Register 1 that contains the address of a standard parameter list upon entry into the user-written allocation exit
PLIST	Stands for standard parameter list
SQCB	Security control block
VSAMPL	Stands for VSAM parameter list, whose address is the first full word in the PLIST
VSCCBADR	Address of the Process step header and is contained in VSAMPL
TCQSH	Process step header portion of the Copy control block (each step of a Process generates a TCQSH)
SDESCR	Source data set descriptor portion of the Copy control block
DDESCR	Destination data set descriptor portion of the Copy Control Block (a sample of the DDESCR Control Block is included in this section)

The following macros map the control block structures.

Macro	Explanation
DGA\$VSMP	Macro that defines the VSAMPL control block
DGA\$TQSH	Macro that defines the TCQSH portion of the Copy control block
DGA\$SDCR	Macro that defines the SDESCR portion of the Copy control block
DGA\$DDSR	Macro that defines the DDESCR portion of the Copy control block

How to Calculate Addresses and Values

Upon entry into the user-written allocation exit, register 1 contains the address of a standard parameter list. The first entry in the PLIST contains the address of the VSAMPL. The second entry in the PLIST contains the address of the SQCB, if available. The VSCCBADR field in the VSAMPL contains the address of the Process step header, TCQSH. The SDESCR and DDESCR following the TCQSH is found by adding displacements to the TCQSH address.

- To calculate the location of SDESCR, add the length of TCQSH (TSHCBHLN) to the TCQSH address (VSCCBADR).
- To calculate the location of DDESCR, add the length of SDESCR (S1SVSLNG) to the SDESCR address.

All displacement values in the Copy control block are referenced from the beginning of the TCQSH control block.

SDESCR and DDESCR contain both fixed length fields and offsets to variable length fields. The allocation exit modifies any fixed length field in DDESCR.

The fields that are referenced in SDESCR and DDESCR using displacement values are variable in length. Do not modify them with the allocation exit.

The DSN field is created with enough space to hold a 100-character name only if the TSEXPDSN bit is on (set) in the TCQSH. If the TSEXPDSN bit is off in the TCQSH, then the Copy control block does not contain the room to expand the DSN. This lack of expansion room means that this copy originated from a IBM Connect:Direct node that did not build the copy control block with an expandable DSN field.

You can find the DSN field by adding D1DDSN to the address of the TCQSH. The DSN field contains a 2-byte length field followed by the DSN. Even though the field can be up to 100 bytes long, the 2-byte length field contains the actual length of the DSN. If you change the length of the DSN, you must modify the 2-byte length field accordingly. The other variable length fields are created with their current values and cannot be lengthened. Do not modify the D1DDSN field.

When allocating the destination file, IBM Connect:Direct first uses values from DDESCR, as specified in the IBM Connect:Direct COPY statement. Any values needed, but not set in DDESCR, are taken from the Type record, if one was specified. Any remaining values are taken from the SDESCR portion of the copy control block.

Note: If the D1DTYPE field is modified by the allocation exit, the exit must clear fields in the DDESCR portion of the copy control block that overrides the corresponding Type fields from the Type record.

Copy Control Block Definitions

Copy control block definitions are generated in the allocation exit program by including the macro name followed by DSECT=YES, as follows:

- DGA\$VSMP DSECT=YES
- DGA\$TQSH DSECT=YES
- DGA\$DDSR DSECT=YES
- DGA\$SDCR DSECT=YES

Copy Control Block Modifications

Modify only the VSAMPL and DDESCR control blocks. For the VSAMPL control block, you are required to modify the VSRTNCD and VSMMSGID fields as follows:

- Set the VSRTNCD field (and R15) to 0 to allow the Copy step to execute.
- Set the VSRTNCD field (and R15) to a nonzero value to fail the Copy step.
- Insert a message ID into the VSMMSGID field, if the VSRTNCD is set to a nonzero value. Take precaution to avoid duplicating existing message IDs.
- Place the message text corresponding to these message IDs in the IBM Connect:Direct message file.

DDESCR Control Block Format

The DGA\$DDSR member of the \$CD.SDGAMAC library lists the DDESCR control block format. You can modify the control block fields listed in the following table. Do NOT modify any displacement fields listed in the member.

Note: Turning a flag on means setting the bit in the byte where the flag is located to one. Turning a flag off means setting the bit in the byte where the flag is located to a zero.



Attention:

Do not modify the fields D1DMEMB, D1DDSN, and D1DVOLN. These fields represent displacements to their corresponding values. However, after calculating the address of the values (by adding the displacement to the address of the TCQSH) IBM Connect:Direct can change the actual values. At the calculated address, you will find a halfword field representing the length of the data that follows. If you change the length of the data, you must also change this halfword to reflect the new length. If the displacement to one of these fields is 0, do not insert a value or displacement. For example, if D1DMEMB=0, no member name was specified and a member name cannot be inserted. Do not set on the D1MEMNAM flag if D1DMEMB=0.

The same applies to D1DDSN. For D1DVOLN, if the displacement is 0, you cannot modify this field or turn on the D1DVOLSER flag. Also, you cannot add volume serial numbers to this list. You can delete volume serial numbers from the list or change the volume serial number. If volume serial numbers are deleted, decrement the length field by 6 for each one deleted. If all volume serial numbers are deleted, make D1DVOLN=0 and turn off the D1VOLSER bit.

Field	Explanation	Use Instructions
D1DTYPE	entry in the IBM Connect:Direct type defaults file	N/A
D1BLKSIZ	block size	To use the block size in the destination data set description portion of the copy control block, set D1BLKSZE flag on. To use the block size in the Type record, set the D1BLKSZE flag off.
D1DEN	tape density	To use the tape density indicated in the destination data set portion of the Copy control block, set the D1DENSTY flag on.
D1DSORG	data set organization	To use this value, set the D1DSORGN flag on. To use the value indicated in the type record, set the D1DSORGN flag off.
D1LRECL	logical record length	To use this value, set the D1LRECLN flag on. To use the value indicated in the type record, set the D1LRECLN flag off.
D1RECFM	record format	To use this value, set the D1RECFMT flag on. To use the value indicated in the type record, set the D1RECFMT flag off.
D1RKP	relative key position	To use this value, set the D1RKYP flag on.
D1TRTCH	7-track recording mode	To use this value, set the D1TRKTCH flag on.

Field	Explanation	Use Instructions
D1LABTYP	label type	To use this value, set the D1LABEL flag on.
D1RETPD	retention period	To use this value, set the D1RETPRD flag on. To use the value indicated in the D1EXPDT field, set the D1RETPRD flag off.
D1EXPDT	retention period	To use this value, set the D1EXPDTE flag on.
D1PRILOC	primary allocation amount	All of the following bits must be set off to use the space allocation values specified in the type record: D1TRK, D1CYL, D1BLK.
D1SECLOC	secondary allocation amount	To use this value, set the D1SECALL flag on. Set all of the following bits to off to use the space allocation values specified in the type record: D1TRK, D1CYL, D1BLK.
D1DIRBLK	number of directory blocks	To use this value, set the D1DIRBLK flag on. Set all of the following bits to off to use the space allocation values specified in the type record: D1TRK, D1CYL, D1BLK.
D1UNIT	unit type or group name	To use this value, set the D1GRPTYP flag on. To use the value indicated in the type record, set the D1GRPTYP off. Set all of the following bits to off to use the UNIT value specified in the type record: D1UNCNT, D1P, D1DEFER, D1GRPTYP.
D1VOLSEQ	volume sequence number	To use this value, set the D1VOLSQ flag on.
D1VOLCT	volume count	To use this value, set the D1VOLCNT flag on.
D1PASSWD	data set password	To use this value, set the D1PWD flag on.
D1DMEMB	displacement to the member name field†	N/A
D1DDSN	displacement to the data set name field†	N/A
D1DVOLN	displacement to the volume serial number†	N/A
† See the Caution that precedes this table.		

I/O Exit

The IBM Connect:Direct I/O exit provides an interface to user-written programs, allowing them to read and write data to or from a file whose organization IBM Connect:Direct does not support or would improperly access. Examples are internal format access to CA-LIBRARIAN or CA-PANVALET files.

Note: Checkpoint/restart is not supported for I/O exits.

If you plan to use an I/O exit, consider the following items:

- All I/O exits must be re-entrant, AMODE 31, follow IBM Assembler linkage standards, and reside in an authorized load library on the node where they are referenced. These exits must not alter any IBM Connect:Direct control block fields (except in the EXTCB as indicated in [I/O Exit Access to Control Blocks](#)). If other IBM Connect:Direct control block fields are altered, the results are unpredictable.
- If an ALLOCATION EXIT is specified, it is not given control when the COPY statement contains an IOEXIT keyword.
- Add any message IDs specified by an I/O exit to the IBM Connect:Direct Message file. See [Custom Messages in the Message Library](#) for instructions.

- Return from the I/O exit in the AMODE under which it was called. For example, if the I/O exit is called in 31-bit mode, the return must be in that mode. Therefore, if IBM Connect:Direct is running on an XA system, return from an I/O exit through a Branch Set Mode (BSM) instruction rather than a Branch (BR) instruction.

After you write the I/O exit to satisfy your specific data set requirements, implement it by specifying the exit name on the IOEXIT keyword on a COPY statement, or on the INSERT and UPDATE TYPE file commands.

Sample I/O Exit

The following sample I/O exit is provided in \$CD.SDGASAMP library.

Exits	Description
DGAXIOX1	This exit processes external data sets whose formats are not supported. It allocates, opens, reads, closes and deallocates a sequential data set.

Note: The I/O exit, DGADSIOX, enables you to copy SMS-compressed data without decompressing and also provides support for copying wildcard-named files. However, you cannot modify this I/O exit. For more information, see *Utility Programs* in *IBM Connect:Direct for z/OS User Guide*.

Specifying the I/O Exit in the COPY Statement

Include the IOEXIT keyword on the COPY statement to indicate that an I/O exit is used. The IOEXIT format on the COPY statement follows.

Statement	Parameters
COPY	FROM (IOEXIT= exitname (exitname[,parameter,...]))
	TO (IOEXIT= exitname (exitname[,parameter,...]))

The subparameters of the IOEXIT parameter of the COPY statement are:

Parameter	Description
exitname	Name of the user-written program to receive control for I/O-related requests.
parameter	Parameter or list of parameters passed to the exit. Their format is the same format as those parameters which you can specify on the RUN TASK statement.

The IOEXIT keyword is valid in either the FROM or TO areas of the COPY statement. This capability enables you to specify a different user-written I/O exit on each side as illustrated in following example.

COPY	FROM	(PNODE, - IOEXIT=(INEXT01,C'DB0A05',X'0E')) -
	TO	(SNODE, - IOEXIT=OUEXT03)

If you specify an exit, it can ignore the values of other parameters in the COPY statement (the DCB information). This issue is beyond the control of IBM Connect:Direct. For information on using the I/O exit through the COPY statement, see the [Connect:Direct Process Language](#) help.

Specify the I/O Exit in the TYPE File

Another method of specifying an I/O exit is to include the IOEXIT keyword on the INSERT and UPDATE Type file commands. The format is the same as the COPY statement. If you specify an IOEXIT parameter on the COPY statement, it overrides any IOEXIT in the Type file entry. The type defaults record must reside on the copy side (source or destination) that references it.

I/O Exit Access to Control Blocks

On entry to the exit, register 1 (R1) contains the address of the pointer to the EXTCB (Exit Control Block). The parameter list addresses point to a 2-byte length followed by the value.

All parameter lists end with the high order bit on in the last address in the list. The macro DGA\$XTCB generates the EXTCB. DGA\$XTCB is supplied in the \$CD.SDGAMAC library.

I/O Exit Requests

The I/O exit is called with IBM Connect:Direct requests that are found in EXTOPER, which is a field in EXTCB. The following are the requests that the input and output I/O exits receive:

BEGIN Request

IBM Connect:Direct makes a BEGIN request to an I/O exit when it begins communication with the exit. The BEGIN is when the exit must allocate work areas in preparation for future requests and is the first request that an I/O exit receives.

OPEN Request

IBM Connect:Direct makes an OPEN request to an I/O exit when the exit allocates and opens the file. EXTDIR contains either S (Send) or R (Received) to indicate whether the file is to be read or written. The I/O exit uses EXTWKARA to anchor any storage obtained and set EXTMAXLN to the maximum record length.

INFO Request

IBM Connect:Direct makes an INFO request to an I/O exit when it wants the exit to retrieve the file attributes and place them into the INFO area (mapped by the DMINFO macro) which is pointed to by EXTVSWRK. These data set attributes are required by IBM Connect:Direct.

Set the following fields in the INFO control block. The values listed are an example of those needed for a sequential data set.

INBLKSZ	= F'80'	block size
INLRECL	= F'80'	record size
INTYPE	= CL4'PS'	data set organization
INRECFM	= CL4' '	blank
INUNIT	= CL8' '	blank
INBLKS	= F'0'	nulls
INUSEBLK	= F'0'	nulls
INBLKTRK	= F'0'	nulls
INTRKCYL	= F'0'	nulls
IN2NDRY	= 8C'0'	character zeros
INLOCTYP	= CL3' '	blanks

GET Request

IBM Connect:Direct makes a GET request to an I/O exit when it wants a record/block read into the buffer. EXTINLNG is set to the length of the data. EXTINARA points to the record obtained.

The exit must indicate normal END-OF-DATA condition to IBM Connect:Direct by returning a value of EXTRCEOD in EXTRTNCD. You may indicate other conditions by providing other values in the previously mentioned fields. The EXTRCEOD in EXTRTNCD values enable IBM Connect:Direct to issue messages that are added to the IBM Connect:Direct message file.

The IOEXIT must allocate a buffer for the input record/block. To determine if IBM Connect:Direct is expecting a record or block, the IOEXIT must examine the source LRECL, BLKSIZE, RECFM and destination LRECL, BLKSIZE, RECFM fields in the EXIT control block. If the source and destination

data set attributes match, then IBM Connect:Direct is expecting a physical block from the IOEXIT, otherwise a logical record is expected.

ADD Request

IBM Connect:Direct makes an ADD request to an I/O exit when it wants a record/block to be inserted. EXTOTLNG is set to the length of the data. EXTOTARA points to the new record/block.

IBM Connect:Direct always sends RECFM=VS and RECFM=VBS files in segments rather than records. If EXTSPAN is set to Y, the data pointed to by EXTOTARA has two segments. Refer to the IBM library of manuals for information on the format of spanned records. The buffer referenced by EXTOTARA contains a physical block if the source and destination data set attributes match; otherwise, it contains a logical record.

CLOSE Request

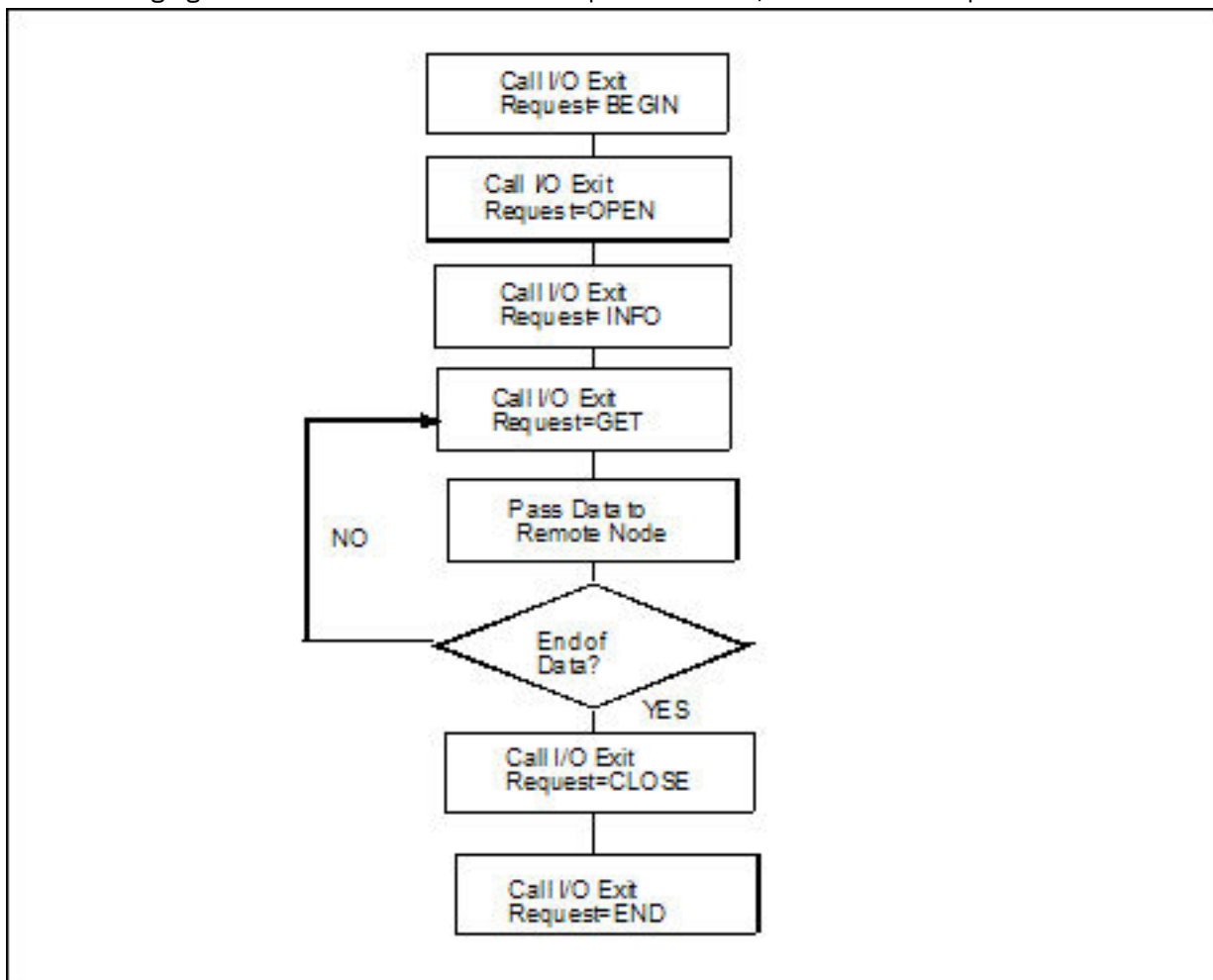
IBM Connect:Direct makes a CLOSE request to an I/O exit when the file closes. Errors returned by the exit on this request are ignored. The EXTABN flag is activated if the CLOSE request is due to abnormal termination.

END Request

IBM Connect:Direct makes the END request to an I/O exit to end communication with the exit. The exit releases any work areas allocated when it received the BEGIN request. This request is the last request an I/O exit receives.

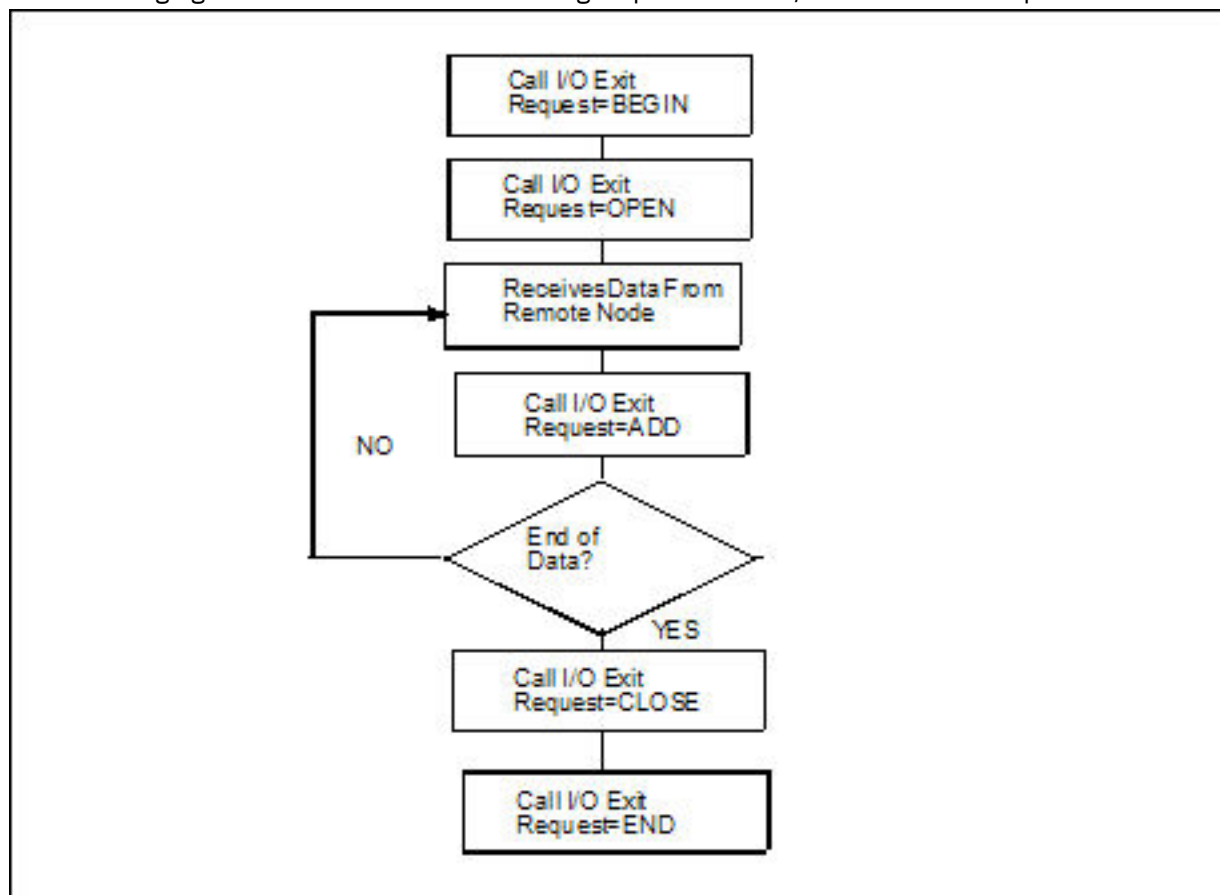
Normal Input Calling Sequence

The following figure illustrates the normal call sequence for an I/O exit used for input.



Normal Output Calling Sequence

The following figure illustrates the normal calling sequence of an I/O exit used for output.



Data Exit

The Data exit functions similarly to the I/O exit; however, the Data exit does not require the I/O management that the I/O exit requires. The Data exit provides an interface to user-written programs, allowing them to add, delete, change, or insert records.

The Data exit is called through the DATAEXIT parameter in the COPY statement or a keyword parameter supplied within the SYSOPTS string.

Note: Checkpoint/restart is supported for Data exits.

DATAEXIT Format

The DATAEXIT format in the COPY statement follows.

Statement	Parameters
COPY	FROM (DATAEXIT= exitname (exitname[,parameter,...]))
	TO (DATAEXIT= exitname (exitname[,parameter,...]))

The DATAEXIT subparameters are:

Parameter	Description
exitname	The name of the user-written program that receives control for data requests.
parameter	A parameter or list of parameters passed to the exit. See the RUN TASK statement.

The following example shows the DATAEXIT parameter in the COPY statement.

COPY01	COPY	FROM	(PNODE DSN=GJONES1.FROM.DSN DATAEXIT=(CD\$DGAXDXX01,CL6 'WEEKLY'))	-	-
		TO	(SNODE DSN=GJONES1.TO.DSN DCB=(DSORG=PS,LRECL=80,BLKSIZE=32000) DISP=(NEW,DELETE,DELETE) SPACE=(CYL,(1,0,0)) DATAEXIT=(CDDATAEX,CL44 'GJONES1.TO.DSN') UNIT=SYSDA)	-	-

The following example shows DATAEXIT used as a SYSOPTS parameter.

COPY01	COPY	FROM	(PNODE DSN=GJONES1.FROM.DSN SYSOPTS="DATAEXIT=(CD\$DXX01,CL6 'WEEKLY')")	-	-
		TO	(SNODE DSN=GJONES1.TO.DSN DCB=(DSORG=PS,LRECL=80,BLKSIZE=32000) DISP=(NEW,DELETE,DELETE) SPACE=(CYL,(1,0,0)) DATAEXIT=(CDDATAEX,CL44 'GJONES1.TO.DSN') UNIT=SYSDA)	-	-

Sample Data Exits

The following DATAEXIT samples are provided in \$CD.SDGASAMP library.

Exits	Description
DGAXDXX1	This sample Data exit examines or changes records from a COPY Process based on input data. You can use it to insert, replace, or delete records.
DGAXDXX2	This sample Data exit converts data from EBCDIC to ASCII if sending a file, or from ASCII to EBCDIC if receiving a file.

If you plan to use a Data exit, consider the following items:

- All Data exits must be re-entrant, follow IBM Assembler linkage standards, and reside in an authorized load library on the node where they are referenced. These exits must not alter any IBM Connect:Direct control block fields (except in the EXTCB as indicated in [Data Exit Access to Control Blocks](#)). If other IBM Connect:Direct control block fields are altered, the results are unpredictable.
- Add any message IDs specified by a Data exit to the IBM Connect:Direct Message file. See [Custom Messages in the Message Library](#) for instructions.
- Return from the Data exit in the AMODE under which it was called. For example, if the Data exit is called in 31-bit mode, the return must be in that mode. Return from a Data exit through a Branch Set Mode (BSM) instruction rather than a Branch (BR) instruction.

After you write the Data exit to satisfy your specific data set requirements, implement it by specifying the exit name on the DATAEXIT keyword on a COPY statement.

Data Exit Access to Control Blocks

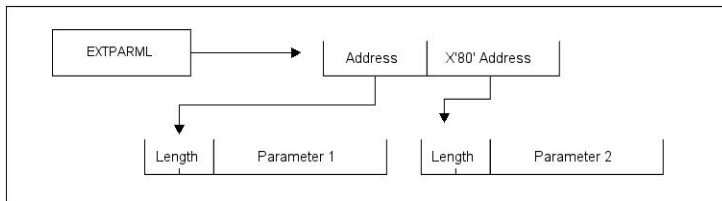
On entry to the exit, register 1 (R1) contains the address of the pointer to an 8-byte parameter list. The address consists of:

- +Pointer to a 4K storage area that is constant throughout the Data exit step. This area is mapped using the DXPARM label within the DGA\$VSMP macro in \$CD.SDGAMAC library.
- +Pointer to the EXTCB (mapped by the DGA\$XTCB macro in \$CD.SDGAMAC library).

If parameters are passed to the Data exit, the EXTPARML field in EXTCB points to a standard z/OS parameter list, pointing to the parameters (half-word length followed by the parameter itself) passed to

the Data exit. If no parameters are passed to the Data exit, EXTPARML points to a full-word field of binary zeros.

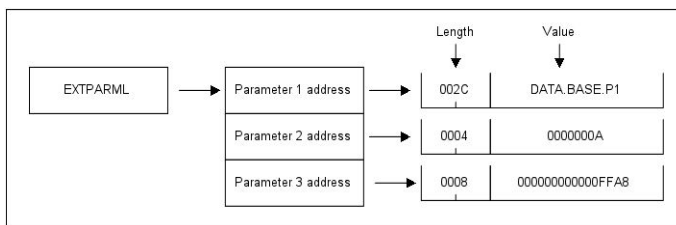
For example, if two parameters are passed to the Data exit, EXTPARML in EXTCB points to two full word pointers (the second pointer will have the high order bit on indicating the last parameter). Each of the pointers point to a half-word length followed by the parameter value as follows.



As another example, assume the following Data exit call.

```
DATAEXIT= (MYTASK,CL44'DATA.BASE.PI' -
          F'0010',XL8'FFAB')
```

Based on this call, the information passed to the exit program is displayed as follows.



To adhere to common linkage standards, IBM Connect:Direct sets the list termination bit (X'80') in the Parameter 3 address.

Data Exit Requests

The Data exit is called with IBM Connect:Direct requests that are found in EXTOPER, which is a field in EXTCB. Input and output Data exits receive the following requests:

Note: Upon return from the Data exit, any nonzero value in EXTRTNCD causes the Process to terminate with RC=8 and an SCPA049I message.

BEGIN Request

IBM Connect:Direct makes a BEGIN request to a Data exit when it begins communication with the exit. The exit is passed to a 4K work area that remains constant throughout this step in the Process. If additional storage is required, it can be obtained and anchored in the 4K work area.

OPEN Request

IBM Connect:Direct makes an OPEN request to a Data exit after the file is open and before the first data record is read.

GET Request

IBM Connect:Direct makes a GET request to a Data exit before adding the record to the buffer for transmission to the remote. At this point, the Data exit instructs IBM Connect:Direct to pass the record unchanged, change the record, delete the record, or insert records by setting the appropriate flag bit in EXTEAI in the EXTCB.

Flag bit setting	Action
All bits off	Pass the record unchanged

Flag bit setting	Action
EXTEAIRR	Replace the record. Update EXTINARA to point to the new record and EXTINLNG with the new record length.
EXTEAIRD	Delete the record.
EXTEAIRI	<p>Insert a new record. (The next call present the original record again and you can continue to insert records.) If pointing to a new record, set EXTEAIRR. Update EXTINARA to point to the new record and EXTINLNG with the new record length.</p> <p>If records are inserted in place of an original record, the original record delete the original record after the inserted records are passed.</p>

PUT Request

IBM Connect:Direct makes a PUT request to the Data exit before sending the record to the z/OS I/O system (IOS). At this point, the Data exit instructs IBM Connect:Direct to pass the record unchanged, change the record, delete the record, or insert records by setting the appropriate flag bit in EXTEAI in the EXTCB.

Flag bit setting	Action
All bits off	Pass the record unchanged.
EXTEAIRR	Replace the record. Update EXTOTARA to point to the new record and EXTOTLNG with the new record length.
EXTEAIRD	Delete the record.
EXTEAIRI	<p>Insert a new record. (The next call presents the original record again and you can continue to insert records.) If pointing to a new record, set EXTEAIRR. Update EXTOTARA to point to the new record and EXTOTLNG with the new record length.</p> <p>If records are inserted in place of an original record, delete the original record after the inserted records are passed.</p>

CLOSE Request

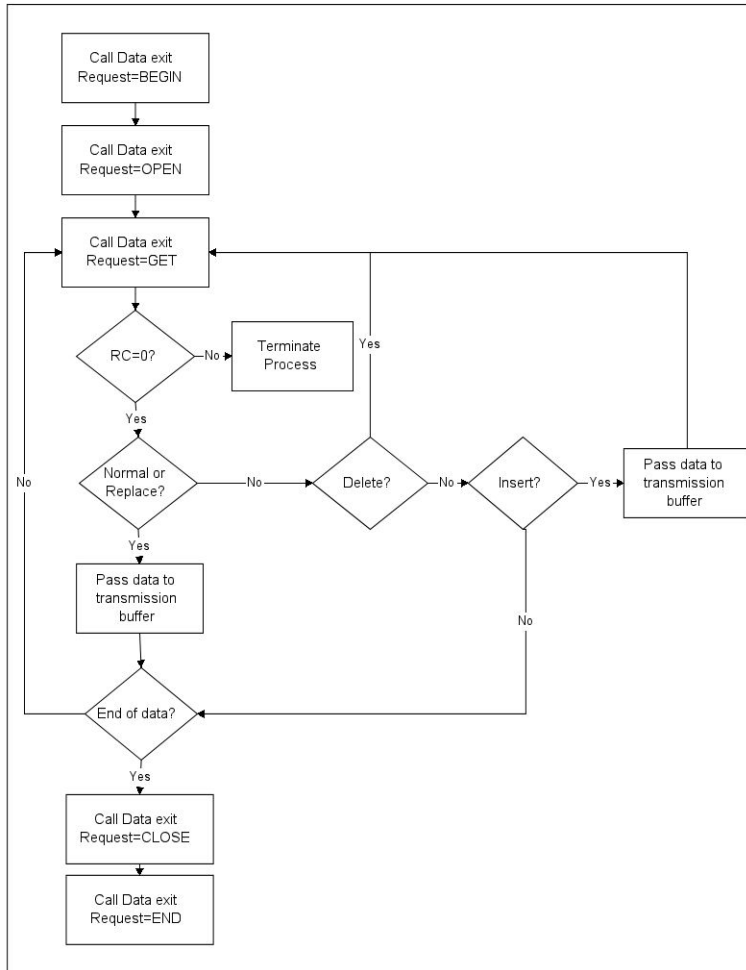
IBM Connect:Direct makes a CLOSE request to a Data exit when the file is to close.

END Request

IBM Connect:Direct makes the END request to a Data exit to end communication with the exit. The exit releases any work areas allocated when it received the BEGIN request. This request is the last request a Data exit receives.

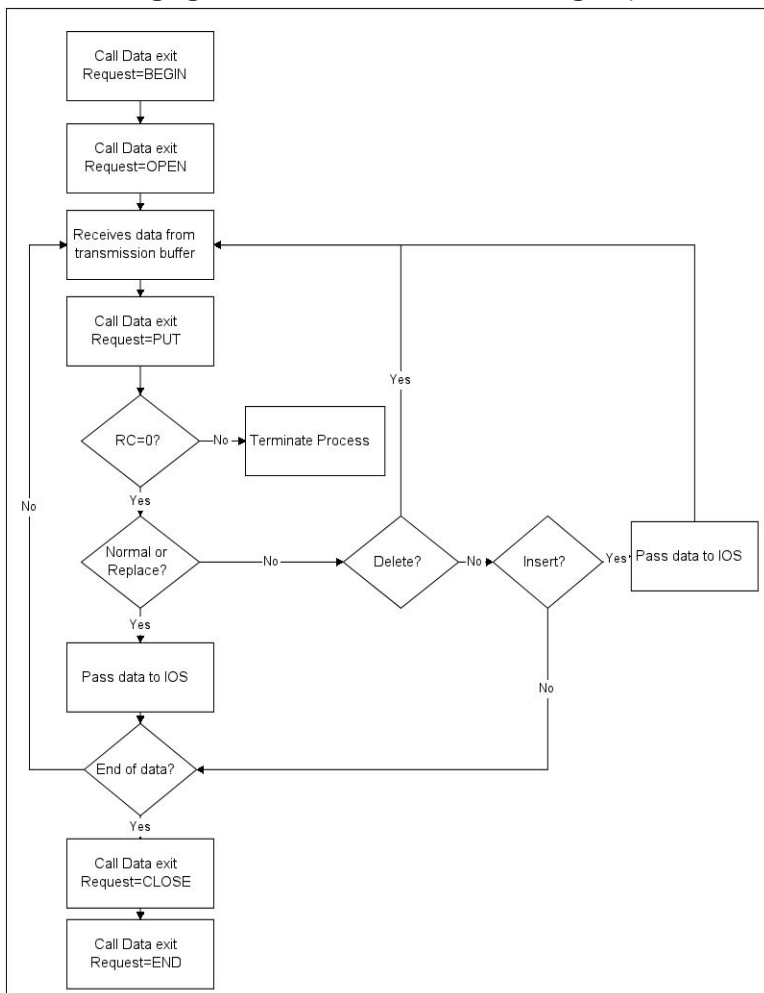
Normal Input Calling Sequence

The following figure illustrates the normal call sequence for a Data exit used for input.



Normal Output Calling Sequence

The following figure illustrates the normal calling sequence of a Data exit used for output.



WLM Exit

If you code the initialization parameter `CDPLEX.WLM.GOAL=YES`, IBM Connect:Direct invokes the `IWMWSYSQ` macro when necessary to query the status of systems in a sysplex. IBM Connect:Direct uses the information returned from the query to determine which system can best handle additional Process work. If you want to override the decision IBM Connect:Direct makes, you can specify a different choice with the WLM exit.

Activate the WLM exit by coding the following initialization parameter:

```
CDPLEX.WLM.GOAL=(YES,exitname)
```

Where `exitname` is the name of the WLM exit.

Exit Calling Convention

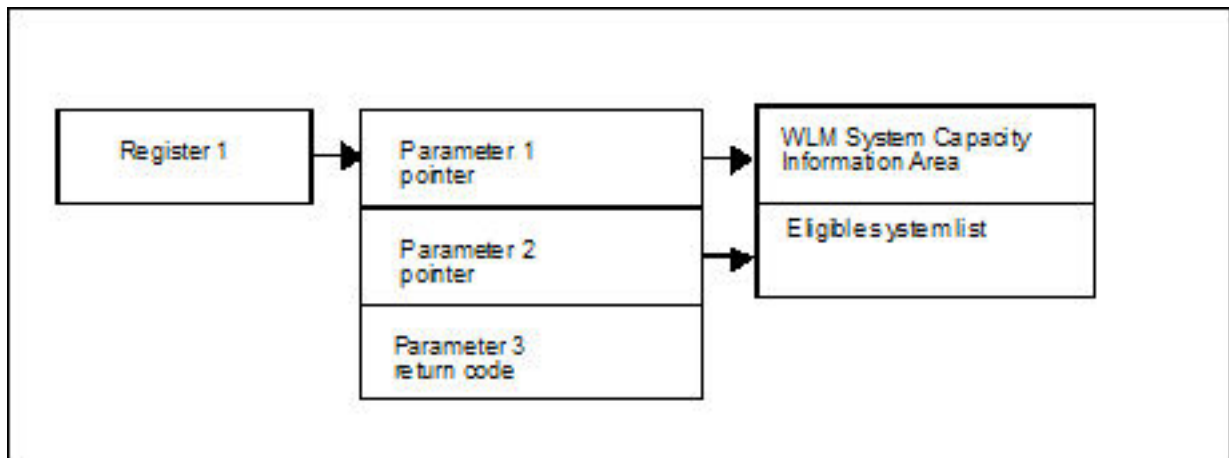
Three parameters are passed to the WLM exit by IBM Connect:Direct.

- The first parameter points to the WLM System Capacity Information Area (IWMWSYSI).
- The second parameter points to a list of system names that were found in goal mode by the WLM query. At least one of these systems must have a IBM Connect:Direct/Plex server active.

- The third parameter is used for the WLM exit return code. The following describes the possible return codes:

Return Code	Description
0	This indicates all systems are considered equal and no choice is made.
Negative value	This indicates that no systems are selected.
1-n	This indicates the chosen system by an index into the eligible system list.

The following figure depicts the information passed to the WLM exit:



Sample WLM Exit

The following sample WLM exit is provided in \$CD.SDGASAMP library.

Exit	Description
DGAXWLMX	This exit analyzes the IWMWSYSI info returned from an IBM Workload Manager query and indicates the least busy system with a return code.

Tapemount Exit

The IBM Connect:Direct tapemount exit provides an interface to StorageTek Tape Silo Software. If you supply a user exit in the initialization parameters, IBM Connect:Direct invokes the exit prior to a tape VOLSER mount request. Using the return codes resulting from this exit, you can obtain the status of the volumes needed to satisfy the mount request prior to the Tape premount message being displayed. (The TAPE.PREMOUNT = YES | NO | LIST parameter determines if a tape premount message will be displayed or not.) If any volume is not available for the Silo to process, the tape mount request is automatically cancelled and an exit return code of 8 or higher is issued to indicate that the Process is being held in error.

Exit Return Code	Explanation
0	The Tape Premount message will be suppressed.
4	The Tape Premount message will be issued.
8 or higher	The Mount Process will be "Held in Error."

Sample Tapemount Exit

IBM Connect:Direct provides the following sample tapemount exit in the \$CD.SDGASAMP.

Exits	Description
DGAXTAPX	This exit interfaces with the StorageTek Tape Silo via SLA macro calls. You can modify this member to work with other vendors' Tape Silo products.

You can use the sample JCL, DGAXATAP located in \$CD.SDGASAMP, to assemble and link-edit the exit.

Observe the following restrictions and requirements:

- The name of the tapemount exit load module is user-defined, but it must not conflict with any other IBM Connect:Direct load module names.
- Because the control blocks provided by IBM Connect:Direct that the exit must access are located in storage requiring 31-bit addressability, you must link-edit the module with AMODE ANY to make it capable of executing in 31-bit mode.
- To activate the exit, specify [“TAPEMOUNT.EXIT = modname” on page 486](#) in the IBM Connect:Direct initialization parameters file. You must link-edit the tapemount exit as re-entrant and place in a load library that the IBM Connect:Direct DTF can access.

The TAPEMOUNT.EXIT parameters are:

Parameter	Explanation
NOVOLS	Used to define the return code if all volumes for a tape mount request are not in the silo. Code with a value of 00, 04, or 08.
VIRTVOL	Tell the exit how to treat a virtual tape VOLSER. Code with a value of OKAY or ERROR.
TEST	Used to supply diagnostic test messages to a DD statement named to match the assembled program name. Code with a value of YES or NO.
UID	Optional local identifier which appears in NDMLOG output with the PTF maintenance listing.

Process Exit for Testing (DGAXPRCT)

The Process Exit for Testing (DGAXPRCT) allows you to perform the following functions:

- Test new applications and customer connections
- Prevent future production work from executing until testing is complete after you have terminated all active production work using the Flush Process command
- Resume regular production work after testing
- Control individual file transfers by application
- Enable and disable individual nodes and applications

While testing is being conducted, only Processes, particularly file transfers, involved with the testing activity are executed. No production data is transferred to applications being tested while at the same time no test data is transferred to production applications.

Flow of the DGAXPRCT Exit

First you tell DGAXPRCT which IBM Connect:Direct Processes to run and not run by storing your preferences as text records in a parameter table stored as a Partitioned Data Set (PDS) member. You can specify the following criteria for DGAXPRCT to use to find matches for one or more Processes to include (using the “I” command code) or exclude (“X” command code) from execution:

- A partial or full Process name
- A partial or full remote node name

- A partial or full IBM Connect:Direct submitter ID and submitter node combination
- A combination of Process name, remote node name and submitter ID/submitter node, all of which must match

In addition to telling IBM Connect:Direct which Processes to run, you tell the system what to do with the Processes which do not get executed. You can specify the following dispositions for Processes not permitted to run:

- Place the Process in the Hold queue
- Place the Processes in the Timer queue for session retry
- Flush the Processes from the queue

To see different ways the DGAXPRCT exit can be used, see [Sample Test Scenarios](#).

The DGAXPRCT exit is invoked by the Stage 2 Security exit before the security checks for a IBM Connect:Direct Process about to be executed have been performed. For information on how the Stage 2 Security exit is processed, see [Stage 2 Security Exit](#).

The Process Exit for Testing reads and validates the DGAXPRCT parameter table each time it is invoked when a Process is executed. If a syntax or other error occurs, IBM Connect:Direct places the Processes in the hold queue and returns a non-zero return code and error message ID. If the table is valid, DGAXPRCT scans the parameter table looking for a pattern that matches the Process that is about to be executed. If a match is found, the Process is permitted to execute if the "I" (Include) command code is in effect. If command code "X" (Exclude) is in effect, the process is not permitted to execute. If a match is not found in the table, DGAXPRCT performs the opposite processing from the case where a match is found, that is, if no match is found and command code "I" is in effect, the Process is not permitted to execute, whereas if command code "X" is in effect, the Process is permitted to execute.

Note: To reverse an action taken, use the "R" (Reverse) command code. If a match is found in an Include list and the "R" command code is also in effect, the Process is excluded. Conversely, if a match is found in an Exclude list and the "R" command code is also in effect, the Process is included.

If a Process is not to be permitted to execute, DGAXPRCT uses the disposition specified in the DGAXPRCT parameter table to either hold, retry, or flush the Process after the DGAXPRCT exit returns with a non-zero return code.

Note: For Processes initiated on remote nodes, the DGAXPRCT exit functions in the same manner as it does for Processes submitted on the local IBM Connect:Direct node. The DGAXPRCT Parameter Table is searched for a matching entry and the remotely initiated Process is either permitted to execute or excluded from execution. However, because the local node is the SNODE for this type of transfer, it cannot enforce the Process disposition setting in the DGAXPRCT Parameter Table. The remote PNODE determines how the Process is handled. Typically, the remote node places the Process in the Hold queue with a status of "HE" (Held in Error) if SECURITY.NOTIFY=HOLD is specified or the DGAXPRCT exit is supported on the remote node. If the remote node does not support SECURITY.NOTIFY=HOLD or the DGAXPRCT exit, the Process terminates.

When both the PNODE and the SNODE invoke the DGAXPRCT exit and the SNODE excludes a Process from executing, the PNODE automatically places the Process in the Timer queue for session retry regardless of the disposition setting in the DGAXPRCT Parameter Table on the PNODE. This processing is necessary because of technicalities in the handling of an SNODE error at the point in the IBM Connect:Direct protocol.

Setting Up and Using the DGAXPRCT Exit

To set up and use the DGAXPRCT Exit, complete the following steps. This roadmap is a high-level view of the procedure. For more information on each step, go to the section referenced in that step.

Procedure

1. Review [Overview of Security Options](#) for information on security including interfaces to other security software and sample exits.

2. Assemble the DGAXPRCT member in \$CD.SDGASAMP library.
3. To activate the stage 2 security exit which invokes the DGAXPRCT exit, specify SECURITY.EXIT=modname in the IBM Connect:Direct initialization parameters file (see SECURITY.EXIT = (module name, DATASET | ALL, PSTKT) | OFF SECURITY = (module name, DATASET | ALL, PSTKT) | OFF for information). You can modify the sample security exit, DGAMGSAF, provided in the \$CD.SDGASAMP library, to use as the base code for your stage 2 security exit.
4. Change the PROCEXIT parameter in the DGASECUR macro to DGAXPRCT. See [Enable the DGAXPRCXT Exit](#).
5. Create a PDS member to store parameters specifying how you want to implement the DGAXPRCT Exit, that is, preferences such as which Processes to run and not run and which queue to place unexecuted Processes. See [Edit the DGAXPRXCT Parameter Table](#).
6. Add DD statements to allocate the DGAXPRCT table and log information. See [Include Required DD Statements](#).
7. Reassemble and link-edit the Stage 2 exit source code. Because the control blocks provided by IBM Connect:Direct that the exit must access are located in storage requiring 31-bit addressability, you must link-edit the module with AMODE 31 to make it capable of executing in 31-bit mode. You must also link-edit the stage 2 security exit as re-entrant and place in an authorized library that the IBM Connect:Direct DTF can access.
8. Submit the startup jobstream to start IBM Connect:Direct.

Note: It is not necessary to restart IBM Connect:Direct when you modify the DGAXPRCT parameter table. The new settings are automatically in effect the next time a Process begins executing and invokes the DGAXPRCT exit, which reads the new values in the table. See [Reusing the DGAXPRCT Exit](#) for more information.

Enable the DGAXPRCXT Exit

The Stage 2 Security Exit executes the DGAXPRCT exit before it performs the necessary security checks for a IBM Connect:Direct Process about to be executed. To enable the Security Exit to invoke the DGAXPRCT exit, you must change the PROCEXIT parameter in the DGASECUR macro from NO to DGAXPRCT, and reassemble the exit. See [DGASECUR Parameters](#) for more information on this and all parameters in the DGASECUR macro.

Include Required DD Statements

Make sure your startup JCL includes the following DD statements:

```
//NDMPXTBL DD DSN=$CDPREF..PRCXTLIB(&NDMPXMEM),DISP=SHR
//USRINFO DD SYSOUT=*
```

The first DD statement allocates the parameter table PDS member while the second DD statement allocates a SYSOUT data set to which user-defined information from User Exits, such as error messages for parameter records incorrectly formatted and matching entries for Processes which run, is logged.

The CONNECTX JCL startup member contains these DD statements. For more information on the startup JCL members, see [DD Statements in Startup JCL](#).

Edit the DGAXPRXCT Parameter Table

You can use the ISPF text editor to create the DGAXPRXCT Parameter Table which defines which IBM Connect:Direct Processes can and cannot run. This table is stored as a Partitioned Dataset (PDS) member.

You must preallocate the data set with the following attributes:

```
DSORG=P0
LRECL=80
RECFM=FB
BLKSIZE=multiple of 80
```

Each table entry or record consists of a single-character command code in column one. Most command codes have a parameter which begins in column two and varies according to the command code function.

Note: The order of the entries in the table is important; the first match stops the table scan and the action requested is taken (allow the process to execute, flush the process, etc.).

Command Code	Description	Subparameters/Examples
*	Comment line.	* Only run the following processes.
E	Enables DGAXPRCT. This command code must be the first non-comment entry in the table.	<p>The second column in this entry must contain one of the following values which indicates the disposition of a process if it is not allowed to run.</p> <p>H – Places the Process in the Hold queue with a status of HE (Held in Error)</p> <p>R – Places the Process in the Timer queue in session retry until number of retries is exceeded. Once this number is exceeded, the Process is placed in the Hold queue with a status of RH (Restart Hold).</p> <p>F – Flushes the process from the queue</p>
D	Disallows DGAXPRCT execution and fails Process execution with a non-zero (error) return code and message NPRX003E.	You can also leave the disposition code in column two to make it easier to change from "E" to "D" and vice versa without having to change column two to a blank for command code "D."
P	<p>Matches Processes based on a full or partial Process name. Supports the wild card trailing asterisk (*). Can be used to enable or disable Process execution for a particular application by using naming conventions to match an application.</p> <p>In addition, remote node and/or submitter@submitter-node can be specified to further qualify the match.</p> <p>The combined format is:</p> <p>Pprocnam[,remote-node[,submitter@submitter-node]]</p>	<p>PCOPY – Matches a single Process</p> <p>PEOM* – Matches all Processes beginning with "EOM" for the End of Month Processing application</p> <p>P* – Matches all Processes\</p> <p>PCOPY,RNODE,SUE@NODE1 – Matches Process COPY that was submitted by SUE on NODE1 and whose remote node is RNODE.</p> <p>PCOPY,,SUE@NODE1 – Matches Process COPY that was submitted by SUE on NODE1.</p> <p>PCOPY,RNODE – Matches Process COPY whose remote node is RNODE.</p> <p>P*,*,SUE@NODE1 – Matches all Processes submitted by SUE on NODE1.</p>

Command Code	Description	Subparameters/Examples
R	<p>Reverses the action to be taken on a match.</p> <p>If the match is found in an include list, the Process is excluded.</p> <p>If the match is found in an exclude list, the Process is included.</p> <p>The combined format is:</p> <p>Rprocnam[,remote-node[,submitter@submitter-node]]</p>	<p>The syntax is the same as for the "P" command code.</p> <p>In this example, all Processes whose remote node begins with RNODE are excluded from execution except those whose remote node is RNODE3.</p> <pre>X R*,RNODE3 NRNODE* L</pre>
N	<p>Matches Processes based on a full or partial remote node name. Supports the wild card trailing asterisk (*).</p>	<p>NCD.NODE1 – Matches a single remote node name</p> <p>NCD.NODEA* – Matches all remote node names beginning with "CD.NODEA"</p> <p>N* – Matches all remote node names</p> <p>Note: P*,CDNODE1 is equivalent to NCDNODE1 and can be specified as R*,CDNODE1 to reverse the action.</p>
S	<p>Matches Processes based on a full or wild card IBM Connect:Direct submitter ID and a full or partial submitter node combination. The format is <id>@<node>.</p>	<p>SACTQ0ACD@TPM002 – Matches a specific submitter ID and node combination.</p> <p>S*@TPM002 – Matches all submitter IDs from node TPM002</p> <p>SACTQ0ACD@* – Matches submitter ID ACTQ0ACD from all nodes</p> <p>SACTQ0ACD@TPM* - Matches submitter ID ACTQ0ACD from all nodes beginning with "TPM"</p> <p>S*@* – Matches all submitter IDs from any node. This is another way to match all Processes.</p> <p>Note: P*,SUE@NODE1 is equivalent to SSUE@NODE and can be specified as R*,SUE@NODE1 to reverse the action.</p>
I	<p>Includes Processes for execution that match the patterns in the table which follow this command code. Either "I" or "X" must be the second non-comment entry in the table. Processes which do not match a pattern in the table are not executed.</p> <p>Note: To choose which command code to use to select Processes, determine which group is smaller and use the corresponding command Code. For example, if the number of Processes to be executed is smaller than the number of Processes to exclude from execution, specify "I" as the command code and add patterns to match that group of Processes.</p>	<pre>ER I NCD.BOSTON</pre> <p>Includes for execution only those Processes whose remote node is CD.BOSTON. Excluded Processes are placed in the Timer queue in session retry</p>

Command Code	Description	Subparameters/Examples
X	Excludes from execution those Processes that match the patterns in the table which follow this command code. Either “X” or “I” must be the second non-comment entry in the table. Processes which do not match a pattern in the table are executed.	EH X SDALLASOPS@* Excludes Processes for execution submitted by DALLASOPS from any node
L	Last entry in table.	

Sample Test Scenarios

The following examples show different applications of the DGAXPRCT exit using DGAXPRCT Parameter Tables to define which IBM Connect:Direct Processes to run and not run.

Specify Which Processes Run

In this example, IBM Connect:Direct executes all Processes that start with ACH or are named DITEST01 or DITEST02. All other Processes are placed in the Hold queue.

```
* Enable processing. Only permit processes matching one of the patterns
* to execute. Hold processes that don't execute.
EH
I
PACH*
PDITEST01
PDITEST02
L
```

Specify Which Processes to Exclude

In this example, IBM Connect:Direct does not execute any Process that starts with ACH or is named DITEST01 or DITEST02. All other Processes are executed.

```
* Exclude matching processes. Permit all others to execute.
EH
X
PACH*
PDITEST01
PDITEST02
L
```

Permit Process Execution by Remote Node and Submitter User ID/Node

In this example, IBM Connect:Direct executes all Processes that match one of the following criteria:

- The remote node name is DI.NODE1
- A remote node whose name starts with DI0017
- Any IBM Connect:Direct submitter ID from node DI0049
- The specific IBM Connect:Direct submitter ID ACHAPP from any node

All Processes not matching one of the above criteria are flushed from the queue.

```
* Only permit matching processes to execute. Flush those that do not.  
EF  
I  
NDI.NODE1  
NDI0017*  
S*@DI0049  
SACHAPP@*  
L
```

Combine Matching Criteria for a More Specific Match

In this example, IBM Connect:Direct executes all Processes that match one or more of the following criteria:

- Processes that begin with XYZ
- Processes that begin with ABC whose remote node is NODE1

All Processes not matching one of the above criteria are flushed from the queue.

```
* Only permit matching processes to execute. Flush those that do not.  
EF  
I  
PXYZ*  
PABC*,NODE1  
L
```

Use the "R" (Reverse) Matching Criteria

In this example, IBM Connect:Direct performs the following actions:

- Executes all Processes that begin with XYZ
- Executes all Processes that begin with ABC whose remote node is NODE1
- Excludes all Processes that begin with ABC and whose remote node is NODE1 when submitted by JOE@NODE2 from execution

All other Processes are flushed from the queue.

```
* Only permit matching processes to execute. Flush those that do not.  
EF  
I  
PXYZ*  
RABC*,NODE1,JOE@NODE2  
PABC*,NODE1  
L
```

Note: When using the "R" (Reverse) matching criteria, always specify the most specific matching criterion first and the most generic matching criterion last.

Stop the DGAXPRCT Exit

In this example, the DGAXPRCT Exit will exclude Processes from being executed, and display a non-zero return code signifying an error along with message ID NPRX003E . The remainder of the table is ignored (including the "F" code to flush Processes from the queue) and all Processes are placed in the Hold queue.

To resume testing and use the DGAXPRCT exit again, change the "D" command code to an "E."

```

* Execute no processes at all. Put them in the hold queue and return.
DF
I
PACH*
PDITEST01
PDITEST02
L

```

Reusing the DGAXPRCT Exit

To facilitate the use of different testing scenarios, you can maintain multiple members in the DGAXPRCT Parameter Table PDS. To reuse the basic DGAXPRCT Parameter table but modify it to change the Processes which run and do not run, follow this procedure. It is not necessary to restart IBM Connect:Direct; the next time a Process begins executing, the new settings will be in effect when the DGAXPRCT exit reads the table.

About this task



CAUTION: To control the execution of the DGAXPRCT exit, make sure that only authorized operators can modify the DGAXPRCT Parameter Table PDS member using your security system.

Procedure

1. Open the DGAXPRCT Parameter Table using the ISPF text editor.
2. Delete the current contents of the table member.
3. Copy another member for the next testing scenario you want to use and modify it as needed.
4. Save the DGAXPRCT Parameter Table PDS member.

The next time the DGAXPRCT exit is invoked when a Process begins executing, the DGAXPRCT exit uses this updated table.

DGAXPRCT Output

This topic contains example JOBLOG and USRINFO data set output resulting from the execution of the DGAXPRCT exit.

Example JOBLOG Output

This example shows JOBLOG output when the DGAXPRCT exit is executing.

```

SVTM055I SESSION (001) ESTABLISHED WITH          SNODE=WWW_TCP
NPRX000I ### Permitted: TSTRUN (00000007) SNODE=WWW_TCP
SVTM036I PROCESS STARTED TSTRUN (          7) SNODE=WWW_TCP
IGD103I SMS ALLOCATED TO DDNAME NDM00027
SVTM052I C001 COPY TSTRUN (          7) SNODE=WWW_TCP
SVTM052I COMPLETED 00000000/SCPA000I
SVTM052I FROM /u/output/testfile7
SVTM052I TO DALLAS.0.TESTFILE.BENCH.M50
SVTM052I COMPLETED 00000000/SCPA000I
SVTM037I PROCESS ENDED TSTRUN (          7) SNODE=WWW_TCP
SVTM056I SESSION (001) TERMINATED WITH          SNODE=WWW_TCP
SVTM055I SESSION (001) ESTABLISHED WITH          SNODE=WWW_TCP
NPRX109E ### Not executed: XSTRUN (00000009) SNODE=WWW_TCP
SVTM056I SESSION (000) TERMINATED WITH          PNODE=WWW_TCP

```

Example USRINFO Output

These examples show USRINFO output that is written to the USRINFO dataset while the DGAXPRCT exit is executing. Each line of output has a timestamp and the hexadecimal address of the IBM Connect:Direct Task Control Area (TCA) under which the DGAXPRCT exit is executing. The TCA address permits correlating output lines when Processes execute concurrently.

The following example shows DGAXPRCT output based on matching the specific Process name, TSTRUN.

```
965630E0 DGAXPRCT V1.14 ENTERED: SQCB@=16987688
965630E0 ##### APPLID TABLE #####
965630E0 EF
965630E0 I
965630E0 PTSTRUN
965630E0 L
965630E0 ##### END APPLID TABLE #####
965630E0 SETTING DISPOSITION TO FLUSH.
965630E0 TABLE MATCH: PNAME,PNUM=TSTRUN ,00000007
965630E0 ENTRY=PTSTRUN
NPRX000I ### Permitted: TSTRUN (00000007) SNODE=WWW_TCP
965630E0 DGAXPRCT EXITED: SQCB@/RC/MSGID 16987688 /00000000 /NPRX000I
96561D60 DGAXPRCT V1.14 ENTERED: SQCB@=16990688
96561D60 ##### APPLID TABLE #####
96561D60 EF
96561D60 I
96561D60 PTSTRUN
96561D60 L
96561D60 ##### END APPLID TABLE #####
96561D60 SETTING DISPOSITION TO FLUSH.
96561D60 NO TABLE MATCH: PNAME,PNUM=XSTRUN ,00000009
NPRX101E ### No entry: XSTRUN (00000009) SNODE=WWW_TCP
96561D60 DGAXPRCT EXITED: SQCB@/RC/MSGID 16990688 /00000004 /NPRX101E
```

The following example shows DGAXPRCT output based on matching the names of all Processes that begin with TSTRUN using the wildcard *.

```
96562720 DGAXPRCT V1.14 ENTERED: SQCB@=16990688
96562720 ##### APPLID TABLE #####
96562720 EF
96562720 I
96562720 PTSTRUN*
96562720 L
96562720 ##### END APPLID TABLE #####
96562720 SETTING DISPOSITION TO FLUSH.
96562720 TABLE MATCH: PNAME,PNUM=TSTRUN ,00000010
96562720 ENTRY=PTSTRUN*
NPRX000I ### Permitted: TSTRUN (00000010) SNODE=WWW_TCP
96562720 DGAXPRCT EXITED: SQCB@/RC/MSGID 16990688 /00000000 /NPRX000I
96561D60 DGAXPRCT V1.14 ENTERED: SQCB@=16990688
96561D60 ##### APPLID TABLE #####
96561D60 EF
96561D60 I
96561D60 PTSTRUN*
96561D60 L
96561D60 ##### END APPLID TABLE #####
96561D60 SETTING DISPOSITION TO FLUSH.
96561D60 TABLE MATCH: PNAME,PNUM=TSTRUN2 ,00000011
96561D60 ENTRY=PTSTRUN*
NPRX000I ### Permitted: TSTRUN2 (00000011) SNODE=WWW_TCP
96561D60 DGAXPRCT EXITED: SQCB@/RC/MSGID 16990688 /00000000 /NPRX000I
```

The following example shows invalid entries in the DGAXPRCT Parameter Table.

```
96562720 DGAXPRCT V1.14 ENTERED: SQCB@=16990688
96562720 ### INVALID TABLE ENTRY:XTSTRUN*
NPRX005E ### Table format error encountered.
96562720 DGAXPRCT EXITED: SQCB@/RC/MSGID 16990688 /00000008 /NPRX005E
```

Custom Messages in the Message Library

You can load special user-defined messages into the IBM Connect:Direct message library. The sample format for the IBM Connect:Direct message source in the following figure is in the DGAXMSGC member of the IBM Connect:Direct sample library, \$CD.SDGASAMP.

Note: Use the exact format as follows. You cannot use comments.

```

DELETE =MSG00001
INSERT =MSG00001
MODULE =MSGSOURC
STEXT= This is an example of the short text message (one).
L01 = This is an example of the long text message (one). As
L02 = many as 12 lines may be used for the long text message.
L03 =
L04 =
L05 =
L06 =
L07 =
L08 =
L09 =
L10 =
L11 =
L12 =

```

Observe the following rules for variables and message IDs:

- The DELETE and INSERT variables are 1–8 characters.
- The MODULE variable is 1–64 characters.
- The STEXT and L01 through L12 variables are 1–63 characters.
- To insert a message ID, the INSERT, MODULE, STEXT, and L01 through L12 variables are required.
- To delete a message ID, the DELETE variable is required.
- To replace a message ID, the DELETE, INSERT, MODULE, STEXT, and L01 through L12 variables are required.

The job stream in the following figure is in the DGAJMSGL member of the IBM Connect:Direct \$CD.SDGAJCL library. After copying the message source into your message source library and making changes as needed, run this job stream to add your messages to the IBM Connect:Direct message library.

```

//JOBNAME          JOB (ACCT), 'NAME', CLASS=A, NOTIFY=TSOID, MSGCLASS=X
//MSGBUILD          EXEC   PGM=DMMMSGLOD, PARM=' $CDHLQ.MSGFILE '
//STEPLIB          DD   DSN=$CDHLQ.SDGALINK, DISP=SHR
//SYSOUT            DD   SYSOUT=*
//INPUT DD          DSN=$HLQ.SDGAMSGS, DISP=SHR

```

Note: For SMP/E installations, add the message source to the NMSGSRC target library as an SMP/E USERMOD. MSGLOAD JCL can then process it.

Make the following changes to the job:

- Change the PARM statement to reference the file name of the IBM Connect:Direct VSAM message file. This value is the same value specified in the MSGDSN initialization parameter.
- Change the STEPLIB DD card to reference the SDGALINK.
- Change the INPUT DD card to reference the message source text. A sample format is in the member DGAXMSGC in the IBM Connect:Direct sample library, \$CD.SDGASAMP.

Message IDs

In a IBM Connect:Direct/Plex environment, messages can originate from any IBM Connect:Direct/Server or from the IBM Connect:Direct/Manager so you should define a 2-character message ID that identifies which IBM Connect:Direct/Plex member originated the message. This message ID is displayed after the message number.

Define the message ID using the CDPLEX.MSGID initialization parameter.

The following example shows a message if the CDPLEX.MSGID value is set to S1. The message ID is highlighted in bold.

SVTM055I	S1	SESSION (001) ESTABLISHED WITH	SNODE=SC.DUB.TPYLA2
SVTM055I	S1	SESSION (001) ESTABLISHED WITH	PNODE=SC.DUB.TPYLA2
SVTM036I	S1	PROCESS STARTED MVSMVST3(1) PNODE=SC.DUB.TPYLA2
SVTM036I	S1	PROCESS STARTED MVSMVST3(1) SNODE=SC.DUB.TPYLA2

Customizing Submit Screens

About this task

When you type variables into a submit screen, the IUI builds a SUBMIT command, and the command goes to a dialog for handling. As delivered on the installation tape, the primary IUI panel, DGA@PRIM, directly invokes the IUI submit panel, DGA@UBMT, when the SB option is selected. You can construct a customized submit screen to contain customized submit options. All menus can contain as many choices as screen space permits. To customize the submit function, modify DGA@PRIM to invoke DGA@SM03 instead of DGA@UBMT, when the SB option is selected, and perform the following procedure.

The following general steps are illustrated in an example which describes how to create a custom SUBMIT screen that copies a file to the existing file at another site at noon every day. You are notified when the Process is complete.

Procedure

1. DGA@SM03 is a submit menu panel which contains only one menu option that invokes DGA@UBMT. You can modify DGA@SM03 to include new menu selections to invoke the new customized submit panels. This step is explained in [Step 1 - Modify the Existing Menu DGA@SM03](#).
2. Define at least one general purpose Process to be invoked by the new custom submit screen. This step is explained in [Step 2 - Define General Purpose Process](#).
3. Provide custom submit screens. These screens:
 - Process variables that are resolved as SUBMITS occurs.
 - Build a command on each screen that communicates with the IUI dialog routines.

This step is explained in [Step 3 - Provide a New Submit Screen](#).

Step 1 - Modify the Existing Menu DGA@SM03

To modify the IBM Connect:Direct Submit Menu, add the following in Submit Menu (DGA@SM03):

- A line in the BODY section to specify the new option (+ 2 = => COPY TO EXISTING FILE AT ANOTHER SITE EVERYDAY AT NOON)
- A line in the PROC section to specify what to do when that option is selected (2,'PANEL(CUSTSUBM)')

The following figure shows the menu displayed after you add information. Only the elements necessary to modify an existing menu are displayed. If the user selects Option 2 on the command line, IBM Connect:Direct gives control to the screen with the name CUSTSUBM and displays that screen.

```

)ATTR
"
)BODY
#UNODE          +          SUBMIT MENU          +&ZDATE
+CMD%= = > _ZCMD          +&ZTIME
#STEXT
+
%PROCESSES:
+
+ 1 = = >  SUBMIT A PROCESS
+ 2 = = >  COPY TO EXISTING FILE AT ANOTHER SITE EVERYDAY AT NOON
+
)INIT
"
)PROC
"
    &SEL = TRANS (TRUNC(&ZCMD, '.'))
    "
    "
    1, 'PANEL (DMISUBMT) '
    2, 'PANEL (CUSTSUBM) '
    *, '?'
    "
)END

```

Note: For SMP/E installation, implement all ISPF panel changes as SMP/E USERMODs to the NISPPLIB target library.

Step 2 - Define General Purpose Process

The second step in creating a customized submit screen is to define a Process that the custom Submit screen invokes. The following Process is named APROC.

```

APROC PROCESS SNODE=&SNODE NOTIFY=%USER
STEP1 COPY FROM (PNODE DSN=&DSN1 DISP=SHR) -
      TO (DSN=&DSN2 DISP=(SHR,KEEP))

```

The SNODE, source file name, and the destination file name specified in the BODY section of the custom submit screen are substituted into the symbolic fields currently in the PROCESS (SNODE, DSN1, and DSN2).

Step 3 - Provide a New Submit Screen

A new submit screen processes variables resolved during submission and builds a command to communicate with the dialog routines. Although the customized submit screen can have any appearance, use the following information to help you design your screens.

Procedure

1. Use the generic submit screen (DGAXCUST) found in the sample library, \$CD.SDGASAMP, as a base for creating the custom screen.
2. Use the least number of input fields necessary when creating the screen to accomplish Process submission.
3. Use existing variables from DGAXCUST, if possible. You can use any variable name, but fewer changes are necessary when you use the existing code. The following ISPF variables are used in the IBM Connect:Direct submit processing.

Variable	Explanation
&PNAME1	Name of Process to be submitted
&DSN	File name containing Process to be submitted
&SNODE	Secondary node name

Variable	Explanation
&H	HOLD specification
&R	RETAIN specification
&PR	Priority of the Process
&NEWNAME	New name for the Process being submitted
&CLS	Process class
&NOTIFY	Notify IBM Connect:Direct user ID
&PNODEID	Security user ID at PNODE
&PNODEPW	Current security password at PNODE
&PNODENPW	New security password at PNODE
&SNODEID	Security user ID at SNODE
&SNODEPW	Current security password at SNODE
&SNODENPW	New security password at SNODE
&DSYMBPARM	Symbolic variable specification
&STIME	Start time value
&SDATE	Start day/date value
&CMD1	Used in constructing command string
&CMD2	Used in constructing command string
&CMD3	Used in constructing command string
&CMD4	Used in constructing command string

Note: Do not change the &CMD1, &CMD2, &CMD3, and &CMD4 ISPF variables. The SUBMIT command string is built into these four variables.

The following figure shows the ATTR and BODY sections of the customized submit screen CUSTSUBM. The ATTR section is the same as in the sample base screen, DGAXCUST. As seen in the BODY section, FROMDSN and TODSN are variables that are symbolically substituted when the general purpose Process APROC is submitted.

```

)ATTR
+ TYPE(TEXT) INTENS(LOW) SKIP(ON)
:   TYPE(INPUT) INTENS(NON)
# TYPE(OUTPUT) INTENS(HIGH) JUST(ASIS) CAPS(OFF)
@ TYPE(OUTPUT) INTENS(LOW) JUST(ASIS)
¢ PAD(_)
)BODY
#UNODE                                + CUSTOMIZED SCREEN
+CMD%= > _ZCMD
#STEXT                                + TIME-&ZTIME
% COPY TO EXISTING FILE AT ANOTHER SITE + DATE-&ZDATE
%      EVERY DAY AT NOON                + JULIAN-&ZJDATE
+ FILE TO BE SENT FROM HERE
%   = > ¢FROMDSN                        +
+
+ NODE TO RECEIVE THE FILE
%   = > ¢SNODE                          +
+
+ RECEIVING FILE ON ABOVE NODE
+   = > ¢TODSN                          +

```


4. Make the necessary changes in the INIT section after deciding how you want to set up the screen. Initialize all INIT section variables to the appropriate default value. For CUSTSUBM, the INIT section is in the following figure.

```
)INIT
.ZVARS = '(V@SEC)'
&NXTHLP = DMJSBMT1
.CURSOR = &FROMDSN          /*CHANGED */
&UNODE1 = &UNODE1
&SPC = ''
..&V@SEC = 'N'
..IF (&PROC ~= 'Y')
    &STEXT = ''
/* &PNAME1 = ''             DELETED */
/* &DSN = ''                 DELETED */
    &SNODE = ''
/* &Q = ''                   DELETED */
/* &H = ''                   DELETED */
    &R = 'Y'                /*CHANGED */
/* &PR = ''                  DELETED */
/* &NEWNAME = ''             DELETED */
/* &CLS = ''                 DELETED */
    &STIME = '12:00'        /*CHANGED */
/* &SDATE = ''               DELETED */
/* &SYMBPARM = ''            DELETED */
    &FROMDSN = ''          /* ADDED */
    &TODSN = ''            /* ADDED */
/* IF (&UNODE = &LNODE)     DELETED */
    &NOTIFY = '%USER'
/* IF (&UNODE ~= &LNODE)    DELETED */
/* &NOTIFY = ''              DELETED */
    IF (&PROC = 'Y')
        IF (&PROC# ~= 'NONE')
            .MSG = IUSB000I
            IF (&STEXT = '')
                .HELP = &NXTHLP
            IF (&STEXT ~= '')
                .HELP = DMI@MSG
```

The following changes are made in the INIT section:

- The cursor field is changed to FROMDSN.
 - &R is changed to Y to indicate RETAIN=YES.
 - &STIME is changed to 12:00 to indicate STARTT=(,12:00).
 - &FROMDSN and &TODSN are added and initialized to blanks.
 - Lines which carry a DELETED comment are removed because they are no longer necessary.
5. Make the necessary changes in the PROC section after deciding how you want to set up the screen. Verify PROC section field values and build the command string to submit the Process. For CUSTSUBM, the PROC section is in the following figures.

```

)INIT
)PROC
  &PROC# = 'NONE'
  &CMD = &ZCMD
  &SEL = TRANS( TRUNC (&ZCMD, '.')
    SPF, 'PANEL(ISR@PRIM) NEWAPPL(ISR)'
    WHO, 'PANEL(DMI@WHO)'
    SW, 'PGM(DMICMD) PARM(&CB@)'
    M, 'PANEL(DMI@MSG)'
    AUTH, 'PANEL(DMI@AUTH)'
    ' ', 'PGM(DMICMD) PARM(&CB@)'
    *, '?' )
  &ZTRAIL = .TRAIL
  IF (&ZCMD = 'SW')
    &SPC = 'SLN'
  IF (&CMD = ' ')
    VER(&FROMDSN, NONBLANK)          /* ADDED */
    VER(&SNODE, NONBLANK)            /* ADDED */
    IF (&TODSN= ' ')                /* ADDED */
      &TODSN=&FROMDSN              /* ADDED */
/* VER(&PNAME1, NAME, MSG=IUSB001I)    DELETED */
/* IF (&PNAME1 ~= ' ')                DELETED */
/*   IF (&DSN ~= ' ')                 DELETED */
/*     .MSG = IUSB002I                 DELETED */
/*     .CURSOR = PNAME1               DELETED */
/* IF (&DSN = ' ')                   DELETED */
/*   VER(&PNAME1, NONBLANK, MSG=IUSB003I) DELETED */
/* VER(&Q, LIST, Y, N, MSG=IUSB022I)    DELETED */
/* VER(&H, LIST, Y, N, C, MSG=IUSB005I)  DELETED */
/* VER(&R, LIST, Y, N, I, MSG=IUSB006I)  DELETED */
/* VER(&V@SEC, LIST, Y, N, I, MSG=IUSB006I) DELETED */
/* VER(&PR, RANGE, 0, 15, MSG=IUSB007I)  DELETED */
/* VER(&NEWNAME, NAME, MSG=IUSB008I)    DELETED */
/* VER(&CLS, RANGE, 1, 255, MSG=IUSB009I) DELETED */
  &USER = TRUNC(&NOTIFY, 1)
  IF (&USER ~= %)
    VER(&NOTIFY, NAME, MSG=IUSB010I)
&PROC = 'Y'

```

```

&CMD1 = ' SUB PROC=APROC' /*CHANGED */
&CMD2 = ''
&CMD3 = ''
&CMD4 = &SYMBPARM
IF (&PNAME1 ~= ' ') /*CHANGED */
&CMD1 = '&CMD1 PROC=&PNAME1' /*CHANGED */
IF (&DSN ~= ' ') /*CHANGED */
&CMD1 = '&CMD1 DSN=&DSN' /*CHANGED */
IF (&SNODE ~= ' ')
&CMD1 = '&CMD1 SNODE=&SNODE'
/* IF (&Q ~= ' ') DELETED */
/* &CMD1 = '&CMD1 REQUEUE=&Q' DELETED */
/* IF (&H ~= ' ') DELETED */
/* &CMD1 = '&CMD1 HOLD=&H' DELETED */
IF (&R ~= ' ')
&CMD1 = '&CMD1 RETAIN=&R'
/* IF (&PR ~= ' ') DELETED */
/* &CMD1 = '&CMD1 PRTY=&PR' DELETED */
/* IF (&NEWNAME ~= ' ') DELETED */
/* &CMD1 = '&CMD1 NEWNAME=&NEWNAME' DELETED */
IF (&NOTIFY ~= ' ')
&CMD2 = ' NOTIFY=&NOTIFY'
&PARMX = ''
&PARMX2 = ''
/* IF (&SDATE ~= ' ') DELETED */
/* &PARMX = 'Y' DELETED */
/* &CMD2 = '&CMD2 STARTT=(, &SDATE, ' DELETED */
IF (&STIME ~= ' ')
IF (&PARMX ~= 'Y')
&PARMX = 'Y'
&PARMX2 = 'Y'
&CMD2 = '&CMD2 STARTT=(, &STIME'
IF (&PARMX2 ~= 'Y')
&CMD2 = '&CMD2 &STIME'
IF (&PARMX = 'Y')
&CMD2 = '&CMD2'
/* IF (&CLS ~= ' ') DELETED */
/* &CMD2 = '&CMD2 CLASS=&CLS' DELETED */
/* IF (&V@SEC = 'Y') DELETED */
/* &ZSEL = 'PANEL (DMIUSRID)' DELETED */
&CMD3= '&&DSN1=&FROMDSN' /*ADDED */
&CMD3= '&CMD3 &&DSN2=&TODSN' /* ADDED */
)END

```

The following changes are made in the PROC section:

- A verify is added for the &FROMDSN variable. It must be non-blank.
- A verify is added for the &SNODE variable. It must be non-blank.
- A test for blanks in &TODSN is added. If &TODSN is blank, it is set to &FROMDSN.
- &CMD1 is changed to contain the string SUB PROC=APROC, the command default.
- &CMD3 is added to contain the string &&DSN1=&FROMDSN. Symbolic substitution is accomplished with this addition. When APROC is submitted, &DSN1 is translated to whatever value is in the &FROMDSN file.
- The next-to-last line is added to concatenate the string &&DSN2=&TODSN to the contents of &CMD3. When APROC is submitted, &DSN2 is translated to the value in &TODSN.
- Lines which carry a DELETED comment are removed because they are no longer necessary.

If Y12.FROMHERE is the file to be sent, CD.THERE is the node to receive the file, Z12.TOHERE is the receiving file, and the Process is APROC, then the command string is built as follows.

```

SUB PROC=APROC SNODE=CD.THERE RETAIN=Y NOTIFY=%USER -
STARTT=(,12:00) &DSN1=Y12.FROMHERE &DSN2=Z12.TOHERE

```

When the Process APROC is submitted, it is resolved as follows.

```
APROC PROCESS SNODE=CD.THERE NOTIFY=%USER  
STEP1 COPY FROM (SNODE DSN=Y12.FROMHERE DISP=SHR) -  
TO (DSN=Z12.TOHERE DISP=(SHR,KEEP))
```

Administering Statistics

The Connect:Direct for z/OS statistics facility logs statistics to a series of VSAM file pairs. Each pair consists of an entry-sequenced file and a key-sequenced file, both with the REUSE attribute.

File Pair Configuration

The minimum configuration is two file pairs, or four files, however you can configure more than two file pairs. Specify the file pairs in the initialization parameters, STAT.DSN.BASE and STAT.FILE.PAIRS, as data set name high level qualifiers and the number of pairs. This specification determines the configuration of the statistics file pair list.

Within each file pair, IBM Connect:Direct writes the statistics records to the entry-sequenced file, while the key-sequenced file maintains index information about the records. On average, IBM Connect:Direct writes records to the key-sequenced file at the rate of about one for every two records written to the entry-sequenced file.

Retrieving Statistics with the SELECT STATISTICS Command

When you issue SELECT STATISTICS commands, the system locates the requested records by using the key-sequenced file as an index to the entry-sequenced file. All the file pairs defined to the DTF are available to SELECT STATISTICS command processing. IBM Connect:Direct searches any file pair that contains records satisfying the SELECT STATISTICS command, not just the files currently being written.

How Records Are Written

IBM Connect:Direct writes the statistics records to the entry-sequenced VSAM files in chronological order, starting at the beginning of the file and proceeding until the file or its paired key-sequenced file is full. The oldest record is always at the beginning of the file and the newest record is last. The system records each statistics record as a single VSAM record. The system does not compress the records or add control information.

When a file pair is full, the system switches to the next in the sequence, and begins writing to it. When the last file pair in the list is full, the system wraps back to the first pair in the sequence.

You can also specify the time of a file switch by using the STAT.SWITCH.TIME initialization parameter. For example, you can specify that statistics files switch at midnight every day, which limits a file pair to records from a single day. IBM Connect:Direct also provides a statistics switch API command that directs the DTF to perform a switch at any time.

When the system has written to all the pairs in the list, the system reuses the pairs. When a switch is made, the system closes the active pair and makes the pair with the oldest data the new active pair. When the system switches to a file, or a file becomes active, Connect:Direct for z/OS does a VSAM RESET. This VSAM RESET erases any records and index information in the active file. The system then writes new records starting at the beginning of the file.

Tools to Monitor the Statistics Facility

IBM Connect:Direct provides the following tools for monitoring the statistics facility:

- INQUIRE STATISTICS command
- S2 statistics records

- SS statistics records
- DGASSTAT utility

INQUIRE STATISTICS Command

The INQUIRE STATISTICS command gives a snapshot of the status of the facility. INQUIRE STATISTICS produces a report that includes the following:

- List of any currently EXCLUDEed record types
- File pair list configuration that includes which file pair is active
- Date and time range covered by each file pair
- Size of each file
- Utilization percentage of the entry-sequenced files
- Count of SELECT STATISTICS commands active against each file pair
- Indication if logging is waiting for SELECT STATISTICS to finish so a file pair can be reset
- Indication if logging is waiting for archive to finish so a file pair can be reset
- Reason for the last switch from each file pair
- Most recent file access return code and message ID for each file
- Utilization percentage of the nonactive key-sequenced files
- Indication of whether archive notification was received for the nonactive files

S2 Statistics Records

The S2 statistics records contain information about the statistics logging function. The system writes the records about once per hour when activity exists in the DTF. Each S2 record contains statistics about the period of time since the prior S2 record. The S2 statistics records include the following information:

- Beginning time and length of the period covered
- Count of records written in the period
- Count of ESDS control intervals written in the period
- Count of total bytes written to the ESDS
- Average statistics record length
- Average records per control interval
- Average ESDS writes per second
- Average KSDS writes per second
- Average logging service time
- Total waits for logging queue element
- Each indexed field including max keys and average keys per control interval

Use the TYPE parameter of the SELECT STATISTICS command to view the S2 records. The system writes the S2 records with the user ID specified in the STAT.USER initialization parameter. If you code a unique ID for STAT.USER and you specify the USER parameter on the SELECT STATISTICS request, you greatly reduce the search time because the user ID is an indexed field. See the *IBM Connect:Direct for z/OS User Guide* for more information on how to use the SELECT STATISTICS command.

For example, if you code STAT.USER=statuser, a SELECT STATISTICS request to display all S2 records looks like the following figure.

```
SELECT STATISTICS WHERE (USER=statuser, TYPE=(S2)) TABLE
```

SS Statistics Records

The SS statistics records contain information about SELECT STATISTICS processing. One SS record is written for each SELECT STATISTICS command that executes. The SS record includes information such as the index that IBM Connect:Direct uses to search the files and the number of requests issued to the keyed and entry-sequenced clusters. The record also includes the number of records examined and rejected.

Use the SELECT STATISTICS command with the TYPE parameter to view the SS record.

Using selection criteria with the SELECT STATISTICS request improves the performance by efficiently locating the requested records. For example, you can include the ID of the user that issued the SELECT STATISTICS command or the approximate time the request was issued, using the STARTT, STOPT, and USER parameters. The following figure shows an example using this selection criteria.

```
SELECT STAT WHERE      -
      (TYPE=(SS) USER=USER1 STARTT=(,NOON) STOPT=(,13:00)) TABLE
```

Using the DGASSTAT Utility to Determine File Usage

Use the DGASSTAT utility to find out the rate at which the system generates statistics records. DGASSTAT also performs an analysis of the contents of the statistics file showing what percentage of the records are of each record type. This utility runs as a batch job step, and analyzes a single statistics entry-sequenced file.

Use the DGASSTAT JCL to report on the statistics files. It calculates the average number of CIs used per day at one DTF.

The job stream example in the following figure, DGAJSTAT , is found in the \$CD.SDGAJCL distribution library.

```
//JOBNAME JOB (ACCT),CLASS=A
//*****
//*
//*      Connect:Direct
//*
//*      This JCL will invoke the utility to produce
//*      a report for a Statistics File.
//*
//*      Change "$CD" to the high-level qualifier(s)
//*      appropriate for your installation.
//*
//*      Change "$CDVSAM.STAT.ESDS01" to match the name
//*      of the Statistics file to be analyzed.
//*
//*****
//STEP1   EXEC PGM=DGASSTAT
//STEPLIB DD DISP=SHR,DSN=$CD.SDGALINK
//SYSOUT  DD SYSOUT=*
//ESDS    DD DISP=SHR,DSN=$CDVSAM.STATS.ESDS01 (ESDS of file pair)
```

Optimizing Statistics Files

Use this information to determine the most efficient use of Statistics file space.

About this task

Note: IBM Connect:Direct does not support extended-format, extended-addressing ESDS Statistics data sets.

In this example, the IBM Connect:Direct software is installed using the default statistics installation of two file pairs with a total capacity of 13,500 records. After running IBM Connect:Direct for a time, you determine that the records log for about 2.5 days before the file pair list wraps. The administrator wants to provide space for 7 days worth of records to be available at any given time. The administrator does the following:

Procedure

1. Use DGASSTAT to determine the number of records written daily.

Run the DGASSTAT utility against the statistics entry-sequenced clusters to determine the rate at which the system generates the statistics records.

For example, DGASSTAT shows that records per day is approximately equal to 5,400.

2. Determine the total capacity of the statistics file.

```
capacity = (records per day) * days
```

Determine the total capacity in this example by multiplying the 5,400 records per day by seven days. In this case, the total capacity of the statistics file is 37,800 records.

3. Determine the number of records per file pair.

In this example, the administrator decides to define four file pairs, so each are given a capacity of 9,500 records, for a total capacity of 38,000 records.

4. Determine the RECORDS parameter value for the key-sequenced clusters.

The number of KSDS records needed depends on both the number of records in the associated ESDS and a factor determined by the ESDS CISIZE. For more information, see *Statistic File Recommendation, IBM Connect:Direct for z/OS Configuration Guide*. Assume the factor in this case is 0.75 which corresponds to an ESDS CISIZE of 8K.

```
KSDS-records = 0.75 * (ESDS-records)
```

Determine the number of KSDS records by multiplying 75% by 9,500, the number of records per file pair. The RECORDS parameter value for the key-sequenced clusters is 7,125.

Results

Based on these calculations, the administrator allocates four file pairs. The entry-sequenced cluster (ESDS) of each pair is defined with RECORDS(9500). The key-sequenced clusters (KSDS) are defined with RECORDS(7125).

Change the File Pair Configuration

Make changes to the statistics files or to the configuration of the file pair list when the DTF is not running. During DTF execution, the files remain allocated by IBM Connect:Direct.

The restrictions that IBM Connect:Direct places on changes to the configuration maintain the integrity of the facility. At DTF initialization time, IBM Connect:Direct checks the usability, validity, and accessibility of the statistics files data.

File Pair Verification

IBM Connect:Direct performs a verification procedure at initialization, as follows:

- Within each file pair, IBM Connect:Direct verifies the appropriate sizing, relative to each other, of the entry-sequenced cluster and the key-sequenced cluster. If the file pair is not relatively sized, then IBM Connect:Direct issues a warning message and initialization continues.
- If either of the files of a pair has data, IBM Connect:Direct attempts to verify that the two files are actually a statistics file pair. IBM Connect:Direct verifies that the key-sequenced file really does contain index information for the associated entry-sequenced file.

IBM Connect:Direct uses control information maintained in the key-sequenced file to perform the verification. The software keeps a control record in the KSDS which contains the data set name and the

control interval size of the paired entry-sequenced file. If this information does not match, statistics initialization fails.

Changing the File Pair

Following are the implications of changing the file pair.

- Changing the control interval size of the ESDS or renaming the clusters causes initialization to fail because the control record in the KSDS no longer matches the files. The two ways to resolve this statistics initialization failure are:
 - First, you can use the DGADBKEY utility to rebuild the key-sequenced cluster. This utility recreates the KSDS control record so that it matches the new names or control interval size. The records in the file pair remain accessible when the DTF is available again.
 - The second solution is to empty the files. The file pair is available for logging new records. However, the old records are no longer available. You may want to archive the files before emptying them.
- Changing the size of a file pair is not a problem. The sizes of both files of a pair change together so that the relative sizes do not change.
 - If the files are made larger and the names remain the same, then copy the records from the old smaller entry and key-sequenced clusters to the new larger ones.
 - Use the same procedure to make a file pair smaller if all the existing records from both files will fit into the smaller space of the new files. If the existing records do not fit, then the new smaller file pair must be left empty initially, and the old records become unavailable.

File Pair List Verification

IBM Connect:Direct generates the statistics file pair list from the initialization parameters STAT.DSN.BASE and STAT.FILE.PAIRS. See [STAT.DSN.BASE = dsname base STAT.FILE.PAIRS = number](#) for an example of a file pair list.

The IBM Connect:Direct statistics facility processes the statistics file pair list in a circular, or wrap-around fashion. The system maintains statistics records in strictly chronological order both within each file pair, and with regard to the file pairs in the list.

At DTF initialization, unless STAT.INIT=COLD is specified in the initialization parameters, IBM Connect:Direct verifies that the order of the file pairs is valid. This verification is done by examining the date and time range of each non-empty file pair in the list. These must be in strictly ascending order throughout the list, except across the wrap point. Empty file pairs may appear anywhere in the list.

Changing the Number of File Pairs

It can be useful to periodically change the number of file pairs in the list. To change the number of file pairs, change the STAT.FILE.PAIRS initialization parameter which specifies the number of file pairs. This action adds to or removes from the end of the list.

- Add empty file pairs to the end of the list unless you specify STAT.INIT=COLD.
- Remove records from the end of the list by reducing the STAT.FILE.PAIRS value. When you remove these records, they become unavailable and can in some cases leave gaps in the statistics data. You may want to archive these records before removing them.

Archiving Statistics

Archiving refers to the process of copying the records from a statistics entry-sequenced cluster to another data set for long-term storage. The output of this process is an archived statistics file. You can write the archive file to a VSAM entry-sequenced cluster with the same characteristics as a statistics ESDS, or to a non-VSAM sequential file on DASD. The system does not store the statistics records in the ESDS in any special format. The system records each statistics record as a VSAM record in an ordinary VSAM ESDS. You can also write the archive file to a magnetic tape or a database table.

Archiving Using a Predefined Process

Using the DTF initialization parameter `STAT.SWITCH.SUBMIT`, you can specify that when the DTF switches from one statistics file pair to another, IBM Connect:Direct submits a predefined Process to archive the records in the previously active ESDS. IBM Connect:Direct submits this archive Process with a symbolic parameter indicating the data set name of the ESDS of the pair.

- The Process can then use IBM Connect:Direct to copy the data to another location. A sample archive Process, `DGAPSTAT`, is in the `$CD.SDGAPROC` distribution library.
- Alternatively, the Process can submit a batch job to archive the data using `IDCAMS REPRO`, or some other utility. Use the `DGADTSUB` utility to substitute the file data set name into the submitted job stream. You can invoke `DGADTSUB` through the `RUN TASK` statement. A sample archive Process, `DGAPSTRJ`, that submits a batch job using `DGADTSUB`.

Timing the Archive

The archive must complete before IBM Connect:Direct needs to reuse the file being archived, that is, at the time of wrap-around of the file pair list. The completion of the archiving Process is important because IBM Connect:Direct erases the contents of the statistics file when the system switches to that file. In other words, archiving must complete within the time required for the file pair list to wrap. Normally, this condition does not present a problem.

Require Confirmation of Archival

The `STAT.ARCH.CONFIRM` initialization parameter specifies whether or not to ensure that data is archived before the system erases the file. If you do not want archiving, IBM Connect:Direct simply resets the files when the switching occurs, and begins writing. If you want archiving, IBM Connect:Direct verifies that the archive is complete before proceeding. In this case, IBM Connect:Direct requires notification of archival. IBM Connect:Direct is notified in several ways:

- If the archive is done using the `COPY` statement in a IBM Connect:Direct Process, then the Process can also invoke the `DGADARRT` utility when the `COPY` successfully completes. IBM Connect:Direct invokes `DGADARRT` through a `RUN TASK` statement, and notifies IBM Connect:Direct that the data is archived.
- If you use a batch job to archive, then the job can send the notification by including a step to execute the `DGADARBT` utility.
- Also, you can issue the API command `STATISTICS ARCHIVED` to inform IBM Connect:Direct to reuse a file pair.

If no indication regarding the completion of the archive exists when IBM Connect:Direct needs to reuse the files, the system issues a message, similar to the example shown below, to the operator console and waits for a reply indicating permission to reuse the file.

```
10.00.01 JOB82592  SSTL013I Statistics file pair switch from 02 to 01, code:TIMER
10.00.02 JOB82592  SSTL009I Arch notification required but not received for file pair 01
10.00.02 JOB82592  *93 SSTL008I Reply "GO" when file available, or "DISABLE" logging.
```

In this situation, all activity in the DTF ceases until a response to the message from the operator exists indicating that the statistics file can now be overwritten. This safeguard occurs as a result of the request that the DTF not erase statistics data unless it is certain that archiving the statistics is complete.

If you have not been requiring archival notification and decide to begin requiring it, you can avoid getting these messages by using the IUI command `STAT, CF` option and forcing all pairs not in use to indicate they have been archived.

Not Requiring Confirmation of Archival

IBM Connect:Direct does not require archive confirmation before reusing a statistics file pair when you specify or default to the DTF initialization parameter, `STAT.ARCH.CONFIRM=NO`. If you specify this

initialization parameter, you are responsible for ensuring that the archive successfully completes before IBM Connect:Direct resets the file. If the file is reset before copying the records, the data is lost.

If the records are in the process of being copied when IBM Connect:Direct needs to reset the file, then IBM Connect:Direct must wait for the copy to complete. This operation is because IBM Connect:Direct must have exclusive access to the file to do the VSAM reset. In this situation, IBM Connect:Direct also issues a message to the operator console and waits for a reply indicating that the file can be reset.

Using the **SELECT STATISTICS** Command with Archived Statistics

Connect:Direct for z/OS provides a means of issuing the SELECT STATISTICS command against archived statistics. To make archived statistics available to SELECT STATISTICS, you must put the archived statistics in the format of a statistics file pair. You must make available a VSAM entry-sequenced cluster with a paired VSAM key-sequenced cluster containing the index information.

For example, if the records are archived to a magnetic tape file, you must first copy the archived records to a VSAM ESDS. Then you can run the DGADBKEY utility to build the required associated VSAM KSDS. Refer to *Program Directory for IBM Connect:Direct for z/OS* and *Planning DASD Requirements in IBM Connect:Direct for z/OS Configuration Guide* for information about the characteristics and relative sizes of the keyed and entry-sequenced clusters of a file pair. See [“DGADBKEY” on page 371](#) for an explanation and example of the DGADBKEY utility.

Use the ARCHDSN parameter of the SELECT STATISTICS command to search archive files. The ARCHDSN parameter names only the key-sequenced clusters of the archive pairs; IBM Connect:Direct locates the associated entry-sequenced clusters using control information in the key-sequenced clusters.

IBM Connect:Direct does not examine the statistics file pair list of the DTF when using the ARCHDSN parameter. IBM Connect:Direct only searches the archive files. SELECT STATISTICS processing does not let you name files currently in the file pair list of the DTF in the ARCHDSN parameter or combine archive files with files in the file pair list. Refer to the *IBM Connect:Direct for z/OS User Guide* for a description of the SELECT STATISTICS command and the ARCHDSN parameter.

Maintaining an Archive File Directory

IBM Connect:Direct also provides the capability of maintaining a directory of statistics archive files. The directory is a VSAM key-sequenced file that contains a record for each archive file. Information in the record includes the data set name of the archive file and the range of dates and times covered by the archived records. Refer to *VSAM Files DASD Requirement and Description* in the *IBM Connect:Direct for z/OS Configuration Guide* for estimating space requirements when allocating the directory file.

To use the directory feature, you must allocate the directory file and specify its name in the STAT.ARCH.DIR initialization parameter. IBM Connect:Direct provides a means of viewing the directory contents using the INQUIRE STATDIR command.

The archive notification utilities, DGADARRT and DGADARBT, write the directory records. If you want to use the directory feature, you must execute one of these utilities from the Process or batch job that archives the records. This condition is true even if you do not specify STAT.ARCH.CONFIRM=YES in the DTF initialization parameters. You must also use these utilities to send archive notification when you are not using the directory feature, but specify STAT.ARCH.CONFIRM=YES in the Initialization Parameters file.

Archive-Related Utilities

This section explains the archiving related utilities: DGADARRT, DGADARBT, and DGADBKEY.

DGADARRT

The DGADARRT utility (ARchive Run Task) has the following functions:

- Notifies IBM Connect:Direct of the availability of a statistics file pair for reuse due to the completion of archiving
- Optionally adds an entry to the directory of archive files

- By default, removes the oldest record from a full archive directory to make room for the newest record.

You can invoke DGADARRT from within a Process through the RUN TASK statement. Use this utility when submitting a Process at statistics file pair switch time that archives the statistical data using IBM Connect:Direct to copy the statistics to another file. When the copy operation successfully completes, the system can update the directory and send the archive notification.

The program accepts three parameters through the RUN TASK statement.

- The first parameter is required and is the data set name of the statistics entry-sequenced cluster that is archived.
- The second parameter is optional, and is the data set name of the archive file.
- The third parameter is optional, and specifies whether to age the archive directory.

DGADARRT always sends archive notification to the DTF. If you specify STAT.ARCH.CONFIRM=NO and no notification requirement exists, the notification has no effect.

The addition of the entry in the directory of archive files depends on the specification of the second parameter string. If the second parameter is present, then the system updates the directory to contain an entry for the new archive file.

The following is an example of an archive Process. This Process copies a statistics file to a sequential tape file and then invokes DGADARRT to send archive notification to the DTF and update the directory of archive files. IBM Connect:Direct passes the data set name of the statistics file to the Process in the form of the symbolic parameter &EDSN.

```

DGAPSTAT PROCESS PNODE=primary.node          -
                  SNODE=secondary.node        -
                  PRTY=10                      -
                  STARTT=(TODAY)               -
                  &EDSN=                       -
ARC              COPY FROM (DSN=&EDSN)         -
                  TO   (DSN=stat.archive.dsn(+1) -
                  DISP=(NEW,CATLG,DELETE)       -
                  DCB=(DSORG=PS,RECFM=VB,LRECL=2048) -
                  UNIT=CART -
                  LABEL=(1,SL) )
                  IF (ARC EQ 0) THEN
                      RUN TASK (PGM=DGADARRT,PARM=("&EDSN", -
                      "stat.archive.dsn(+0)"))
                  EIF

```

Whether the archive directory ages off the oldest record to make room for the newest depends on the third parameter, ARCAGE.

ARCAGE=Y - Requests that if the archive directory is full, the oldest record is deleted to make room for the newest. The user receives notification that the record is aged off the archive directory. The amount of time it takes to add a record to the archive directory is insignificant. But the amount of time it takes to age off the oldest record is noticeable and increases with the size of the archive directory, because all of the records in the ESDS must be rewritten. This is the default.

ARCAGE=N - Requests that if the archive directory is full, the utility stops without updating the archive directory.

The following example shows the ARCAGE parameter specified in the DGADARRT utility for use within a Process through the Run Task statement:

```

DGAPSTAT PROCESS PNODE=primary.node           -
                  SNODE=secondary.node         -
                  PRTY=10                      -
                  STARTT=(TODAY)               -
                  &EDSN=                       -
ARC      COPY    FROM (DSN=&EDSN)              -
                  TO   (DSN=stat.archive.dsn(+1) -
                  DISP=(NEW,CATLG,DELETE)       -
                  DCB=(DSORG=PS,RECFM=VB,LRECL=2048) -
                  UNIT=CART -
                  LABEL=(1,SL) )
                  IF (ARC EQ 0) THEN
RUN TASK (PGM=DGADARBT,PARM=("&EDSN", -
                  "stat.archive.dsn(+0)", -
                  "ARCAGE=Y"))
EIF

```

To prevent aging, set ARCAGE=N. As a result, the newest record is discarded, rather than the oldest.

DGADARBT

The DGADARBT utility (ARchive BaTch) has the following two functions:

- Notifies IBM Connect:Direct that a statistics file pair is archived and is available for reuse
- Optionally adds an entry to the directory of archive files
- By default, removes the oldest record from a full archive directory to make room for the newest record.

Execute DGADARBT as a step within a batch job. Use DGADARBT when submitting a job that archives the statistical data by executing IDCAMS or some other utility to COPY the data to another file at statistics file pair switch time. The system can update the directory and send archive notification upon successful completion of the copy operation.

DGADARBT requires that the system allocate the archived statistics file with the data definition name (DDNAME) of STESDS.

DGADARBT always sends archive notification to the DTF. If you specify STAT.ARCH.CONFIRM=NO and no notification requirement exists, then the notification has no effect.

If you want DGADARBT to update the directory of archive files, the system must allocate the following DDNAMEs:

- STADIR, the directory file
- STARCH, the archive file

In the following example, the archive Process submits a batch archive job using DGADTSUB to substitute the statistics file data set name into the job stream. The system passes this data set name to the archive Process as the symbolic parameter &EDSN. Refer to the *IBM Connect:Direct for z/OS User Guide* for information about how to use DGADTSUB. The submitted job uses the IDCAMS utility to copy the statistics records to an archive file. If the IDCAMS step is successful, IBM Connect:Direct invokes DGADARBT to both send the archive notification and update the directory.

The following figure is a sample archive Process.

```

ARCHSTRJ PROCESS PNODE=primary.node           -
                  SNODE=secondry.node         -
                  PRTY=10                      -
                  STARTT=(TODAY)               -
                  &EDSN=                       -
RUN TASK (PGM=DGADTSUB,                        -
          PARM=("DSN=SYS3.CONNECT.INSTALL.JCL(ARCHJOB),DISP=SHR", -
          "DSNAME &EDSN"))

```

The following figure is a sample archive job stream.

```

//ARCHJOB JOB (ACCT),CLASS=A
//*
//* ----- *
//* IDCAMS step to archive the Statistics ESDS: *
//* ----- *
//*
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=&DSNAME
//OUTPUT DD DISP=(NEW,CATLG,DELETE),
// DSN=stat.archive.dsn(+1),
// UNIT=CART,
// LABEL=(1,SL)
//SYSIN DD *
REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
//*
//* ----- *
//* DGADARBT step to notify DTF that ESDS was archived: *
//* ----- *
//*
//STEP2 EXEC PGM=DGADARBT,COND=(0,LT,STEP1)
//STEPLIB DD DSN=prod.ndmllib,DISP=SHR
//SYSOUT DD SYSOUT=*
//STESDS DD DISP=SHR,DSN=*.STEP1.INPUT
//STARCH DD DISP=SHR,DSN=*.STEP1.OUTPUT
//STADIR DD DSN=stat.archdir.dsn,DISP=SHR
//

```

The DGADARBT utility accepts the same ARCADE parameter as specified for DGADARRT. It is the first and only OS parameter for this utility.

The following example shows the ARCADE parameter specified in the DGADARRT utility to run as a step in a batch job:

```

//ARCHJOB JOB (ACCT),CLASS=A
//*/
//* ----- *
//* IDCAMS step to archive the Statistics ESDS: *
//* ----- *
//*
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=&DSNAME
//OUTPUT DD DISP=(NEW,CATLG,DELETE),
// DSN=stat.archive.dsn(+1),
// UNIT=CART,
// LABEL=(1,SL)
//SYSIN DD *
REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
//*
//* ----- *
//* DGADARBT step to notify DTF that ESDS was archived: *
//* ----- *
//*
//STEP2 EXEC PGM=DGADARBT,COND=(0,LT,STEP1),PARM='ARCADE=Y'
//STEPLIB DD DSN=prod.ndmllib,DISP=SHR
//SYSOUT DD SYSOUT=*
//STESDS DD DISP=SHR,DSN=*.STEP1.INPUT
//STARCH DD DISP=SHR,DSN=*.STEP1.OUTPUT
//STADIR DD DSN=stat.archdir.dsn,DISP=SHR
//

```

To prevent aging, set ARCADE=N. As a result, the newest record is discarded, rather than the oldest.

DGADBKEY

The DGADBKEY utility (Build KEYS) loads a statistics key-sequenced cluster with index information for an associated statistics entry-sequenced cluster. DGADBKEY must execute as a batch job step.

DGADBKEY enables the recreation of index information for archived statistics data so that you can issue a SELECT STATISTICS command. You can also use this utility to rebuild index information for statistics files in the DTF file pair list in certain cases. Refer to [Change the File Pair Configuration](#) for additional information.

DGADBKEY requires the allocation of DDNAMEs, ESDSnn and KSDSnn, with the entry-sequenced and key-sequenced clusters respectively. IBM Connect:Direct loads the entry-sequenced cluster with the statistics records for building the index information. The key-sequenced cluster must either be empty, or be defined with the REUSE attribute. DGADBKEY erases any records in the KSDS before writing the new information. The size of the KSDS is about 25% of the size of the associated ESDS. The KSDS must have the characteristics of a statistics key-sequenced cluster. Refer to *VSAM Files DASD Requirement and Description* in the *IBM Connect:Direct for z/OS Configuration Guide* for details about allocating statistics clusters.

The following is an example of a job stream to execute DGADBKEY.

```
//DGADBKEY JOB (ACCT),CLASS=A
//STEP1 EXEC PGM=DGADBKEY
//STEPLIB DD DISP=SHR,DSN=prod.SDGALINK
//ESDS01 DD DISP=SHR,DSN=stat.esds01
//KSDS01 DD DISP=SHR,DSN=stat.ksds01
//SYSOUT DD SYSOUT=*
//
```

PARM=CONSOLIDATE allows DGADBKEY to build a KSDS for an ESDS that contains multiple archived statistics datasets. To use this parameter, create a large flat file of archived statistics, loading them in date order from the oldest to the newest. Next, load this file to an ESDS that will hold all the records and create a KSDS for this ESDS. Then execute DGADBKEY using PARM=CONSOLIDATE to build the KSDS file. This file pair is available as an Archived Statistics Dataset.

Sample Archiving Setup

Assume that you have the following requirements for archiving:

- Statistics records must remain available for seven days in the file pair list before being overwritten by new records. After seven days, they must be available in archive files.
- Each archive file can contain no more than one day of statistics records.
- Batch jobs executing the IDCAMS utility to copy the records to sequential files on magnetic tape must perform the archiving. The archive files must be available for 365 days.
- Maintain a directory of archive files.
- Ensure that statistics data is not overwritten before being archived.
- Establish a procedure for making archived statistics available to the SELECT STATISTICS command.

Sample Statistics Configuration

This section describes how to configure the statistics facility to satisfy these requirements.

You must determine how to configure the statistics file pair list. The rate at which you log statistics records, availability of the statistics records for seven days before being overwritten, and each archive file containing no more than one day of statistics records determine the size and number of file pairs required.

For the statistics records to remain available for seven days after being generated, the total record capacity of all the entry-sequenced statistics files is seven times the average number of records generated daily. Run the DGASSTAT utility to determine how many records, on average, the system writes daily.

According to the sample requirements, a single archive file contains up to the same number of records as a single statistics entry-sequenced cluster, and an archive file contains no more than a day of records. Each ESDS also holds about a day of records, implying that seven statistics file pairs exist. Use the STAT.SWITCH.TIME initialization parameter to initiate a file pair switch every day at midnight rather than

depend on a file pair switch occurring regularly as a result of file pairs filling. To ensure that the switches do not occur before midnight as a result of a file pair becoming full, make each ESDS slightly larger than the daily requirement.

If you determine, using DGASSTAT, that the system writes statistics records at the rate of about 11,000 daily, define seven file pairs each with a capacity of about 12,000 records (RECORDS(12000)). This figure implies that the associated key-sequenced clusters are defined with RECORDS(9000).

RECORDS(365) defines the directory of archive files because the system generates the archive files at the rate of one daily and retains the files for one year.

SYSTEMS.CD.STATS is the data set name prefix for the statistics clusters. The data set name of the archive directory is SYSTEMS.CD.STATS.DIRECTRY. Member ARCHPROC in the data set SYSTEMS.CD.ADMINLIB contains the archive Process that is submitted at file pair switch time.

The following initialization parameters are necessary for the sample archive requirements.

```

STAT.DSNBASE = SYSTEMS.CD.STATS      /* data set name base      */
STAT.FILE.PAIRS = 7                  /* number of file pairs    */
STAT.SWITCH.TIME = ( 00:00 )         /* switch at midnight      */
STAT.SWITCH.SUBMIT = SYSTEMS.CD.ADMINLIB(ARCHPROC) /* archive proc */
STAT.ARCH.DIR = SYSTEMS.CD.STATS.DIRECTRY /* use directory */
STAT.ARCH.CONFIRM = YES               /* be sure archive completes */

```

The archive Process in the member ARCHPROC follows.

```

ARCHIVE PROCESS  &EDSN=,              - /* passed stats dsname    */
                  SNODE=CD.PROD,      - /* PNODE=SNODE            */
                  PNODE=CD.PROD,      - /*                          */
                  CLASS=1,             - /* lowest class           */
                  PRTY=15,             - /* highest priority       */
                  REQUEUE=YES          - /* re-queue on error      */
/*                          */
/* invoke DGADTSUB to submit the archive job... */
/*                          */
      RUN TASK (PGM=DGADTSUB,          - /* execute DGADTSUB,     */
PARM=("DSN=SYSTEMS.CD.JCL(ARCHJOB),DISP=SHR", - /*job */
/* stream to sub*/
      "DSNAME &EDSN"))                /* pass stat dsname      */

```

The archive job stream in SYSTEMS.CD.JCL(ARCHJOB) follows.

```

//ARCHJOB JOB (ACCT),ARCHIVE,CLASS=A,MSGCLASS=Z,MSGLEVEL=(1,1)
//***** archive the statistical data *****
//ARCHIVE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,CD=&DSNAME          /* from DGADTSUB */
//OUTPUT DD DSN=SYS.NDM.ARCH.STATS(+1),DISP=(NEW,CATLG,DELETE),
//        UNIT=TAPE,DCB=(BUFNO=6)
//SYSIN DD *
      REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
//***** notify Connect:Direct that the file pair can be reused, ****
//***** and update the directory of archive files *****
//NOTIFY EXEC PGM=DGADARBT,COND=(0,LT) /* if no errors */
//STEPLIB DD DISP=SHR,DSN=SYS.CD.SDGALINK
//STESDS DD DISP=SHR,DSN=*.ARCHIVE.INPUT /* stat file */
//STDIR DD DISP=SHR,DSN=SYSTEMS.CD.STATS.DIRECTRY /* archive directory */
//STARCH DD DISP=SHR,DSN=*.ARCHIVE.OUTPUT /* archive file */
//SYSOUT DD SYSOUT=*
//

```

The previous archive job stream indicates that the IBM Connect:Direct administrator manages requests for access to archived statistics records. The submitted requests specify a range of dates and times for the necessary records.

The administrator issues the INQUIRE STATDIR command to determine which archive files contain records for the specified period. The administrator runs the following job stream to create a usable archived statistics file pair for each archive file that it finds. The first step creates the archive file and copies the record to it. The second step builds the index information.

```
//RESTORE JOB (ACCT),RESTORE,CLASS=A,MSGCLASS=Z,MSGLEVEL=(1,1)
//ARCHIVE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=SYS.CD.ARCH.STATS.GnnnnVnn /* arch seq */
//SYSIN DD *
  DEFINE CLUSTER
    (NAME(SYS.CDARCH.Dyymmdd.KSDS) - /* define archive KSDS */
    VOLUMES(USRVOL) - /* supply archive date yymmdd */
    INDEXED NOIMBED
    FREESPACE(0 0)
    KEYS(27 0)
    RECORDSIZE(32 78)
    REUSE NOREPLICATE
    SHAREOPTIONS(2))
  DATA
    (CONTROLINTERVALSIZE(4096)
    RECORDS(9000)
    NAME(SYS.CDARCH.Dyymmdd.KSDS.DATA))
  INDEX
    (CONTROLINTERVALSIZE(512)
    NAME(SYS.CDARCH.Dyymmdd.KSDS.INDEX))
  DEFINE CLUSTER
    (NAME(SYS.CDARCH.Dyymmdd.ESDS) - /* define archive ESDS */
    VOLUMES(USRVOL)
    REUSE NONINDEXED NOIMBED
    RECORDS(12000) - /* same size as stats files */
    RECORDSIZE(275 2048)
    SHAREOPTIONS(2))
  DATA
    (CONTROLINTERVALSIZE(4096)
    NAME(SYS.CDARCH.Dyymmdd.ESDS.DATA))
  IF MAXCC = 0 - /* if clusters allocated OK */
  THEN REPRO INFILE(INPUT) - /* then load with stats */
    OUTDATASET(SYS.CDARCH.Dyymmdd.ESDS)
/*
//*****
//***** rebuild statistics index information
//*****
//BLDKEY EXEC PGM=DGADBKEY,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=SYS.CD.NDMLIB
//SYSOUT DD SYSOUT=*
//ESDSnn DD DISP=SHR,DSN=SYS.CDARCH.Dyymmdd.ESDS /* ESDS cluster */
//KSDSnn DD DISP=SHR,DSN=SYS.CDARCH.Dyymmdd /* KSDS cluster */
//
```

The archived statistics are now available and you can issue SELECT STATISTICS against the statistics by coding the name of the key-sequenced file with the ARCHDSN parameter, as follows.

```
SELECT STATISTICS WHERE -
(PNAME=USERPROC, ARCHDSN=(SYS.CDARCH.Dyymmdd))
```

Displaying the Status of the Statistics Logging Facility

The INQUIRE STATISTICS command displays the current status of the IBM Connect:Direct statistics logging facility.

The INQUIRE STATISTICS command has the following format.

Label	Command	Parameters
(optional)	INQUIRE STATISTICS	

No parameters are required for the INQUIRE STATISTICS command.

Statistics Inquiry through the Batch Interface

To use the INQUIRE STATISTICS command from the batch interface, perform the following steps.

Procedure

1. Place your command in a batch job stream as described in the *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running.

Note: You must set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

3. Verify your results.

Statistics Inquiry through the IUI Interface

To use the INQUIRE STATISTICS command from the IBM Connect:Direct IUI, perform the following steps.

Procedure

1. Access the statistics facility by selecting option INQ from the Connect:Direct Administrative Options Menu. The Inquire DTF Internal Status screen is displayed.
2. Type ISTA and press ENTER to display the status of the statistics logging facility.
3. Verify your results from the statistics logging facility display that is displayed. The report includes information such as the configuration of the statistics file pair list, the active file pair, file percentage utilizations, date and time ranges in the files, and additional information about the statistics facility.

Results

The following figure shows a partial sample report.

=====		
CD.ART	*INQ STATS*	DATE: 08.31.2018 TIME: 14:35:29
=====		
Status	=> Enabled	Sec. Name =>
Return Code	=> 0	Message ID => SSTL000I
Last "S2"	=> 00:00:00	Que Wait => No
Dsn Base	=> EPETE1.CD.ART.STATS	Q threshld => 375
Rst KSDS Vers	=> 2	Q low-watr => 1453
Excluded	=>	
***** F I L E P A I R #01 *****		
Status	=> Active	KSDS Vers => 2
Start Date	=> 08.31.2018	End Date => 08.31.2018
Start Time	=> 14:32:21	End Time => 14:35:29
KSDS Size	=> 18432000	ESDS CIS => 24576
ESDS Size	=> 73728000	ESDS Loc. => 26180419
Reset Pend.	=> No	Arch. Wait => No
Last Switch	=>	Sel. Count => 0
KSDS Status	=> Alloc, Open	ESDS Stat. => Alloc, Open
L-cmd	=> ENDREQ	L-cmd => ENDREQ
L-msg	=> SVS0000I	L-msg => SVS0000I
L-rc	=> 0	L-rc => 0
L-fdb	=> 0	L-fdb => 0
***** F I L E P A I R #02 *****		
Status	=> Empty	

Displaying the Statistics Archive File Directory

The INQUIRE STATDIR command displays the IBM Connect:Direct statistics archive file directory.

The INQUIRE STATDIR command has the following format and associated parameters.

Label	Command	Parameters
(optional)	INQUIRE STATDIR	STARTT = ([date day][,hh:mm:ssXM])

No parameters are required for the INQUIRE STATDIR command.

The following table describes the optional parameters used with the INQUIRE STATDIR command:

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	<p>This parameter specifies that the directory display is to begin with the first archive file created after the designated starting date and time. The date or day and time are positional parameters. If you do not specify the date or day, a comma must precede the time. If you omit this parameter, the display begins with the first directory entry.</p> <p>date</p> <p>This parameter specifies that the directory display is to start with this specific date. You can specify the day (dd), month (mm), and year (yy).</p> <p>You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). You can use periods or back slashes (/) to separate the components of a date value.</p> <p>You can omit the separators only for transfers between mainframe nodes. However, you must use separators for transfers between mainframes and all other platforms.</p> <p>After you designate the date order in your initialization parameters, you can use the following date formats:</p> <p>DATEFORM=MDY specifies the date format as:</p> <ul style="list-style-type: none">• mm/dd/yy or mm/dd/yyyy• mm.dd.yy or mm.dd.yyyy <p>DATEFORM=DMY specifies the date format as:</p> <ul style="list-style-type: none">• dd/mm/yy or dd/mm/yyyy• dd.mm.yy or dd.mm.yyyy <p>DATEFORM=YMD specifies the date format as:</p> <ul style="list-style-type: none">• yy/mm/dd or yyyy/mm/dd• yy.mm.dd or yyyy.mm.dd <p>DATEFORM=YDM specifies the date format as:</p> <ul style="list-style-type: none">• yy/dd/mm or yyyy/dd/mm• yy.dd.mm or yyyy.dd.mm <p>The following Julian date formats are valid:</p> <ul style="list-style-type: none">• yyddd or yyyyddd• yy/ddd or yyyy/ddd• yy.ddd or yyyy.ddd <p>If only date is specified, the time defaults to 00:00.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM]) (continued)	<p>day</p> <p>This parameter specifies to display the first archive file created after this day of the week. Valid names include MOnday, TUESday, WEdnesday, THursday, FRiday, SATurday, and SUNday. You can also specify YESTER to search for archive files created after yesterday or TODAY to search for the archive files created after today.</p> <p>hh:mm:ssXM</p> <p>Requests the first archive file created after this time of day, specified in hours (hh), minutes (mm), and seconds (ss). XM can be AM or PM. You can express the time of day using the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00–24:00. If you use the 12-hour clock, you can express 1:00 hours as 1:00AM, and you can express 13:00 hours as 1PM.</p> <p>If you do not use either AM or PM, IBM Connect:Direct assumes the 24-hour clock. You do not need to specify minutes and seconds. You can also specify NOON, which displays files created after noon, or MIDNIGHT, which displays archive files created after midnight. The default for the time is 00:00:00, the beginning of the day.</p> <p>If you specify time of day but not date, the output shows the first available entry in the archive directory for files created after that time of day. Archive files from all later times and dates display up to and including the stop time.</p>

Viewing the Statistics Archive Directory through the Batch Interface

To use the INQUIRE STATDIR command from the batch interface, perform the following steps.

Procedure

1. Place your command in a batch job stream as described in the *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running.

Note: Set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

3. Verify your results.

Viewing the Statistics Archive Directory through the IUI Interface

To issue the INQUIRE STATDIR command in the IBM Connect:Direct IUI, perform the following steps.

Procedure

1. Select option INQ from the Connect:Direct Administrative Options Menu to display the statistics facility. The Inquire DTF Internal Status screen is displayed.
2. Type IDIR and press ENTER to display the directory. A sample of the screen follows.

node.name	INQUIRE STATISTICS ARCHIVE DIRECTORY		hh.mm
CMD ==>			mm/dd/yy
			yyyy.ddd
START DATE	==> _____	(Gregorian or Julian)	
START TIME	==> _____	(HH:MM:SSXM)	

3. Supply the beginning date and time to limit the display for the INQUIRE STATDIR command. A report showing the results of the inquiry is displayed. The following figure shows a partial sample report.

```

=====
node.name          *INQUIRE STATDIR*  DATE:  mm/dd/yyyy  TIME:  hh:mm:ss
=====
Archival DSN:      USER01.STT.DGAPSTAT.G0008V00
Archival Notification: 03/02/1998  98.061  00:01:28
Oldest Record:     03/01/1998  98.060  00:00:06
Newest Record:     03/01/1998  98.060  23:59:54

Archival DSN:      USER01.STT.DGAPSTAT.G0009V00
Archival Notification: 03/03/1998  98.062  00:01:35
Oldest Record:     03/02/1998  98.061  00:00:11
Newest Record:     03/02/1998  98.061  23:59:45
.
.
.

```

Switching the Statistics File Pair

The STATISTICS SWITCH command initiates a statistics file pair switch. The currently active file pair closes, and logging continues on the next file pair in sequence. This command provides a means of initiating a file pair switch at any given time. Otherwise, switching occurs when the active file pair fills, or when a time of day specified in the STAT.SWITCH.TIME initialization parameter occurs.

The STATISTICS SWITCH command has the following format.

Label	Command	Parameters
(optional)	STATistics SWITCH	

No parameters are required for the STATISTICS SWITCH command.

Initiating a Statistics File Pair Switch through the Batch Interface

To use the STATISTICS SWITCH command from the batch interface, perform the following steps.

Procedure

1. Place your command in a batch job stream as described in the *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Initiating a Statistics File Pair Switch through the IUI Interface

The IUI provides a formatted panel that facilitates the issuing of the STATISTICS SWITCH command. To issue the STATISTICS SWITCH command through the IBM Connect:Direct IUI, perform the following steps.

Procedure

1. Select the STAT option of the Administrative Options Menu to access the Statistics Command panel.
2. Select option FS on the panel to initiate the file pair switch.

Recording Statistics for Specific Record Types

The STATISTICS ON/OFF command enables and disables recording of specific statistics record types. When you initialize the DTF, IBM Connect:Direct enables the recording of all record types unless you specify the STAT.EXCLUDE initialization parameter. You can use the INQUIRE STATISTICS command to find out which types are currently disabled.

Use the STATISTICS ON/OFF command prudently when excluding Statistics records logging because some types of records are critical for problem diagnosis. Do not exclude the following record types:

- CT – Copy Termination
- PS – Process Submit
- PT – Process Termination
- RJ – Run Job
- RT – Run Task
- SW – Submit within Process
- WO – WTO

Other record types are less critical and you can exclude them.



CAUTION: Excluding record types can make problem analysis and resolution more difficult.

The STATISTICS ON/OFF command has the following format and associated parameters.

Label	Command	Parameters
(optional)	STATistics ON OFF	TYPE = (record type list)

The following parameter is required for the STATISTICS ON/OFF command:

Parameter	Description
TYPE	This parameter specifies the list of statistics record types whose recording is enabled or disabled. Use the 2-character identifier to specify record types. These identifiers are in the table beginning on “Statistics Records” on page 318.

Excluding Statistics Logging through the Batch Interface

To use the STATISTICS ON/OFF command from the batch interface, perform the following steps.

Procedure

1. Place your command in a batch job stream as described in *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Refer to [“Recording Statistics for Specific Record Types”](#) on page 378 for more information on what not to exclude.

Excluding Statistics Logging through the IUI Interface

To use the STATISTICS ON/OFF command from the IUI, perform the following steps.

Procedure

1. Select the STAT option from the Connect:Direct Administrative Options Menu. The Statistics Command screen is displayed.
2. Select option EN to enable logging or option DI to disable logging. Supply the list of affected record identifiers in the area provided, and press ENTER.

Refer to [“Recording Statistics for Specific Record Types”](#) on page 378 for more information on what not to exclude.

Notifying IBM Connect:Direct of Statistics File Archival

The STATISTICS ARCHIVED command notifies IBM Connect:Direct that the statistics file is archived. It enables the system to erase and overwrite the file with new records.

When you specify STAT.ARCH.CONFIRM=YES in the DTF initialization parameters, IBM Connect:Direct cannot reuse a statistics file pair until it receives confirmation that the archive is complete. The STATISTICS ARCHIVED command provides an additional means of sending this notification. Ordinarily it is sent by the DGADARRT utility after the archive is done by a IBM Connect:Direct COPY Process, or by the DGADARBT utility after the archive is done by a batch step.

The STATISTICS ARCHIVED command has the following format and associated parameters.

Label	Command	Parameters
(optional)	STATistics ARCHived	file pair number

The following parameter is required for the STATISTICS ARCHIVED command:

Parameter	Description
file pair number	This parameter specifies a number from 1–20 that identifies the statistics file for which archive notification is sent. This number is given as the relative number of the file pair in the file pair list. The first pair in the list is file pair number 1.

Issuing Archive Notification through the Batch Interface

To use the STATISTICS ARCHIVED command from the batch interface, perform the following steps.

Procedure

1. Place your command in a batch job stream as described in *IBM Connect:Direct for z/OS User Guide*.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Issuing Archive Notification through the IUI Interface

To use the STATISTICS ARCHIVED command from the IUI, perform the following steps.

Procedure

1. Select the STAT option from the Connect:Direct Administrative Options Menu. The Statistics Command screen is displayed.
2. Select option CF, supply the number of the file pair for notification of archival, and press ENTER.

Managing the Transmission Control Queue

IBM Connect:Direct stores submitted Processes in the Transmission Control Queue (TCQ). The TCQ controls Process execution. The *IBM Connect:Direct for z/OS User Guide* contains information about how to submit Processes, how to control those Processes once they are in the TCQ, the logical queues that make up the TCQ, and the status values of Processes in the TCQ. The TCQ consists of two interdependent VSAM data sets:

- The Transmission Control Queue, or TCQ, is a Relative Record Dataset (RRDS) which contains an internal form of the Process language of each Process and status flags.
- The Transmission Control Index, or TCX, is an RRDS containing a single record. It contains bitmaps, that indicate the availability of TCQ space.

The default size of the TCQ, as determined by the sample installation JCL, is 1000 records, but it can be as large as 4016 records if the sample TCX is used. The size of a Process can range from 1 to 43 records,

depending upon the how many steps it contains. The average Process size varies by installation. If the average Process size is 5 records, the sample TCQ can contain approximately 200 Processes.

In order to use a TCQ with a capacity that exceeds 4016 records, the TCX must be defined with a record size and control interval (CI) size larger than those specified by default in the installation JCL. The *IBM Connect:Direct for z/OS Configuration Guide* and *Program Directory for IBM Connect:Direct for z/OS* contain more information about planning your space requirements.

Note: Both the TCQ and TCX can be defined with a CISIZE (Control Interval size) of up to 30,720 bytes. The maximum number of TCQ records that can be mapped by the maximum-sized TCX is 122,804.

To hold the maximum size Process (1 MB), the CISIZE of the TCQ must be at least 24 KB bytes.

Configure the TCQ

IBM Connect:Direct provides initialization parameters that allow you to configure the TCQ. These parameters fall into the following categories:

- Controlling startup—two parameters determine what the TCQ does with existing Processes:
 - TCQ = WARM | COLD, whose default value of WARM specifies that all existing Processes in the TCQ are retained. COLD requests that the TCQ be cleared of all processes.
 - CONFIRM.COLD.START = YES | NO, whose default value you must change to force the operator to confirm the request for a COLD start before executing it.
- Controlling efficiency of the TCQ—the following parameters provide several flexible configuration options in this area.
 - MAX.AGE lets you specify the number of days to wait before purging a Process. With this parameter you can also manage the Wait and Hold queues by specifying which type of Process to purge (that is, those with a specific status) or the number of days to wait to purge for each status type.
 - MAX.AGE.TOD used to change the system default of automatically purging the TCQ at midnight and whenever IBM Connect:Direct is initialized. Optional.
 - TCQ.THRESHOLD specifies when a warning is issued to indicate the TCQ is reaching capacity and Processes may be deleted. It is also the auto-deletion threshold for the PR queue.
- Holding Processes—two parameters determine if submitted Processes are held.
 - QUIESCE specifies whether or not IBM Connect:Direct holds Processes from execution.

Note: The QUIESCE parameter helps you in your efforts to clean up a TCQ, that has become corrupt. See [Using the TCQ/TCX Repair Utility \(DGADTQFX\)](#) for details.
 - REQUEUE specifies whether to requeue a Process, that ABENDS or results in a return code greater than 4.
- Retaining Processes after Execution—set the PROCESS.RETENTION parameter to save a Process in the PR queue after it has executed. Then view or select completed Processes.

How to Troubleshoot the TCQ

IBM Connect:Direct provides several ways to recover from a system malfunction associated with a TCQ problem.

The TCQ/TCX Repair Utility, DGADTQFX allows you to solve corruption problems related to the TCQ/TCX data sets without having to cold start the DTF and reinitialize the TCQ. The DGADTQFX batch program retains the original TCQ/TCX data sets used in production, and creates a new validated copy of the TCQ by removing all invalid Processes. The DGADTQFX utility can also be used to create a TCQ and TCX for pre-version 4.6 IBM Connect:Direct. Processes that are larger than 64 KB (the pre-version 4.6 Process limit) are removed from the new TCQ and TCX when the BACKLEVEL parameter is used.

Note: The DGADTQFX utility builds a new TCX/TCQ pair with increased CISIZES to hold larger Processes.

Using the TCQ/TCX Repair Utility (DGADTQFX)

The TCQ/TCX Repair Utility (DGADTQFX) can be used to help solve corruption problems related to the TCQ/TCX data sets.

Run the DGADTQFX utility in one of the following ways:

- Rebuild TCX mode, which creates a new TCX by using the current TCQ to indicate the existence of Processes.
- Use TCX mode, which creates a new TCQ using the current TCX to indicate the existence of Processes. When upgrading from Connect:Direct releases prior to 4.4, using TCX mode is recommended.

Note: Use different names to distinguish the original and new TCQ/TCX data sets in case you need to go back and reuse the original data sets.

The Rebuild TCX mode is the preferred mode for rebuilding the TCQ/TCX after encountering TCQ corruption problems, which cause U3083 abends.



CAUTION: The DGADTQFX utility is rebuilding the TCX based on Processes that remain in the TCQ. The utility will determine the highest Process number in the TCQ and will set the next available Process number to the next number. It is possible for this utility to reuse Process numbers.

The Use TCX mode is recommended for TCQs associated with systems running versions of IBM Connect:Direct prior to Version 4.4. Prior to Version 4.4, completed Processes were retained in the TCQ and were not deleted.

The DGADTQFX program, located in \$CD.SDGALINK, has one execution parameter for specifying the report type.

Parameter	Description
PARM= SUMMARY DETAIL BACKLEVEL REMOVEPR	Specify SUMMARY to produce a report at the Process level. Specify DETAIL to produce a report, which shows steps within each Process, such as RUN TASK, RUN JOB, SUBMIT, and COPY. Specify BACKLEVEL to create a TCQ and TCX for pre-version 4.6 IBM Connect:Direct systems. Specify REMOVEPR to remove all Processes on the PR queue.

Normally, you run DGADTQFX with IBM Connect:Direct shut down but you could run it in production. The data sets and reports created will be correct as long as no update activity to the input TCQ takes place while the utility executes. The program issues a warning message if the VSAM timestamp for the input TCQ is changed during execution.

The return codes associated with the DGADTQFX utility are described in the following table.

Return Code	Meaning
0	No errors were found in the input TCQ
4	At least one error was found and removed or a warning message was issued
8	A severe error occurred during execution and the utility was terminated

Initializing the DTF After DGADTQFX Has Found Errors

If errors were found and corrected when you ran the DGADTQFX utility, replace the original corrupted data sets in use with the new data sets created by DGADTQFX. To allocate the new data sets to IBM Connect:Direct, use IDCAMS ALTER or regenerate the network map.

Using IDCAMS ALTER

Procedure

1. Shut down the DTF, if necessary.
2. Rename the old TCQ and TCX to save the original data sets.
3. Using ALTER, rename the new TCQ and TCX data sets using the original data set names.
4. Initialize the DTF and specify the QUIESCE=YES initialization parameter.
5. Use SELECT PROCESS to display the TCQ contents, and delete any unwanted Processes.
6. Issue the MODIFY Sessions command to resume DTF operation.
7. After you are confident that IBM Connect:Direct is operating normally with the new TCQ and TCX data sets, delete the original TCQ and TCX datasets.

Regenerating the Netmap

Procedure

1. Shut down the DTF, if necessary.
2. Execute the Unload Netmap utility, DGADNTLD.
3. Change the names of the TCQ and TCX data sets in the unloaded member. The names are defined within the LOCAL.NODE definition.
4. REPRO the old network map data to preserve a copy of it for fallback purposes.
5. Delete, define, and reload the netmap.
6. Initialize the DTF and specify the QUIESCE=YES initialization parameter.
7. Use SELECT PROCESS to display the TCQ contents, and delete any unwanted processes.
8. Issue the MODIFY Sessions command to resume DTF operation.
9. After you are confident that IBM Connect:Direct is operating normally with the new TCQ and TCX data sets, delete the original network map, TCQ and TCX data sets.

DGADTQFX Examples

The following JCL samples are provided in \$CD.SDGASAMP:

- DGAXTQF1, which runs DGADTQFX in Rebuild TCX mode. DGADTQFX will create new TCQ and TCX data sets and print a detailed report.
- DGAXTQF2, which runs DGADTQFX in Use TCX mode. DGADTQFX will use the current TCX data set to indicate the existence of TCQ Processes. The program will print a summary report.
- DGAXTQF3, which runs DGADTQFX in BACKLEVEL mode. DGADTQFX will create new TCQ and TCX datasets and print a summary report. Processes larger than 64 KB are removed.

The following sample shows part of the JCL within \$CD.SDGASAMP library member DGAXTQF1.

```
//*****  
//*  
//*  Run the TCQ FIX utility in "Rebuild TCX" mode.  Create  *  
//*  new TCQ and TCX with any invalid contents of input  *  
//*  TCQ removed.  Request detail-level reporting for    *  
//*  input and output TCQ:                               *  
//*  
//*****  
//STEP2      EXEC PGM=DGADTQFX,PARM=DETAIL  
//STEPLIB    DD DISP=SHR,DSN=CD.LOADLIB  
//SYSOUT     DD SYSOUT=*  
//TCQIN      DD DISP=SHR,DSN=CD.OLD.TCQ  
//TCQOUT     DD DISP=SHR,DSN=CD.NEW.TCQ  
//TCXIN      DD DISP=SHR,DSN=CD.OLD.TCX (not needed in this mode)  
//TCXOUT     DD DISP=SHR,DSN=CD.NEW.TCX  
//TCQINRPT   DD SYSOUT=*
```

You can accomplish tasks depending upon which DD statements are present in the batch JCL:

- If TCXOUT is present, Rebuild TCX mode is requested; otherwise, Use TCX mode is requested.
- If TCQINRPT is present, an input TCQ report is generated; otherwise, it is not.
- If TCQOUT is present, a new TCQ is created; otherwise, it is not. You can therefore analyze and report on the input TCQ without creating any data sets.

Note: TCQOUT and TCXOUT must both be pre-allocated empty VSAM data sets.

DGADTQFX Output

The sample reports shown in this section are a result of running DGADTQFX in Use TCX mode. The first report shows two invalid TCQ Processes, which were detected and skipped during the copy of the input TCQ to the output TCQ. The second report lists all Processes in the input TCQ in sequential Process number order.

DGADTQFX Output TCQ Report (Summary)

```
Connect:Direct for z/OS
DGADTQFX execution on 18 Mar 2003 14:58:16
Mode: Use TCX
Output TCQ Summary
PName      PNum    Cur Step Submitter Node    Other Node    Stat  UserID    Submitted
-----
D3103UPR      4 STEP0108 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:08
D3104UPR      5 STEP0103 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:14
D3105UPR      6 STEP0103 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:17
D3106UPR      7 STEP01    PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:21
D3107UPR      8 STEP01    PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:24
Processes Skipped Summary
PName      PNum    Cur Step Submitter Node    Other Node    Stat  UserID    Submitted
-----
D3101UPR      2          PLEX.JOE      PLEX.TOM.TCP    HO HI CBENN1    19 DEC 2002  11:08:45
D3102UPR      3          PLEX.JOE      PLEX.TOM.TCP    HO HI CBENN1    19 DEC 2002  11:08:59
Totals:
Processes found in Input TCQ:          7
Processes written to Output TCQ:       5
Processes not copied to Output TCQ (skipped): 2
STQF001E  DGADTQFX ended; RC=04
```

DGADTQFX Input TCQ Report (Summary)

```
Input TCQ Summary
PName      PNum    Cur Step Submitter Node    Other Node    Stat  UserID    Submitted
-----
D3101UPR      2          PLEX.JOE      PLEX.TOM.TCP    HO HI CBENN1    19 DEC 2002  11:08:45
D3102UPR      3          PLEX.JOE      PLEX.TOM.TCP    HO HI CBENN1    19 DEC 2002  11:08:59
D3103UPR      4 STEP0108 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:08
D3104UPR      5 STEP0103 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:14
D3105UPR      6 STEP0103 PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:17
D3106UPR      7 STEP01    PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:21
D3107UPR      8 STEP01    PLEX.JOE      PLEX.TOM.TCP    EX EX CBENN1    19 DEC 2002  11:09:24
```

Supporting DBCS and MBCS

Overview of DBCS

Some languages have too many symbols for all characters to be represented using single byte codes. For example, the English language can be defined within a single byte range from 1-256, or x'00' through x'FF'. The Korean and other ideographic languages contain several thousand characters. To create these coded character sets, two bytes are needed for each character.

The IBM Connect:Direct Double-byte Character Set (DBCS) support provides a mechanism for translating ASCII and EBCDIC DBCS data. DBCS support translates Single-byte Character Set (SBCS) and DBCS data in the form that is supported on the requested platform.

DBCS character representation differs between operating systems. Specifically, a mainframe represents data in 8-bit EBCDIC code and a PC represents data in 7-bit ASCII code. For the mainframe environment, DBCS can be used exclusively within a file or be mixed with SBCS characters. Special character indicators exist to tell the difference between SBCS and DBCS characters. The special character indicators are shift-out (SO) and shift-in (SI), or x'0E' and x'0F' respectively for IBM mainframes. Shift-out denotes shifting from SBCS to DBCS mode and shift-in denotes shifting from DBCS to SBCS mode. SO/SI combinations are not required if DBCS is exclusive within a file. For the PC, the SO/SI characters are not recognized. In this environment, DBCS is represented by setting the high order bit of the ASCII code. See the table in [RULES](#) for correct mapping of DBCS characters by language.

Note: A DBCS table can be extremely large and complex. Use the sample tables in this documentation as a reference only. They do not successfully translate all characters.

Translation Tables

IBM Connect:Direct provides the following translation tables in both load module and source form. The executable tables are located in \$CD.SDGALINK and the source tables are in \$CD.SDGASAMP. You can copy and customize the source code format for your processing environment.

For more information on how to use these tables with the SYSOPTS parameter in the COPY statement, see the [Connect:Direct Process Language](#) help.

Table Name	Description
DGATXKSC	host EBCDIC to ASCII KS5601
DGATKSCX	ASCII KS5601 to host EBCDIC
DGATXKPC	host EBCDIC to DBCS-PC Korean
DGATKPCX	DBCS-PC Korean to host EBCDIC
DGATXJIS	host EBCDIC to Japanese International Standard
DGATJISX	Japanese International Standard to host EBCDIC
DGATXBG5	Chinese new host code to Chinese Big5
DGATBG5X	Chinese Big5 to Chinese new host code
DGATXC55	Chinese new host code to Chinese 5550
DGATC55X	Chinese 5550 to Chinese new host code
DGATJEFX	Japanese host EBCDIC Katakana to ASCII
DGATXJEF	ASCII to Japanese host EBCDIC Katakana
DGATGBKX	GBK to Chinese new host code
DGATXGBK	Chinese new host code to GBK

Customize Translation Tables

You can create and update the translation tables through a preprocessor that takes simple batch input in a predefined format and creates output compatible with the assembler. You can then assemble and link-edit the output to produce a translation table you can load.

Input to the batch preprocessor consists of six main parameters and the END parameter. All input begins in column one. The following table defines the batch preprocessor parameters.

Parameter	Required	Default	Format	Definition
NAME	No	XLATE	8 characters	Table name information
TITLE	No	DBCS TRANSLATION TABLE	60 characters	Table title information
DEFAULT	No	0000	2 byte hex representation	Default translation character
RULES	No	80-FF	2 byte hex representation	Language rules
SBCS	No	Standard	2 byte hex representation	Single-byte character set translation table
DBCS	Yes	None	4 byte hex representation	Double-byte character set translation table
END	Yes	None		Terminates DBCS, SBCS, and RULES parameters

DBCS

creates the double-byte character set translation table. This table translates all double-byte data during a file transfer. This parameter has no default and is required. The DBCS parameter data begins in column one and is terminated with the END statement.

The following example shows the syntax for the DBCS parameter.

```
DBCS
f1f2,t1t2
END
```

f1 denotes the first byte of the FROM DBCS character.

f2 is the second byte of the FROM DBCS character.

t1 is the first byte of the TO DBCS character.

t2 is the second byte of the TO DBCS character.

DBCS Example

The following example translates x'89A1' to x'B0ED', x'89A2' to x'B0EE', x'89A5' to x'B0EF', and so on to x'D37B' to x'C8F0'.

```
DBCS
89A1,B0ED
89A2,B0EE
89A5,B0EF
89A8,B0F0
89A9,B0F1
89AA,B0F2
89AB,B0F3
D375,C8EE
D377,C8EF
D37B,C8F0
END
```

END

is mandatory to terminate each of the following parameters:

- DBCS
- RULES
- SBCS

NAME=[tablename | XLATE]

is an 8-character parameter for displaying table information in batch format. NAME is optional and is for informational use only. If you use NAME, it must be the first parameter defined. If you use NAME with TITLE, NAME and TITLE must be the first two parameters defined. The default for NAME is XLATE.

The following example shows the syntax for the NAME parameter.

```
NAME=DGATXKSC
```

The DGATXKSC table is provided in \$CD.SDGASAMP.

TITLE=[title name | DBCS TRANSLATION TABLE]

is a 60-character parameter for displaying table information in batch format. TITLE is optional and is for informational use only. If you use TITLE, it must be the first parameter defined. If you use TITLE with NAME, NAME and TITLE must be the first two parameters defined. The default for TITLE is DBCS TRANSLATION TABLE.

The following example shows the syntax for the TITLE parameter.

```
TITLE=HOST EBCDIC TO ASCII KS5601 TRANSLATION
```

The DGATXKSC table is provided in \$CD.SDGASAMP.

DEFAULT=nnnn

contains the hexadecimal representation you define as the replacement for invalid DBCS code points. This default character is displayed wherever a nontranslatable character is displayed in the data being received. The default is 0000.

nnnn denotes the hexadecimal character defined to replace an invalid DBCS code point.

The following example shows the syntax for the DEFAULT parameter.

```
DEFAULT=FFFF
```

RULES

defines what constitutes a double-byte character for the defined language. RULES is only used when receiving a file from a platform other than z/OS or MVS, because the host cannot determine valid DBCS characters without language rules. The default is any character within the range of x'80' through x'FF', meaning IBM Connect:Direct interprets any character within this range as the first byte of a DBCS pair. Both characters in the pair are translated to host DBCS. If specified, use the END statement to terminate the RULES parameter.

Language Options Table

The following table identifies valid language options for the RULES parameter.

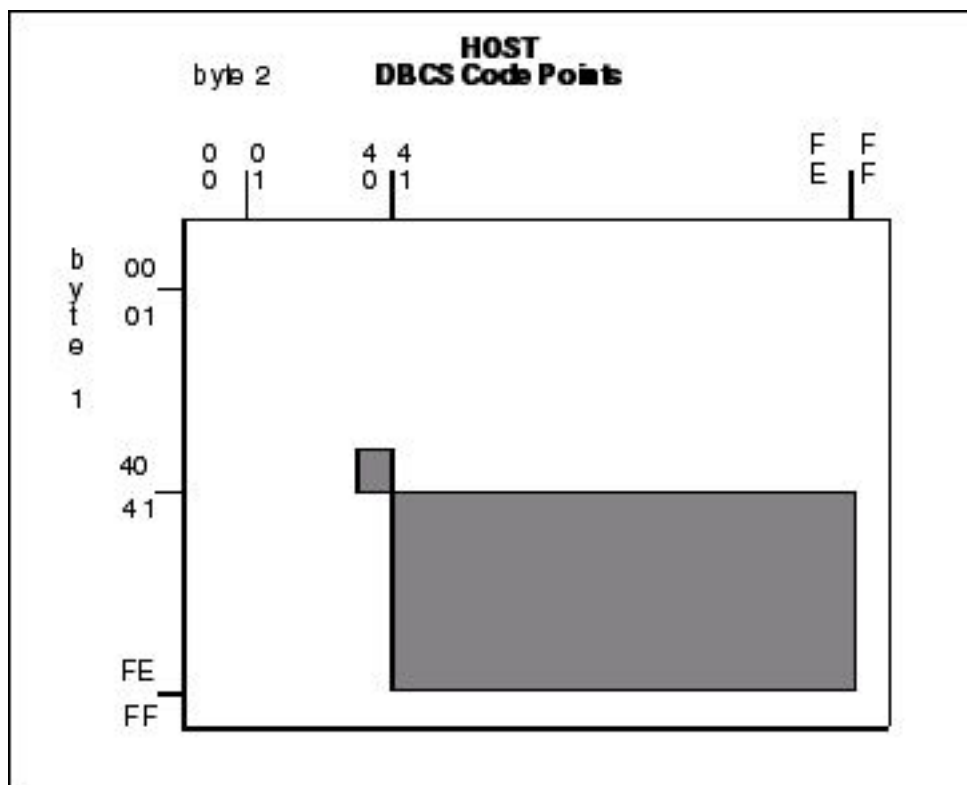
Language Option	Range
KS5601 (Korean Standard)	x'A1'-x'AC' x'B0'-x'FD'
KOREAN (Old Style)	x'81'-x'BF'
JAPANESE	x'81'-x'9F' x'E0'-x'FC'
CHINESE (Traditional/Simplified and 5550)	x'81'-x'FC'
BIG5 (Chinese)	x'A4'-x'C6' x'C9'-x'F9'
x'01'-x'FF'	user selectable

The following example shows the syntax for the RULES parameter.

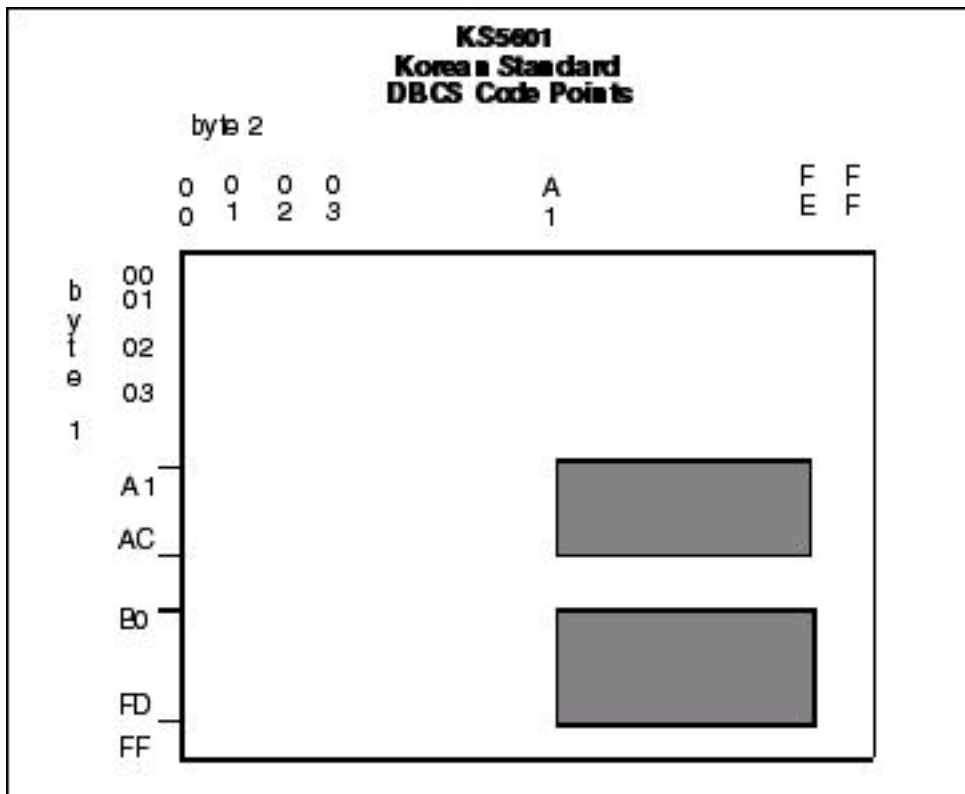
```
RULES
KS5601
END
```

The KS5601 language option is in the DGATXKSC table, which is provided in \$CD.SDGASAMP.

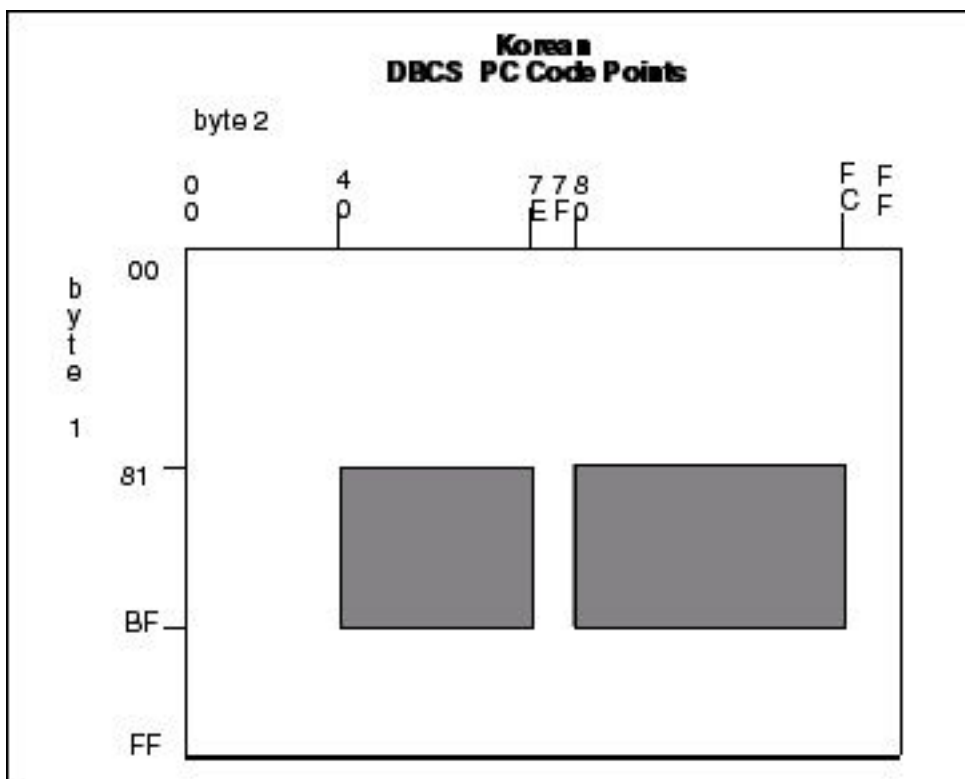
The following graphic represents the IBM Connect:Direct hexadecimal DBCS code points.



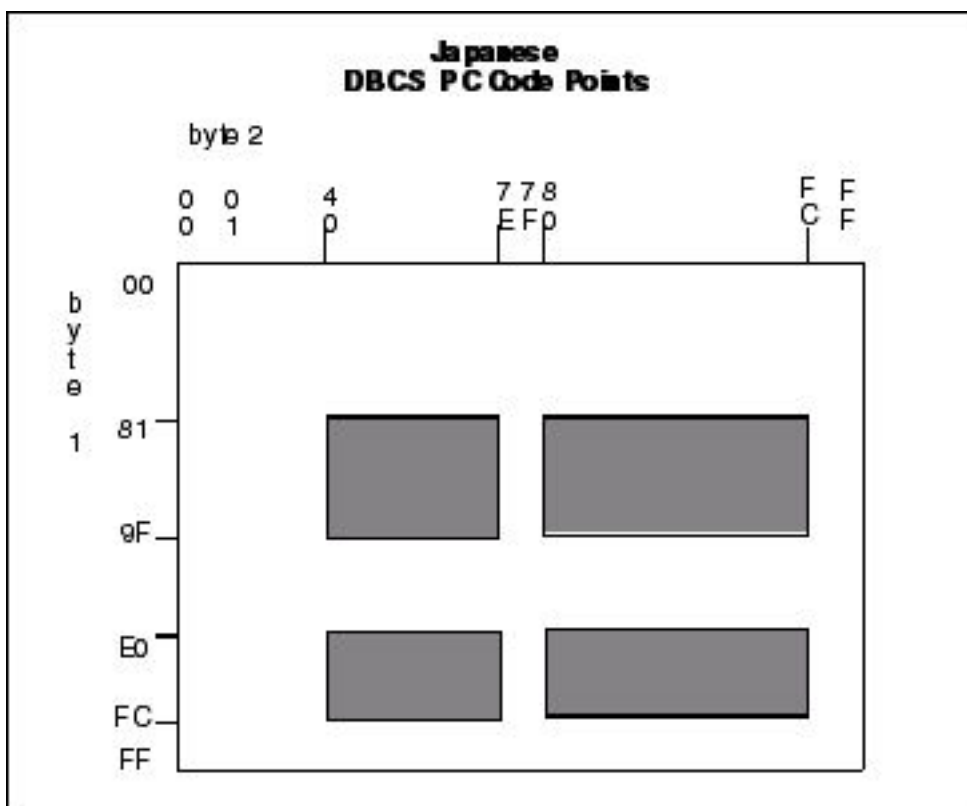
The following graphic represents the Korean Standard (KS5601) hexadecimal DBCS code points. The first character of each code point coincides with the range values in the [Language Options Table](#).



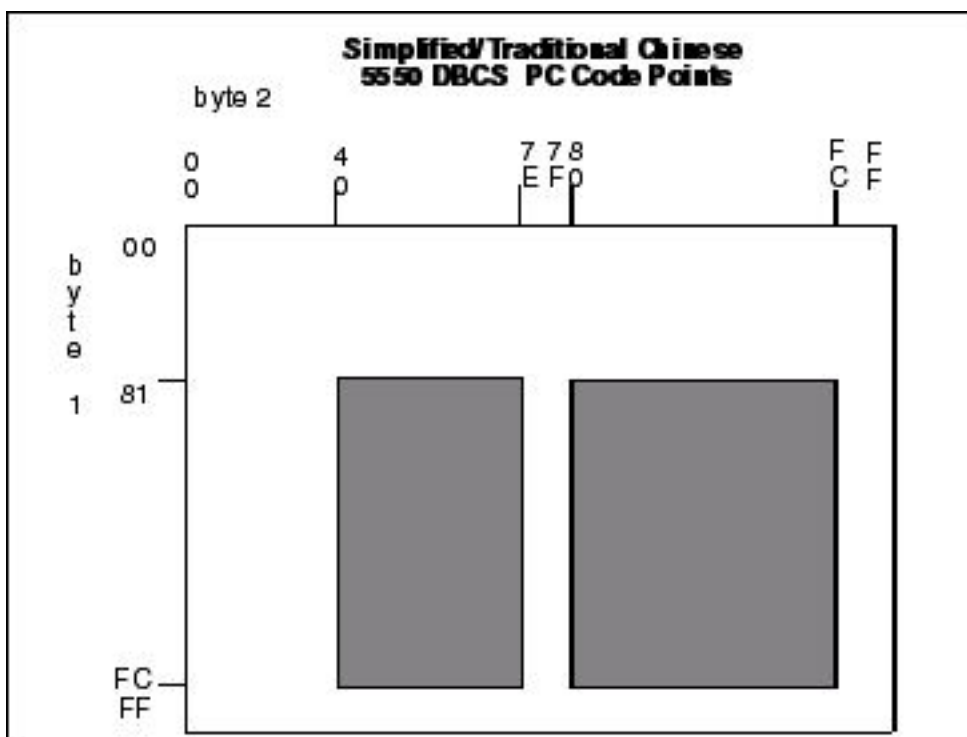
The following graphic represents the Korean hexadecimal DBCS code points. The first character of each code point coincides with the range values in the [Language Options Table](#).



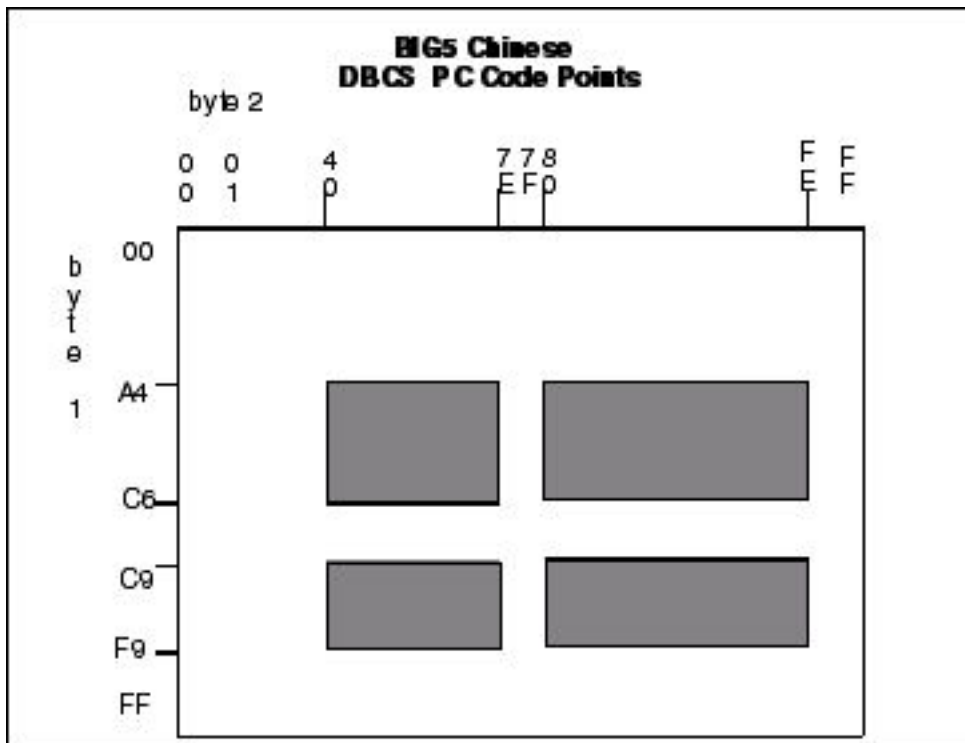
The following figure is a graphic representation of the Japanese hexadecimal DBCS code points. The first character of each code point coincides with the range values in the [Language Options Table](#).



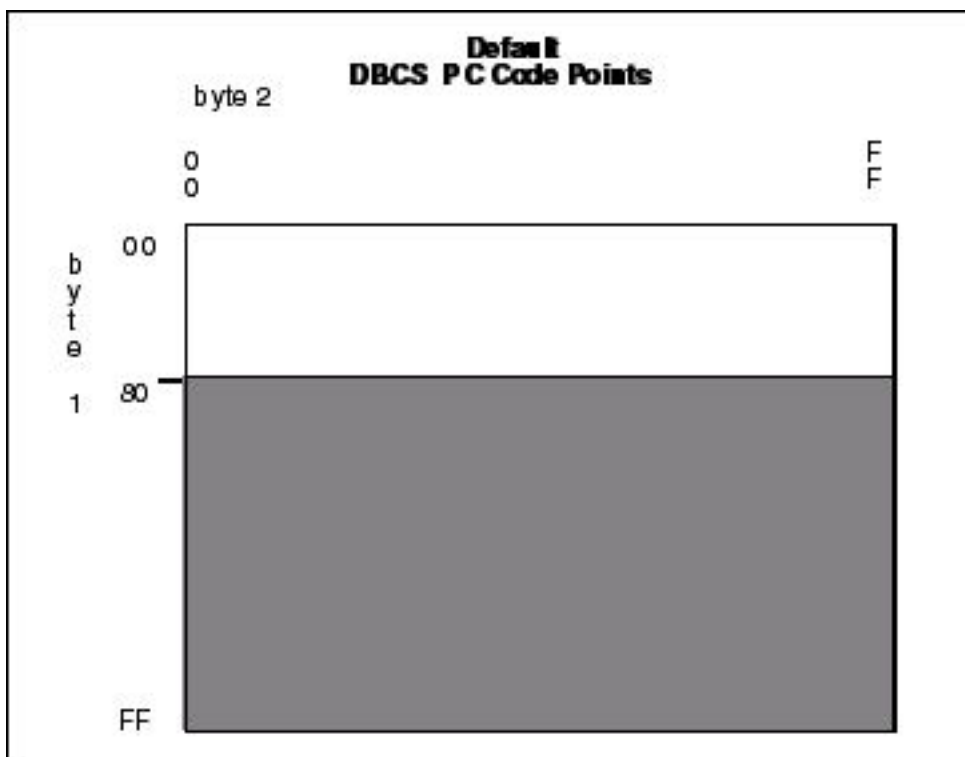
The following graphic represents the Traditional Chinese hexadecimal DBCS code points. The first character of each code point coincides with the range values in the [Language Options Table](#).



The following graphic represents the Chinese (BIG5) hexadecimal DBCS code points. The first character of each code point coincides with the range values in the [Language Options Table](#).



The following graphic represents the default hexadecimal DBCS code points.



RULES Examples

The following example translates all characters as DBCS that adhere to the KS5601 standard, or all characters that start with an x'A1' through x'AC' or x'B0' through x'FD'. Treat these characters as double-byte characters.

```
RULES
KS5601
END
```

The following example translates all characters as DBCS that adhere to the customized table. Treat all characters that start with x'90' through x'94' or x'B0' through x'B4' as double-byte characters.

```
RULES
90
91
92
93
94
B0
B1
B2
B3
B4
END
```

SBCS

creates the single-byte character set translation table. This table translates all single-byte data during a file transfer. The default translation table provided when the parameter is not specified, translates all EBCDIC characters in the range of x'00' through x'FF' to its ASCII equivalent, within the range of x'00' through x'7F'. When receiving the file from a PC, the data is translated from ASCII to EBCDIC. Terminate the SBCS parameter with the END statement.

If you define SBCS, you must begin all data in column one and only one hexadecimal character pair is allowed per line.

The following example shows the syntax for the SBCS parameter.

```
SBCS
ff,tt
END
```

ff denotes the FROM translation.

tt denotes the TO translation.

SBCS Example

The following example translates x'C1' to x'41', x'C2' to x'42', x'C3' to x'43', and so on.

```
SBCS
C1,41
C2,42
C3,43
C4,44
C5,45
C6,46
END
```

Comments

Comments allow you to include additional information in a batch preprocessor. Comments are available as a convenience and do not affect IBM Connect:Direct. The format for a comment is an asterisk (*) in column 1, followed by the comment. The following figure is a sample comment with an asterisk in column 1.

* DEFAULT=FFFF instead of 0000.

Sample Preprocessor Input Data Stream

The following sample is the syntax for a preprocessor input data stream. The SBCS and DBCS tables are incomplete and would require many pages to produce a valid table.

```
NAME=MYTABLE
TITLE=SAMPLE TRANSLATION TABLE
RULES
80
81
82
83
84
END
SBCS
C1,41
C2,42
C3,43
C4,44
C5,45
C6,46
END
DBCS
89A1,B0ED
89A2,B0EE
89A5,B0EF
89A8,B0F0
89A9,B0F1
89AA,B0F2
89AB,B0F3
D375,C8EE
D377,C8EF
D37B,C8F0
END
```

Sample JCL to EXECUTE the Preprocessor

The following sample JCL executes the preprocessor against the input source. The output produced by the preprocessor is in assembler CSECT form and is input to the assembler. The assembled object is then link-edited to produce a load module.

The following JCL is contained in the member DGAJDBCS in the \$CD.SDGAJCL library.

```

//JOBNAME JOB (ACCTN), 'ADMINISTRATOR', CLASS=A,
// REGION=4098K, MSGLEVEL=(1,1), MSGCLASS=X
//*****
//*
//* JCL TO CREATE DBCS TRANSLATE TABLE
//*
//* REPLACE THE FOLLOWING ENTRIES IN THE PROCEDURE STATEMENT
//* BELOW WITH SITE DEPENDENT INFORMATION
//*
//* TABLE= NAME OF THE SOURCE TRANSLATE TABLE
//*
//*****
//BLDDBCS PROC TABLE=XXXXXXXX, CDPREF='$CD', TEST=NOTEST, RENT=RENT
//*****
//* STEP1 CREATE ASSEMBLER OUTPUT FROM PRE-PROCESSOR INPUT
//*****
//STEP1 EXEC PGM=DMDBCSPR
//STEPLIB DD DSN=&CDPREF..SDGALINK, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//CDTABIN DD DSN=&CDPREF..SDGASAMP(&TABLE), DISP=SHR
//CDTABOUT DD DSN=&&SRC, DISP=(,PASS), SPACE=(CYL,(1,1)),
// DCB=(BLKSIZE=1600, LRECL=80, RECFM=FB, DSORG=PS),
// UNIT=SYSDA
//*****
//* STEP2 ASSEMBLE OUTPUT CREATE BY PRE-PROCESSOR
//*****
//STEP2 EXEC PGM=ASMA90,
// PARM=' OBJECT, NODECK, XREF (SHORT), &RENT, &TEST, USING (WARN (0X
// ), NOMAP), FLAG (NOCONT)'
//SYSLIB DD DSN=&CDPREF..SDGASAMP, DISP=SHR
// DD DSN=SYS1.MACLIB, DISP=SHR
// DD DSN=SYS1.MODGEN, DISP=SHR
//SYSTEM DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL,(10,5))
//SYSUT2 DD UNIT=SYSDA, SPACE=(CYL,(10,5))
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL,(10,5))
//SYSLIN DD DSN=&&OBJ, UNIT=SYSDA, DISP=(,PASS), SPACE=(TRK,(5,5)),
// DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSIN DD DSN=&&SRC, DISP=(OLD,DELETE)
//*****
//*

```

Information on IBM Connect:Direct Processes, including how to use the table created from the previous JCL sample using the COPY statement, is available at the [Connect:Direct Process Language](#) help.

Alternate Logic Applied to DBCS and SBCS Translation (LOGIC=B | (B,RC))

An alternate form of translation for data that contains both double byte and single byte data is provided. It supports the translation of mixed mode files using selectable translation tables.

The normal way that data is described, when a file contains both double and single bytes characters, is through the use of the shift-in (SI) and shift-out (SO) sequences. By default shift-out is normally 0x0E and shift-in 0x0F, but these values can be configured. The support within the Connect:Direct products allows for the shift-out and shift-in values to be specified if they are different than 0x0E and 0x0F. Record boundaries are honored. SI or SO strings do not span record boundaries.

For more information on how to implement this alternate form of translation using the LOGIC=B|(B,RC) subparameter in the SYSOPTS parameter in the COPY statement, see the [Connect:Direct Process Language](#) help.

It is expected that data which contains DBCS and SBCS strings will use alternating SO, SI sequences to define the start and end of each string. So a typical file record might look like:

SI	single byte string	SO	double byte string	SI	single byte string
----	--------------------	----	--------------------	----	--------------------

Any exception to this, such as two SI sequences without an intervening SO sequence, is considered invalid. When an invalid sequence is encountered, certain data strings are ignored and all translation is stopped, depending on the sequence of SI and SO markers.

Alternate Translation allows for the file to be completely transferred and an error (message SCPA074I with return code 8) noted at the completion of the transmission. You also have the option of using LOGIC=B|(B,RC) to specify a different return code within the range of 1-254 for this specific message. If you specify LOGIC=B without specifying a return code, the default return code of 8 is used. Connect:Direct will use the specified return code when issuing the SCPA074I message. The following coding examples show how to specify a return code for SCPA074I and the expected results for each:

```
SYSOPTS="DBCS=(EBCXKSC,0E,0F,LOGIC=(B,4))"
```

In the example above, if the DBCS transfer produces the SCPA074I message, the return code generated will be 4.

```
SYSOPTS="DBCS=(EBCXKSC,0E,0F,LOGIC=(B,254))"
```

In the example above, if the DBCS transfer produces the SCPA074I message, the return code generated will be 254 or X'FE'.

See the [Connect:Direct Process Language](#) help for more information about specifying a return code for the SCPA074I message.

Rules

These rules take into account any combination of SI and SO sequences, even ones with no intervening byte string. Since these rules are dependent on future sequences of SO or SI characters within the file.

The rules follow this truth table:

Note: SI still means Shift In to single byte mode and SO means Shift Out to double byte mode. However, no translation will be done if the SI, SO sequences are invalid.

PRIOR	NEXT	
	SI	SO
SI	Do not translate the remainder of the record	SBCS translation until SO occurs
SO	DBCS translation until SI occurs	Do not translate the remainder of the record

For PAD specification, if EBCDIC to ASCII translation or ASCII to EBCDIC translation, PAD character will replace SI and SO characters for the entire file unless an invalid shift sequence is found. When an invalid shift sequence is found in a record, the shift characters are left as is and the PAD character is not substituted from that point to the end of the record.

1. Each record is evaluated independently. All translation decisions are made on each individual record. SI/SO byte strings are not evaluated across record boundaries.
2. A record may or may not start with a shift character. If a shift character is not found in the first byte of the record, the byte string is considered SBCS data until the next shift character in that record is discovered.
3. If an invalid shift sequence is discovered, stop translation and send the record in a mixed format with the first part translated and the last part in the original, untranslated form.
4. If a PAD character is specified, use it for those portions of the record that are translated. Untranslated portions of the record will be left with any SI or SO characters as is.

5. End of record that occurs without a SI or SO sequence means data that precedes the end of file is treated as normal DBCS or SBCS data for translation depending on the previous shift state.

MBCS Conversions

Multibyte Character Set (MBCS) support enables you to convert between Unicode and other code sets supported on the z/OS platform. To perform an MBCS conversion, use the CODEPAGE parameter of the COPY statement FROM and/or TO SYSOPTS clauses.

You can perform MBCS conversions in the following ways:

- Perform a conversion on the FROM node only and then send the Unicode file to the TO node.
- Send a file to the TO node and let that node perform the conversion.
- Perform a conversion from one z/OS compatible code set to a Unicode code set supported on the local node (specified in the FROM clause CODEPAGE parameter). Then send the encoded Unicode file to the remote node to be converted to another z/OS compatible code set.

Instead of requiring that each IBM Connect:Direct node provide the capability to convert from any supported character set to any other supported character set, the recommended approach is to convert the original character set to a common intermediate form (UTF-8 or UCS-2) on the local node, transmit the intermediate form to the remote node, and then perform the conversion to the final desired character set on the remote node. This way, each node is responsible only for conversion between the Unicode encoding and the character sets relevant to and supported by the node.

Note: To convert between Unicode (ISO 10646) and other code sets, IBM Connect:Direct makes calls to system routines which are part of the optional z/OS Language Environment component - National Language Support. Verify that your z/OS installation supports the code set conversions specified in the Process language.

When you output an MBCS file, it should allow for a flexible output record length, thus z/OS files should be specified as Variable Format (RECFM=VB). Also, to allow for a possible increase in data length due to conversion, the LRECL of the receiving file must be larger than the LRECL of the sending file. In the example shown below, the LRECL of the sending file is 80.

```
STEP01  COPY FROM (PNODE DISP=SHR
          DSN=TEST3.MBCS0001.IBM930
          SYSOPTS="CODEPAGE=(IBM-930,UTF-8)" )
          TO (SNODE DISP=(,CATLG)
          UNIT=SYSDA SPACE=(CYL,(3,3))
          VOL=SER=USER01
          DCB=(RECFM=VB,LRECL=90,BLKSIZE=24000)
          DSN=CHICAGO.MBCS0001.IBM1047
          SYSOPTS="CODEPAGE=(UTF-8,IBM-1047)" )
```

For this particular MBCS conversion, the receiving file was successfully created by specifying LRECL as 90. Other conversions may require a larger value to avoid an SVSJ032I error during the Copy. If RECFM=VB, BLKSIZE for the output file must be at least as large as LRECL+4.

To display the CODEPAGE specification for a COPY step in a Process after step completion, use the Select Statistics command for an SY Statistics record. Each node involved in a COPY generates an SY record containing the SYSOPTS relevant to that node.

Except for syntax, the CODEPAGE parameter is not validated when the Process is submitted. However, when the Process is executed, an MBCS001E error will result on the node attempting the conversion if an invalid code set is specified.

For additional Process examples, search on MBCS Conversion in the [Connect:Direct Process Language help](#).

Analyzing TCP/IP Performance

Performance problems occur when a system and its network are not operating as effectively as they should as indicated by slow response times and a decrease in users' productivity. These problems can be intermittent or can indicate a growing strain pointing to capacity issues. Causes can be multifaceted and include both hardware and software origins or can be quickly solved with proper configuration settings.

Because of the multi-faceted nature of TCP/IP issues, this section serves as an analysis tool by providing a checklist of possible factors. By properly analyzing and defining the problem in terms of a set of symptoms and potential causes, you can either solve the performance problem yourself or can provide support with documentation.

When you analyze performance, you must also consider your organization's priorities, such as the following goals:

- To maximize throughput in order to achieve the maximum data transfer rate
- To "fill the pipe" and run at full capacity

As you test and fine-tune settings to address specific factors in a problem area, record the results and note any unusual interactions or behavior. You may want to change company standards and create a checklist to accommodate new procedures or settings.

General TCP/IP Problems

There are many causes of general TCP/IP problems.

This table lists TCP/IP problems and factors to consider:

Problem	Factors to Consider
Host Issues	<ul style="list-style-type: none">• Inadequate memory• Slow disk speed/contention• Slow channel speed/contention• Excessive workload• Inadequate processors/slow processor speed• Inefficient performance groups and dispatch priorities• Resource competition among applications on same system• Resource competition among LPARs

Problem	Factors to Consider
Network issues	<p>At the link level, look for:</p> <ul style="list-style-type: none"> • Link errors • Hardware or interface errors • Latency problems <p>Note: UDT is an alternative transport layer protocol designed for high latency with high-bandwidth connections. If you transfer files with a trading partner who is geographically distant and determine that the data congestion is caused by a high-latency, high-bandwidth connection, analyze your system and see if you would benefit from UDT. For additional information, see the <i>IBM Connect:Direct for z/OS Release Notes</i>.</p> <ul style="list-style-type: none"> • Collisions <p>At the IP layer, look for:</p> <ul style="list-style-type: none"> • Discarded packets • Reassembly failures • Whether the DoNotFragment Bit is set • TOS (TCP/IP Type of Service) such as Telnet with low delay (interactive priorities overriding batch transmissions) • Small MTU/MSS (Maximum Transmission Unit/Maximum Segment Size) <p>Note: If the MTU is too small, inefficiency results whereas if it is too large, datagram fragmentation may result.</p>

Problem	Factors to Consider
Network issues (cont'd)	<p>At the TCP layer, look for:</p> <ul style="list-style-type: none"> • Segments retransmitted • Connections reset • Frequency of ACKs • Window size too small <p>In the TCP/IP stack, look for:</p> <ul style="list-style-type: none"> • The maintenance level of the two TCP stacks involved • The use of PORT and 1364 TCP CDSTC NODELAYACKS which may delay ACKS • The values of TCPSENBFRSIZE and TCPRCVBFRSIZE in the TCP/IP PROFILE data set <p>Note: These values affect all applications using the TCP/IP protocol whereas the V2.BUFSIZE initialization parameter (see below) affects the operation of the IBM Connect:Direct application only.</p> <ul style="list-style-type: none"> • Whether the value set for PATHMTUDISCOVERY is an MTU size of 8992 (the default) <p>In the IBM Connect:Direct global initialization parameters file, look at:</p> <ul style="list-style-type: none"> • “V2.BUFSIZE = (maximum transmission buffer size, TCP/IP send/receive buffer size)” on page 494–The default is V2.BUFSIZE=(32K,128K). Adjust if needed based on the bandwidth and speed of your communications lines. • DEBUG–Make sure this setting is 000000000 so that internal traces are turned off. • “TCP.API.TIMER = 00:00:00 hh:mm:ss” on page 487–To reduce the number of hung sessions, specify this keyword. Set this value to at least 20 minutes and specify TIMEOUT=YES on IUI signon panels for individual users. Session waits before timing out and exiting.

Problems Involving Executing IBM Connect:Direct Processes

This table lists factors to consider if a IBM Connect:Direct Process is executing inefficiently.

Note: Because FTP is a utility program integrated into the TCP stack (thus running at the dispatch priority of the TCP stack), FTP may transfer data at a faster rate than an external application when sending a single file from the same source to the same destination. The only time an external data transfer application exceeds the transfer rate of FTP is when parallel data transfers take place between the same source and destination. Using parallel data transfers between the same source and destination is how most large production environments operate.

Factors to Consider	Suggestions
Are you using compression? If so, what type?	<ul style="list-style-type: none"> • Compression is generally unnecessary unless you use a slow line. Send data in its original state. • If you use extended compression and want to see the effects of changing the default values for the parameters related to extended compression, see Increasing Throughput and Decreasing CPU Utilization.
Are you using checkpoints? If so, what is the interval?	On a fast link make this interval large, for example, 100M.

Factors to Consider	Suggestions
Are you sending text or binary files?	When comparing IBM Connect:Direct with FTP, send files only in Binary mode with both IBM Connect:Direct and FTP. Binary mode must be used with Text files because FTP strips trailing blanks and sends only a partial file. Note: IBM Connect:Direct only sends complete files unless sending HFS files, where trailing blanks can be stripped.
What does the file structure look like?	To speed up the transfer, use a larger blocksize.
Are you using striped extended-format data sets for files that have large amounts of data or in which time is of the essence?	Depending on the number of stripes, you could see a dramatic increase in the I/O rate.
Can you break down the Process so as to send multiple files at once using IBM Connect:Direct's parallel session capabilities?	For testing purposes, set the PARSESS parameter in the network map to at least 10 then submit 10 file transfers in IBM Connect:Direct and 10 in FTP. After verification, change the PARSESS value to fit your environment.
Are you changing DCB attributes?	Avoid giving the sending and receiving data sets different DCB attributes since that forces the transfer to "record mode," which increases CPU utilization and TCP or SNA I/O.

How to Improve BSAM Data Transfer Rates

To optimize BSAM sequential data set transfer rates, take one or more of the following approaches:

- If MAXSTGIO is currently defined in the initialization parameter file, review the setting and consider setting it to the 1 MB default or greater. To fine-tune and set the Number of Channel Programs (NCP) in the DCB parameter of the COPY statement, include the second positional parameter as well.
- Increase the block size when it is advantageous to do so. Make the block size of a disk data set close to (but not more than) half-track blocking (27998 for non-extended 3390 disk data sets or 27966 for extended 3390 data sets). This improves performance by increasing the number of bytes transferred per I/O. For example, when transferring a data set with an LRECL of 80, it takes much longer to transfer 27920 bytes in 349 blocks (BLKSIZE=80) than in 1 block (BLKSIZE=27920).

Note: Exceeding half-track blocking on disk wastes a significant amount of storage capacity without improving the transfer rate.

For tape-to-tape transfers, a larger block size improves both performance and capacity. For disk-to-tape transfers, the I/O performance benefit of reblocking to an LBI block size (> 32760) may be outweighed by the CPU performance hit of transferring the data set in "record mode."

Troubleshooting BSAM Data Transfers

Data transfer rates using BSAM vary significantly from run to run even on the same system. For example, even with a high NCP, the transfer rate deteriorates when the I/O subsystem is moderately busy. If problems should occur, review these factors:

- The REGION specified on the job card. With the higher number of I/O buffers comes the risk that if too many Processes run simultaneously, the above-the-line storage can be exhausted. To prevent out of storage abends, review both the MAXSTGIO initialization parameter and the job's region.
- The number of concurrent transfers occurring on one DTF, even when all system components (CPU, DASD, CU, CHPID or network) run below capacity. For example, where a single transfer attains a transfer

rate of 76 MB per second, two concurrent transfers potentially reduce it to 66 MB per second for each transfer, three transfers to 58 MB per second, and 4 transfers to 52 MB per second.

- Hardware caching. When you run the same test case multiple times, usually the first runs slower than subsequent runs. For example, the initial transfer rate might be 38 MB/second, which increases to 75 MB/second for the second and subsequent runs.
- Network. The transfer rate varies widely according both to the speed of the network and the volume of traffic on it.
- Data set's device, CU, and CHPID configuration, speed, and how busy they are.
- CPU speed, and how busy it is (sometimes a limiting factor).
- Compression. Compression sometimes slows down the transfer due to extra CPU use.

Problems Involving Checkpoints

If the value specified for the checkpoint interval either as an initialization parameter or in the TO clause of a COPY statement is too small, it can significantly reduce transmission speed. Since the purpose of checkpointing is to save time in a restart, it is usually unnecessary to have a checkpoint interval that translates to less than a second of transmission time. A checkpoint interval that translates to 5 seconds of transmission time between checkpoints is normally more than adequate. For example, if the transmission rate to the other node is typically 10 MB/second, and you desire to lose no more than 5 seconds of transmission time in a restart, then you would set CKPT=50M.

Increasing Throughput and Decreasing CPU Utilization

Depending on your company's computing environment, IBM Connect:Direct for z/OS provides a variety of methods to improve how efficiently your system is running, including the following:

- The Compression Control Feature, which allows you to control the type of compression used between nodes
- Extended compression for environments using slow lines and high CPU capacity
- The zIIP exploitation feature, which takes advantage of System z Integrated Information Processor (zIIP) hardware to free up the general purpose processor resulting in potentially greater performance and cost-competitiveness when compared with less efficient and more complex distributed network solutions

In addition, if you are using Connect:Direct Secure Plus, there are other factors you should take under consideration to fine-tune performance in this area.

Compression Control Feature

The Compression Control Feature (CCF) allows you to control the type of compression used between nodes, that is, standard versus extended, and how compression is used on a node, that is, whether the selected compression type is required, may be used, or is not allowed at all. This control may be forced globally by the PNODE or negotiated by the SNODE. Global compression control settings are specified in the initialization parameters, whereas settings for specific nodes are specified in the adjacent node entry for those nodes.

Because this feature overrides compression settings specified in a Process, its default settings allow both types of compression so that the Process determines the type of compression that will be used (standard "COMPRESS" or extended "COMP EXT").

When control of compression is being exercised, the PNODE always applies its rules to the Process and then attempts negotiation. Negotiation is not performed if the other node does not have the CCF code installed. (The node having Process control is in charge in this case.)

Where a partner node also has CCF installed, then the two nodes will negotiate the type of compression that will be used. If one node says that compression must be used and the other says that it is not allowed, then the step will fail, causing an IMPASSE situation (SCPA994I). Also, if one node says FORCE EXT and the other replies FORCE STD, this is also an IMPASSE situation.

The parameters that govern compression control come from either or both the following initialization parameters and network map parameters:

- Initialization parameters
 - COMPRESS.EXT
 - COMPRESS.STD
 - COMPRESS.NETMAP.OVERRIDE
 - COMPRESS.NEGO.FAIL
- Network Map parameters
 - COMPRESS.EXT
 - COMPRESS.ST
 - COMPRESS.STD.PRIMECHA

The CCF initialization parameters are refreshable, so they may be changed at any time without requiring the DTF to be recycled. When you do make changes to these initialization parameters, your changes will cause global overrides to Processes on your system(s). For example, where standard compression is being used in Processes, if the initialization parameters are changed to specify COMPRESS.STD=DISALLOW, none of your Processes will use standard compression (as long as netmap overrides are disallowed or the node definition does not contain compression control keywords).

If you want to put the netmap compression control keywords into effect, you have to specify values for those keywords in the ADJACENT.NODE definitions because they do not have default values.

For example, if the netmap is changed for a node (or nodes) to specify COMPRESS.EXT=FORCE, then all Processes that run on that node will behave as if COMP EXT was specified in each COPY step.

Because of situations such as these, it is important for you to contact your partners to discuss how compression will be handled between the two IBM Connect:Direct systems to avoid IMPASSE copy failures.

CPU Usage Examples

You have several options for controlling compression on nodes that give you flexibility in determining whether to spend CPU cycles on compression on a global or node-by-node basis.

Stop Using Compression

To completely stop using compression, specify the following in the initialization parameters file:

```
COMPRESS.EXT=DISALLOW
COMPRESS.STD=DISALLOW
COMPRESS.NETMAP.OVERRIDE=DISALLOW
```

These values will prevent compression from being performed, and will not allow any netmap entries to override the compression controls with regards to Processes under this node's control.

If the partner node(s) have CCF, then they will respond to these negotiation specifications by either terminating compression, or by sending an SCPA994I, RC=8 message to show that their system specified FORCE as one of the compression types.

If the partner node does not have CCF, compression will be stopped only when this node is the PNODE (with the exception of PRECOMP files, which are precompressed files). If this system is the SNODE, then a message will be written to the RPLERRCK trace file when it is forced to do compression because the PNODE does not have CCF.

Prevent Standard Compression

To prevent only standard compression, specify the following:

```
COMPRESS.STD=DISALLOW  
COMPRESS.NETMAP.OVERRIDE=DISALLOW
```

In this case, only Processes specifying COMPRESS will be affected on this node. If the partner node has CCF, then all Process steps that specify COMPRESS will be forced to no compression. However, if the partner has specified COMPRESS=FORCE, then the step will fail with an SCPA994I, RC=8 message.

When a partner node is the PNODE and does not have CCF, then the compression specified in their Process will be performed. If the compression that is being done is disallowed on this node, then the node will write a message to the RPLERRCK trace file to note the situation.

Control Compression on a Node-by-Node Basis

To control compression on a node-by-node basis, you can specify the following initialization parameter (or simply allow the parameter to default):

```
COMPRESS.NETMAP.OVERRIDE=ALLOW
```

You can then set the netmap compression parameters for each node as needed to allow extended and/or standard compression.

You can also set up the reverse situation on a node-by-node basis, where only certain nodes are prevented from using compression. In this case, specify COMPRESS.NETMAP.OVERRIDE=ALLOW in the initialization parameters file and specify the following netmap parameters for the nodes that will not use compression:

```
COMPRESS.EXT=DISALLOW  
COMPRESS.STD=DISALLOW
```

Using Extended Compression

TCP/IP connections accommodate greater bandwidth to transfer large files than older technologies eliminating the need to compress data. Data can be sent in its original state saving both time and the need to decompress data once it has been transferred. However, if you have a slow line and high CPU capacity, compression may be warranted. Under other conditions, compression consumes a lot of CPU and slows transfer rates considerably while giving little if any return given the cost of CPU time.

Different Methods of Using Extended Compression

You can compress and store files in ZLIB-compressed format using one of the following methods:

- On a global basis using the Extended Compression (ECZ) initialization parameters. If you always transfer the same type of data, you may benefit by changing the global default values of the extended compression initialization parameters.
- On a Process basis using the EXT parameters in the COPY statement. If you send a variety of data types, it is probably more efficient to retain the default values of the initialization parameters and override them on a Process-by-Process basis using the COPY statement. For information on overriding the extended compression parameters in the COPY statement, see the [Connect:Direct Process Language](#) help.
- Through the DGASACMP batch utility. This option allows the CPU consumption, or overhead of extended compression, to be offloaded from IBM Connect:Direct to the standalone utility. The CPU time consumed by this utility can be much greater than that consumed by IBM Connect:Direct because the DGASACMP utility must use record mode compression whereas IBM Connect:Direct can use block mode compression. You can also use the DGASACMP batch utility to decompress the data and store it in its original format on the remote node where compressed data has been sent. The DGASACMP utility produces a report, which shows how much the data was read, written, compressed, and how long it

took to compress so you can determine the benefits of changing the default values of the extended compression parameters.

Changing the Values of ECZ Parameters

The effects of changing the default values for the ECZ.COMPRESSION.LEVEL, ECZ.MEMORY.LEVEL, and ECZ.WINDOWSIZE extended compression parameters are not always predictable and can significantly increase CPU utilization. The default values for the three parameters produce very good results for a wide variety of data types. Typically, it is only beneficial to change these default values if line speeds are limited, data is repetitive, and CPU is available.

zIIP Exploitation Feature

The zIIP Exploitation Feature (ZEF) is designed to offload CPU time to the System z Integrated Information Processor (zIIP), a special-purpose processor integrated into the System z server platform infrastructure. By freeing up the general purpose processor (CP), the efficiency of environments where CPU-intensive activities are being performed, such as ZLIB compression and SSL/TLS encryption, can be significantly enhanced.

zIIPs execute programs that are structured to operate under control of z/OS-preemptable enclave service request blocks (SRB). An enclave is a z/OS construct that allows a unit of work or transaction to be assigned a goal by the z/OS Workload Manager (WLM). If a program can operate under control of an enclave SRB, then it can be made eligible to run on a zIIP processor. The COPY operation, which is the core operation of IBM Connect:Direct for z/OS, uses multiple phases in a pipeline to copy data from source to destination. Because some of these phases are CPU-intensive, they are good candidates for zIIP-eligible work.

Requirements to Use the zIIP Exploitation Feature

The ZEF feature requires the following:

- Your System z must have one or more online zIIP processors.
- You must be running z/OS 1/10 or later.
- You must activate ZEF by performing one of the following actions:
 - Add the ZIIP parameter to your global or local initialization parameter file. See [“ZIIP = NONE | EXTCOMP | SSLTLS | ALL | PROJECT” on page 496](#) for a complete description of all settings. Because this parameter is not refreshable, you must restart IBM Connect:Direct to activate the ZEF feature.
 - Use the MODIFY command and specify the ZIIP setting you want to use. See [“IBM Connect:Direct MODIFY Command” on page 437](#) for more information.

Choosing the Appropriate ZEF Setting

The default value for the ZIIP parameter is NONE meaning that no enclave SRBs will be created and so no CPU time will be offloaded to a zIIP.

To offload all eligible activities to zIIP, you can select the ALL setting. These activities include all SSL/TLS encryption and decryption and extended compression and decompression performed during COPY steps. To check the results of using the ZEF feature, you can check the Copy Termination and Process Termination (PT) records in SELECT STATISTICS.

In the following example, no zIIP processor was online, so all work was done under a CP and is shown in the *Time on CP* field and no time (0) is shown in the *Time on zIIP* field. The *zIIP Qualify* time indicates the amount of time that could have been processed on the zIIP, had one been available 100% of the time it was needed. If work executes on a zIIP processor in a fraction of the time it takes on a CP, then the wall time for the entire process can be estimated as $(Time\ on\ CP - zIIP\ Qualify) + (zIIP\ Qualify * fraction)$. The fraction is needed because the *zIIP Qualify* time is normalized to standard processor speed (that is, the fraction is the normalization factor). For more information on how these new IBM Connect:Direct statistics are calculated, see [“Additional Information on How Time is Calculated for Statistics” on page 405](#).

=====			
CD.ART		SELECT STATISTICS	
		06.21.2011	
=====			

Function	=> Process Term	Start Time	=> 16:17:03
Process Name	=> ZIIP1	Stop Time	=> 16:17:06
Process Num	=> 1..	Comp Code	=> 00000000
		Comp Msg	=> SVTM100I
Userid	=> EPETE1		
Job Name	=> EPETE1	Job ID	=> TSU58526
Secondary Node	=> CD.B0B		
Time on CP	=> 00:00:13.449		
Time on zIIP	=> 00:00:00.000		
zIIP Qualify	=> 00:00:10.598		

You can project how much time could be offloaded to a zIIP without actually using the zIIP. If a zIIP processor is online, you can specify the PROJECT setting for the zIIP initialization parameter. All activities will continue being dispatched to the CP. If there is no zIIP processor online, there is no difference between the PROJECT and ALL settings.

To segregate extended compression and encryption activities, you can use the EXTCOMP setting to offload only extended compression and decompression CPU time to a zIIP or the SSLTLS setting to offload only SSL and TLS data encryption and decryption CPU time.

Potential Issue When Using ZEF

Because the ZEF feature does not constrain which SRBs are offloaded to zIIPs when ALL is specified, all available zIIPs could easily be dominated by IBM Connect:Direct when multiple concurrent COPY steps with extended compression are executing. To allow zIIP-eligible work to be executed on a CP if all zIIPs are in use, make sure to set the IEAOPTxx parameter, IIPHONORPRIORITY, to YES. For more information, see *IEAOPTxx (OPT parameters)* in *MVS Initialization and Tuning Reference (SA22-7592)* or *IEAOPTxx (OPT parameters)* for the online information center.

Additional Information on How Time is Calculated for Statistics

Note: IWMEQTME is a Workload Management (WLM) service documented in *z/OS MVS Programming: Workload Management Services*. For more information, refer to document number, SA22-7619.

The three new time-related statistics are derived as follows:

- Time on CP = (IWMEQTME CPUTIME) – (IWMEQTME ZIIPTIME)
- Time on zIIP = (IWMEQTME ZIIPTIME)
- zIIP Qualify = (IWMEQTME ZIIPQUALTIME)

If the installation has no zIIP processors:

Time on zIIP (IWMEQTME ZIIPTIME) = 0

Therefore, Time on CP = (IWMEQTME CPUTIME) – 0

However, regardless of whether the installation has a zIIP processor:

Time on CP (IWMEQTME CPUTIME) – (IWMEQTME ZIIPTIME) + Time on zIIP (IWMEQTME ZIIPTIME) = (IWMEQTME CPUTIME)

The (IWMEQTME ZIIPTIME) and (IWMEQTME CPUTIME) times in the Process Termination record example above will be the same (except for rounding errors) as reported by SMF and RMF for the enclave. But for step statistics reported in COPY and RUN TASK records, times are "slices" of the enclave times and will add up to slightly less than the corresponding Process Termination record times. These differences should be negligible. This occurs because there is a small amount of between-step processing charged to the enclave that is not charged to any particular step. All of the between-step time is CP time; none is zIIP qualified.

Because SSL and TLS handshakes occur before an enclave is created, their CPU time is not reported in any enclave time.

Because IBM Connect:Direct reports zIIP-qualified time, not zIIP-eligible time, the IEAOPTxx parameter setting for PROJECTCPU has no effect on the times in IBM Connect:Direct statistics. However, if PROJECTCPU=NO, there may be nothing in SMF or RMF to compare IBM Connect:Direct statistics to.

Considerations When Using Connect:Direct Secure Plus

When using Connect:Direct Secure Plus, be aware of the following :

- CPU utilization increases dramatically with every increase in the length of the encryption key. Use the lowest level of encryption allowed by your security policy.
- Whenever possible, use an encryption key that is supported in the z/ hardware (3DES or AES128).
- Even though extended compression is not recommended for high speed networks, using extended compression with files that compress well (80-90%) can reduce total CPU utilization, especially if the encryption key is not implemented in the hardware.
- If Connect:Direct Secure Plus is being used between two Connect:Direct for z/OS nodes (Version 5.0 or later), but not all files must be encrypted, consider using one of the following options:
 - Specify OVERRIDE=YES on the remote node record in the Connect:Direct Secure Plus parameter file and SECURE=OFF in the PROCESS statement.
 - Specify OVERRIDE=YES on the remote node record in the Connect:Direct Secure Plus parameter file and SECURE = (ENCRYPT.DATA=N) in your PROCESS or COPY statement. ENCRYPT.DATA=N tells IBM Connect:Direct to not encrypt the actual file data being copied but rather just the control block information, such as userid or password, used to establish a session.

Note: Both trading partners must support this capability.

Using zFBA for File Transfer

Large file transfers can cause performance issues. You can improve large file transfer performance by using the zFBA feature to transfer large files between Connect:Direct zOS and Connect:Direct UNIX.

Utilizing the zFBA feature you can improve performance for large file transfers between Connect:Direct zOS and UNIX because there is minimal TCPIP stack usage in the transfer. This can significantly decrease the CPU utilization in both the Connect:Direct zOS and TCPIP address spaces, as well as decrease the transfer duration.

You must ensure that a wide data bandwidth is in place to allow zFBA devices to maximize benefit from the interface. Smaller files will not benefit from this interface as performance will likely not improve; therefore, it is recommended that you not use the zFBA feature to process smaller files.

For additional information, see *z/OS MVS Programming: Authorized Assembler Services Guide*, SA23-1371-00 *zOS FBA Services*

How zFBA works with IBM Connect:Direct

IBM Connect:Direct utilizes a TCP/IP connection with its remote partner for its normal application (FMH) communication path, and two zFBA devices on the DS8K for data transfer.

Connect:Direct for z/OS uses an IOS API to allocate, read, write, unallocate and erase the zFBA devices. Connect:Direct for z/OS defines and manages the zFBA devices within its configuration in its NETMAP file and the remote IBM Connect:Direct partner is not required to define the zFBA configuration in its NETMAP file. IBM Connect:DirectUNIX only need to request that Connect:Direct for z/OS allocate two zFBA devices for the COPY.

In a non-zFBA COPY, the data and all FMHs are sent using TCPIP in units where the maximum size is determined through the negotiation between two nodes.

In a zFBA COPY step, the data (and any checkpoint FMHs) are transferred with a pair of zFBA devices, in units of 32MB (the last one is usually smaller). The zFBA devices must be defined to be at least 32MB and

any amount in excess of 32MB is unused by the IBM Connect:Direct zFBA feature. During the bulk of the transfer, the sending node is writing to one zFBA device in a pair; the receiving node is simultaneously reading the other zFBA device and writing the data to the destination data set. When the sender is done writing to a zFBA device, it communicates to the receiver over the TCPIP connection directing it to the new data. When the receiver has finished reading the data, it communicates back to the sender that the device is available and allows IBM Connect:Direct to reuse the zFBA device. The switching of zFBA devices occurs many times in a COPY step.

Special Considerations when using zFBA with IBM Connect:Direct

You can use zFBA devices for COPY between two Connect:Direct for z/OS nodes with the following requirements:

- Both Connect:Direct for z/OS nodes MUST run in separate LPARS
- Connect:Direct for z/OS nodes require the same zFBA configuration defined in both NETMAP files

The only protocols that support zFBA transfers are TCPIP and UDT; however, it is highly recommend that TCPIP be used. Any other protocol used will generate message SCZF002I and the COPY step will continue without using zFBA.

Once the data path is determined to be zFBA, if an allocation error occurs on z/OS or UNIX, the connection terminates and message SCZF003E or SCZF006E is generated. SCZF006E is a special case indicating that the PNODE and SNODE are on the same LPAR. SCZF003E errors are repeatable.

IBM Connect:Direct uses a fixed size of 32MB regardless of the actual zFBA device size; therefore, the minimum device size is 32MB. Each RU is rounded up to the next zFBA block size multiple (for the current generation DS8K, there are 512 bytes per block). As many whole RUs are packed in a 32MB chunk as will fit.

CDPLEX supports zFBA transfers.

You can not modify the zFBA setting using the INITPARM refresh. You must utilize the new global initialization parameter of zFBA so the WLM can direct the process to the correct server that supports zFBA.

The following is not supported when using zFBA: Secure+, CTCA, and any SNA protocol.

zFBA Error Handling

There are several possible errors that could indicate additional user action is required.

The following errors are possible when using zFBA:

Error SCZF0021

If a COPY statement has ZFBA=2, but the zFBA is not allowed for that process, an SCZF0021 occurs. The following list shows the Reason Text for Error Message SCZF0021.

```
1 = The INITPARM ZFBA=NO was specified or allowed o default
2 = ZFBA is not supported with SNA, UDT, or CTCA
3 = ZFBA is not supported with Secure Plus
4 = The Remote node does not support or is not configured for ZFBA
5 = The Netmap has no entry for the node
6 = There are no ZFBA device range for the node in netmap
```

If generated, the third line of the message explains why it is not allowed.

```
18.19.24 JOB02828 SCZF002I ZPUSH1 COPY ZFBA30S ( 1) SNODE=CD.BOB.NOZFBA 535
535 SCZF002I Not using zFBA for COPY data transport. Reason:
535 SCZF002I No ZFBA device range for node in netmap.
```

Error SCZF003E

An error occurred in IOSFBA ALLOCATE. The IOSFBA RC and RSN are included in the error message.

Example (message compacted):

```
13.59.48 JOB09406 SCZF003E ZPUSH1 COPY ZFBA01 (1) SNODE=CD.B0B 020
020 SCZF003E Cannot Allocate zFBA devices,RC=0C,RSN=01
```

The RC and RSN codes can be found in SYS1.MACLIB(IOSFBA).

Error SCZF005E

The FMH 76 is missing or invalid. A reason is included in the error message.

The following list shows the Reason Text for Error Message SCZF005E:

1. RECV error,or not FM76-L
2. Device Count ^= 2-10
3. RCVFM764 Error
4. RCVFM765 Error
5. SNDFM764 Error
6. SNDFM765 Error
7. Curr Dev^0;FM76-4 RECV not needed
8. Curr Dev^0;FM76-5 RECV not needed

Error SCZF006E

An error occurred in IOSFBA ALLOCATE in the SNODE in a CD z/OS to CD z/OS transfer. Since the PNODE ran first and was able to allocate the same ZFBA devices, this error is usually caused by the PNODE and SNODE running on the same LPAR (RC=0000000C & RSN=00000001). This error requires that you move a CD z/OS node to a different LPAR.

Example (message compacted):

```
14.22.14 JOB09417 SCZF006E ZPUSH1 COPY ZFBA01 (1) PNODE=CD.ART 600
600 SCZF006E Cannot Allocate zFBA devices,RC=0C,RSN=01
```

Using zEDC with IBM Connect:Direct

zEnterprise Data Compression (zEDC) is a compression acceleration capability that allows you to perform hardware-based data compression that streamlines data exchanges, saves on storage, and reduces CPU consumption. IBM now provides a new zlib version that supports the zEDC Express Accelerator for compression and decompression activities to improve throughput and CPU usage.

You can use zEDC with IBM Connect:Direct by implementing the new initialization parameter, ZEDC. If the zEDC Express Accelerator is available, the compression is performed by the hardware; otherwise, a new software version of zlib, 1.2.7, will be used. During product initialization, a new message appears indicating that this version of zlib is available and if the zEDC hardware is available.

With zEDC hardware compression, the object being compressed or decompressed is performed using the hardware or the software. This is determined based on the size of the first data block from the object passed to the IBM ZLIB code.

This threshold is controlled by a system wide global parameter specified in SYS1.PARMLIB member IQPPRMxx using global parameters DEFMINREQSIZE and INFMINREQSIZE.

Note: On IBM zEnterprise z15 and above processors IQPPRMxx is no longer valid. The z15 has no PCIE zEDC cards anymore and IQPPRMxx is solely used for PCIE. However, documentation for IQPPRMxx does provide the default values on z15 for the old global parameters. On a z15, those thresholds can be specified using environment variables _H_ZC_DEFLATE_THRESHOLD and _H_ZC_INFLATE_THRESHOLD. That allows the threshold to be tailored by application using ENVIRON DD input of the new variables. The valid values are in the range 1-9999999, where each number is a multiple of 1024.

When this feature is used, it is utilized by the extended compression feature of IBM Connect:Direct and is subject to the negotiated Compression Control Facility (CCF). When the PROCESS is executed, extended compression is negotiated if the ZIIP for extended compression is not enabled. The appropriate environment variable uses the zEDC accelerator and uses the buffer length restrictions defined by the operating system and zEDC. IBM Connect:Direct statistics data indicates the version of zlib being used and if the hardware zEDC Express accelerator was used.

During product initialization, when zEDC is specified as ON or by default, IBM Connect:Direct will inquire the availability of the zEDC hardware and provide the appropriate initialization message, SITA374I, on the status of the zEDC hardware. If the hardware is unavailable, IBM Connect:Direct will continue but software compression is forced.

For a detail overview, requirements and planning for zEDC, please reference, *z/OS MVS Programming: Callable Services for High-Level Languages, SA23-1377-00 zEnterprise Data Compression (zEDC)*.

Simultaneous Session and Asset Tracking Reporting

IBM Connect:Direct utilizes Simultaneous Session and Asset Tracking to continually monitor server sessions, track metrics such as the high-water mark, generate SMF records to document the values, and generate audit reports with this data.

This feature is enabled by default and enables Connect:Direct administrators manage their server licensing requirements to:

- Help determine if the current usage of Connect:Direct servers is within the license entitlement levels and can prevent potential license violations
- Track usage of IBM Connect:Direct server and its neighbors and assist administrators decommission devices that are no longer accessing IBM Connect:Direct servers.

IBM Connect:Direct sets several important Global initialization parameters that supply values to process its various functions. For example, SESSION.HIGHWATER.SMF parameter is set to define a session expiration time. When this time period elapses, the Connect:Direct server records high-session limits, the time/date it occurred, and other additional information about the server itself such as, License type. This information is recorded as a SMF record into a zO/S standard facility, SMF and into its Statistics facility, Connect:Direct Server's Statistics to be utilized by IBM Control Center and other AIJ applications.

Note: Global initialization parameter SESSION.HIGHWATER.SMF is by default set to 60. To modify this and other related parameters such as, SERVER.TYPE and SERVER.MSU see, [Global Initialization Parameters](#).

For example, Connect:Direct records and maintains the current session count internally and record this externally by writing the information to SMF record type 133. The SMF record type and the recording interval will be established at initialization time as specified with a new initialization parameter. Each Connect:Direct Server will record this information separately.

If the SMF record type of 133 is being used by another application, then you will need to specify a new SMF record type using the SESSION.HIGHWATER.SMF initialization parameter.

Post-processing of the SMF records will be done by a batch job, for which sample JCL and control cards are distributed in SDGAJCL that will produce audit reports from the data.

Additionally, this data is also recorded as a H2 record type in Server Statistics. Use SELECT STAT command to format a report for this record type. For more information see, *SELECT STATISTICS Command* in *IBM Connect:Direct for z/OS User Guide*.

SMF Record Layout

The new SMF record layout is in the table below. It specifies the SMF TYPE or is defaulted to by the new INITPARM (CDHWRTY=XL1'85'=133). It contains a subtype of 2 (CDHWSTY=XL2'0002'), and a subsystem identification of CDHW (CDHWSSI=CL4'CDHW ') to aid in identification. The first 24 bytes are the standard SMF record header for records with subtypes. The remaining bytes are the High Water mark information.

Note: For programs, such as DFSort, that reference field position rather than field offset, add 1 to the offset for control card usage. For programs, such as REXX, EXECIO and ISPF BROWSE/EDIT, which do not handle the RDW field, subtract 4 from the offset to calculate field position.

For a sample layout, see the table below.

Table 1. Mapping of the CDzOS High Water Mark SMF Record

Offsets		Name	Length	Format	Description
00	00	CDHWLEN	2	binary	Record length. This field and the next field (total of four bytes) form the RDW (record descriptor word).
02	02	CDHWSEG	2	binary	Segment descriptor
04	04	CDHSFLG	1	binary	System indicator:
05	05	CDHWRTY	1	binary	Record type, from parameter 1 of the SESSION.HIGHWATER.SMF INITPARM.
06	06	CDHWTME	8	binary	Time since midnight, in hundredths of a second, when the record was moved into the SMF buffer.
10	0A	CDHWDTE	4	packed	Date in the form 0cyydddF when the record was moved into the SMF buffer. See Standard SMF record header for a detailed description.
14	E	CDHWSID	4	EBCDIC	System ID (from SMCASID).
18	12	CDHWSSI	4	EBCDIC	Subsystem ID c'CDHW'.
22	16	CDHWSTY	2	binary	Record subtype x'0002' = Session High Water Mark
24	18	CDHWSNAM	8	EBCDIC	System name from CVTSNAME.
32	20	CDHWJOB	8	EBCDIC	Job name of CDzOS server.
40	28	CDHWITME	4	binary	Time, in the form HHMMSSSTH, when the CDzOS server was initialized.
44	2C	CDHWIDTE	4	packed	Date in the form yyyydddC when the CDzOS server was initialized.
48	30	CDHWJID	8	EBCDIC	8-character job identifier
56	38	CDHWPLX	8	EBCDIC	CDPlex name or blanks
64	40	CDHWSRV	8	EBCDIC	CDPlex server name or blanks
72	48	CDHWNOD	16	EBCDIC	CD local node name
88	58	CDHWUNU1	1		Unused
89	59	CDHWRCR	1	EBCDIC	Record Creation Reason: 'I'=Interval (CDHWRINT) reached 'T'=Termination
90	5A	CDHWRINT	2	binary	Recording Interval, in minutes, from parameter 2 of the SESSION.HIGHWATER.SMF INITPARM.

<i>Table 1. Mapping of the CDzOS High Water Mark SMF Record (continued)</i>					
Offsets		Name	Length	Format	Description
92	5C	CDHWMAXP	2	binary	Maximum Allowed Concurrent Processes, from MAXPROCESS INITPARM.
94	5E	CDHWSHWM	2	binary	Session High Water Mark for this interval.
96	60	CDHWHTME	4	binary	Time in the form HHMMSSTH, when the Session High Water Mark was first reached in this interval.
100	64	CDHWHDTE	4	packed	Date in the form yyyydddd when the Session High Water Mark was first reached in this interval.
104	68	CDHWPRCT	2	Binary	MAXPRIMARY Sessions
106	6A	CDHWSECT	2	Binary	MAXSECONDARY Sessions
108	6C	CDWHOS	4	EBCDIC	OS Type
112	70	CDHWOSVR	4	EBCDIC	OS Version
116	74	CDHWCDVR	4	Binary	CD Server Version
120	78	CDHWLIC	8	EBCDIC	License Edition
128	80	CDHWTyp	4	EBCDIC	License Sub-Type PROD/TEST
132	84	CDHWMSU	4	Binary	MSU Value
136	88	CDHWGMT	4	Binary	GMT Offset
140	8C	CDHWNNUM	4	Binary	Number of Nodes
144	90		2		Reserved
146	92	CDHWHSTL	2	Binary	Length of Hostname
148	94	CDHWHOST	255	EBCDIC	Hostname
403	193		5		Reserved
408	198	CDHWCNT	8	Binary	Number of times HW Count reached
416	1A0		96		Reserved

H2 Stat Record layout

The H2 statistics record is generated and written to the Connect:Direct Statistics file at the same interval the SMF record is generated. This H2 statistics record can be formatted as a report with the Select Statistics command or the SS or S2 selection from the Connect:Direct IUI.

<i>Table 2. Mapping of the CDzOS High Water Mark SMF Record</i>					
Offset Decimal	Offset Hexadecimal	Name	Length	Format	Description
00	00	H2RECLN	2	Binary	H2 Record length

Table 2. Mapping of the CDzOS High Water Mark SMF Record (continued)					
Offset Decimal	Offset Hexadecimal	Name	Length	Format	Description
02	02	H2RTYPE	2	EBCDIC	H2 Record Id
04	04	H2TIME	4	Binary	Time when STAT record is written
08	08	H2DATE	4	Binary	Julian Date when STAT record written
12	0C	H2HSNAM	8	EBCDIC	zOS System Name
20	14	H2HJOB	8	EBCDIC	Server JOB Name
28	1C	H2HITME	4	Binary	Server Initialization Time
32	20	H2HIDTE	4	Packed	Server Initialization Date
36	24	H2HJID	8	EBCDIC	Server JOB Number
44	2C	H2HPLX	8	EBCDIC	CD Plex XCF Group Name
52	34	H2HSRV	8	EBCDIC	CD Plex Member Name
60	3C	H2SNOD	16	EBCDIC	Server Nodename (LNODE)
78	4E	H2HRINT	2	Binary	HW Recording Interval
80	50	H2HMAXP	2	Binary	Maximum Process (MAXPROCESS)
82	52	H2HSHWM	2	Binary	High Session value
84	54	H2HHTME	4	Binary	Time Reached in this Interval
88	58	H2HHDTE	4	Packed	Date Reached in this Interval
92	5C	H2HPRCT	2	Binary	Primary Session Count
94	5E	H2HSECT	2	Binary	Secondary Session Count
96	60	H2HOS	4	EBCDIC	OS Type
100	64	H2HOSVR	4	EBCDIC	OS Version
104	68	H2HCDVR	4	Binary	CD Server Version
108	6C	H2HLIC	8	EBCDIC	License Edition
116	74	H2HTYP	4	EBCDIC	License Sub-Type PROD/TEST
120	78	H2HMSU	4	Binary	MSU Value
124	7C	H2HGMT	4	Binary	GMT Offset
128	80	H2HNUMN	4	Binary	Number of Nodes
134	86	H2HHSTL	2	Binary	Length of Hostname
136	88	H2HHOST	255	EBCDIC	Hostname

Reporting

Several audit reports can be generated using the SMF data.

- Detail Report
- Summary Report
- CSV output file for importing into an Excel Spreadsheet

The SDGAJCL contain the following samples:

- DGAJDET – Collect SMF data using IFASMFDP and create report using DFSORT
- DGAJSUM – Collect SMF data using IFASMFDP and create report using DGA#SUM
- DGAJCSV – Collect SMF data using IFASMFDP and create report using DGA#CSV

Examples

Example 1 – SMF Record Detail Report. Sorted by SMFID, Date and Time of Write.

107/30/18 02:51:10 Simultaneous Session Interval Detail by: SMFID, Date and Time											
107/30/18 02:51:10 Simultaneous Session Interval Detail for SMFID: IRVK											
0---SMF Record Written-----Interval High Water Mark-----											
SID	Date	Time	Date	Time	Value	Pri	Sec	R	Jobname	Job ID	Name

IRVK	2018.201	06:39:14.29	2018.201	06:38:12.90	0	0	0	I	EPETE1A1	JOB02927	CD.ART
IRVK	2018.201	06:40:13.34	2018.201	06:39:14.29	0	0	0	T	TIC CP	DIAGNOS	CD.ART
IRVK	2018.206	15:43:39.76	2018.206	15:43:21.06	0	0	0	T	EPETE1A1	JOB03081	CD.ART
IRVK	2018.206	15:45:03.25	2018.206	15:44:02.24	0	0	0	I	EPETE1A1	JOB03083	CD.ART
IRVK	2018.206	15:45:03.28	2018.206	15:44:02.24	0	0	0	I	EPETE1B1	JOB03084	CD.BOB
IRVK	2018.206	15:46:00.07	2018.206	15:45:03.28	0	0	0	T	EPETE1B1	JOB03084	CD.BOB
IRVK	2018.206	15:46:00.16	2018.206	15:45:03.25	0	0	0	T	EPETE1A1	JOB03083	CD.ART
IRVK	2018.206	15:47:26.10	2018.206	15:46:23.98	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:47:26.14	2018.206	15:46:24.01	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:48:26.86	2018.206	15:47:26.10	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:48:26.88	2018.206	15:47:26.14	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:49:27.90	2018.206	15:48:26.86	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:49:27.93	2018.206	15:48:26.88	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:50:27.90	2018.206	15:49:27.90	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:50:27.93	2018.206	15:49:27.93	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:51:28.95	2018.206	15:50:27.90	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:51:28.98	2018.206	15:50:27.93	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:52:29.40	2018.206	15:51:28.95	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:52:29.43	2018.206	15:51:28.98	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
CD.BOB											
IRVK	2018.206	15:53:29.40	2018.206	15:52:29.40	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:53:29.43	2018.206	15:52:29.43	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:54:29.40	2018.206	15:53:29.40	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:54:29.43	2018.206	15:53:29.43	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:55:29.60	2018.206	15:54:29.40	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:55:29.63	2018.206	15:54:29.43	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:56:30.65	2018.206	15:55:29.60	0	0	0	I	EPETE1A1	JOB03085	CD.ART
CD.ART											
IRVK	2018.206	15:56:30.68	2018.206	15:55:29.63	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:57:30.65	2018.206	15:56:30.65	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:57:30.68	2018.206	15:56:30.68	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:58:30.65	2018.206	15:57:30.65	0	0	0	I	EPETE1A1	JOB03085	CD.ART
CD.ART											
IRVK	2018.206	15:58:30.68	2018.206	15:57:30.68	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	15:59:31.70	2018.206	15:58:30.65	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	15:59:31.72	2018.206	15:58:30.68	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	16:00:31.70	2018.206	15:59:31.70	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	16:00:31.72	2018.206	15:59:31.72	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	16:01:32.75	2018.206	16:00:31.70	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	16:01:32.77	2018.206	16:00:31.72	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	16:02:32.75	2018.206	16:01:32.75	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	16:02:32.77	2018.206	16:01:32.77	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	16:03:32.75	2018.206	16:02:32.75	0	0	0	I	EPETE1A1	JOB03085	CD.ART
IRVK	2018.206	16:03:32.77	2018.206	16:02:32.77	0	0	0	I	EPETE1B1	JOB03086	CD.BOB
IRVK	2018.206	16:04:33.80	2018.206	16:03:32.75	0	0	0	I	EPETE1A1	JOB03085	CD.ART

Isolating Problems

To expedite the diagnostics process, prepare the following information for the IBM Support representative:

- Version, release, maintenance level, and operating system for the IBM Connect:Direct products being used on both nodes, for example, Connect:Direct for z/OS 6.0 with Connect:Direct for HP NonStop 3.4.00.
- Release levels of the operating system, VTAM, and other software involved
- Details describing the complete scenario, including:
 - Commands that are issued, with exact syntax and order
 - Files or devices that are involved. Note the file contents and the type of file (such as DCB file attributes, GDG, Tape, SMS, striped, compressed)
 - Interface that you used, such as batch interface, IUI, or operator interface

- Error messages for both nodes
- System logs for both nodes, including any messages generated while the problem occurred. Check the SYSLOG for Connect:Direct for z/OS nodes
- Which side is PNODE/SNODE
- The direction of the data transfer
- What connection protocol you are using (LU0, LU6.2, IBM TCP/IP, TCPAccess)
- If the node is a IBM Connect:Direct/Plex or stand alone server
- IBM Connect:Direct Process including the FILE attributes for the FROM and TO files for both nodes
- I/O device types
- Network map information
- Statistics information for both nodes including ALLOC information

Note: Be prepared to recreate the problem. Problem recreation is your responsibility.

Supporting Documentation

Problem determination can be involved and can require extensive research. We ask you to gather some of the following supporting documentation to help analyze the problem. Not all of this information is applicable to all problems or all operating environments.

- SYSLOGs
- IBM Connect:Direct statistics for both nodes
- RPLERRCK output
- ESTAE output
- VTAM definitions
- APPLID definitions
- Logmode table entry
- Class of Service (COS) entry
- D NET VTAM display
- IBM Connect:Direct traces
- Network map information

You can use your network management software or VTAM commands to isolate any session-related problems.

Diagnostic Rexx Exec for IUI Environments

If you are experiencing problems with the IBM Connect:Direct IUI, you can use the DGA#CAP Rexx exec available in the SDGAISPC target library to facilitate diagnosis of IUI problems.

To invoke the DGA#CAP Rexx exec, type the following TSO command:

```
TSO %DGA#CAP
```

The exec issues an ISPF long message containing the dataset name where the output is written. Contact IBM Support and give the Support representative the output generated by the DGA#CAP Rexx exec.

Providing Dumps to Customer Support

A IBM Connect:Direct ABEND can occur when a system failure or system error exists or when the FORCE parameter is used with the STOP CD command.

When an ABEND is reported, Support searches the problem tracking database for any similar problems. Often the ABEND is a known bug or a common error, and a solution is readily available. Otherwise, provide a full SVC dump for diagnosing ABENDs.

If support cannot locate a reference to the ABEND, you may be required to provide the following information:

- Send a complete SVC dump, not a snap dump.
- A copy of the Process and the statistics for that Process. If the Process has symbolics, include the symbolic substitution data. Also, ensure that the statistics records include WTO records.

Note: Include statistics from both IBM Connect:Direct nodes.

If the ABEND occurred while executing a Process that has executed successfully, determine what changes were made, either in the operating system or within IBM Connect:Direct, and do the following:

- Send console logs for both IBM Connect:Direct nodes.
- Note whether the ABEND caused either IBM Connect:Direct node to terminate.
- If the ABEND can be recreated, provide details.
- Send a copy of the system log and network error log for both operating systems, which can indicate any unusual situations occurring with the operating system or network at the time of the ABEND.

Note: A system log is required when analyzing an ABEND. It is preferable to review the log for both systems; however, it is essential for the system reporting the ABEND. When one of the nodes is not in a z/OS environment, check the output files for IBM Connect:Direct.

- Send RPLERRCK DD output to review I/O errors and other information. See [DD Statements in Startup JCL](#) for more information.
- Send ESTAE DD output to review ABEND conditions and some special I/O errors. See [DD Statements in Startup JCL](#) for more information.

If several ABENDs occur simultaneously, send the dump from the first ABEND only. Subsequent ABENDs are usually a result of the original ABEND.

When sending a dump on tape, send the JCL that created the tape. DSN, VOLSER, LABEL, and DCB attributes are needed to facilitate tape unloading. If available, send a printout of the tape management product display of the VOLSER.

Note: If you are sending a tape with more than one file, ensure that the JCL that created the tape references the correct file in the LABEL= parameter. This reference ensures that a previous file is not inadvertently overlaid.

Types of Dumps

To determine and resolve problems, Customer Support may request any of the following types of dumps:

- [IBM Connect:Direct Data Transmission Facility \(DTF\) Dumps](#), which include SYSMDUMP and CDSVCDMP dumps
- [IBM Connect:Direct IUI ABEND Dumps](#), which include z/OS Time Sharing Option (TSO) address space dumps
- [DGADBATC Batch Dumps](#)
- [VSAM File Dumps](#)

IBM Connect:Direct Data Transmission Facility (DTF) Dumps

This dump is generated when an ABEND occurs. It contains the contents of the IBM Connect:Direct address space, which is copied into one or more of the data sets that you specify by ddname in the DTF JCL (CDSVCDMP, SYSABEND, or SYSMDUMP).

When an ABEND occurs, IBM Connect:Direct produces an SVC dump with all of the information for the address space, regardless of whether the ABEND occurred in the main task or a User Exit subtask. Unless

otherwise specified in the JCL, the dump is written to the standard SVC dump data set, SYS1.DUMPxx. You can specify an alternate data set by using the CDSVCDMP DD statement in the JCL.

Note: If the system attempts an SVC dump and fails with: *ERROR* Unable to take an SVC dump; reason: 0B, the Dump Analysis and Elimination (DAE) component of the operating system found an earlier dump of this problem already exists.

Turn off the SVC dump by modifying your JCL. For more information, see [Turning Off the SVC Dump](#).

Multiple DTF SVC Dumps Capture Using CDSVCDMP

You can use CDSVCDMP to capture multiple DTF SVC dumps. A dump is created for each ABEND and written to a separate data set. You can specify whether the dumps are written to the SYS1.DUMPxx data sets or to unique user-specified data sets.

To write dumps to user-specified data sets, you must create a data set name for each dump that can occur. The first data set name must end with .SYSMDP00. Each additional data set name must end with .SYSMDPnn, where nn is a consecutive number up to a value of 99. For example, if you want to create enough data sets to write five dumps, create five data set names beginning with xxxx.yyyy.SYSMDP00 and ending with xxxx.yyyy.SYSMDP04.

You must define the data sets with the same attributes as your SYS1.DUMPxx data sets. The data sets must be preallocated and on the same disk volume.

To record multiple SVC dumps, set the JCL statement for CDSVCDMP as follows.

```
//CDSVCDMP DD DSN=XXXX.YYYY.SYSMDP00,DISP=SHR
```

The first dump is written to the .SYSMDP00 data set. When an additional ABEND occurs, a dump is written to the next data set, .SYSMDP01. Each additional ABEND creates a dump to the next .SYSMDPnn data set as long as ABENDs occur and enough .SYSMDPnn data sets are available. If an ABEND occurs and all data sets are full, the dump is not created and a message is issued stating that all .SYSMDPnn data sets are full.

You do not have to empty or reset these dump data sets. When the DTF is initialized, and you are using the .SYSMDPnn data sets, IBM Connect:Direct writes over the existing data in the data sets. If you want to save the existing data, save the data sets using a different data set name before you restart IBM Connect:Direct.

Turning Off the SVC Dump

If you want to turn off the SVC dump, place the following statement in your DTF JCL.

```
//CDSVCDMP DD DUMMY
```

Changing Dump Options

For the SDATA parameter specify SDATA=(ALLSDATA). If SDATA=(ALLSDATA) is not an acceptable default for your system, ensure that the SYS1.PARMLIB member corresponding to the ddname in the JCL for the DTF specifies, at a minimum, the following.

```
SDATA=(NOSUM,PSA,RGN,SQA,SWA,TRT,LPA,GRSQ,CSA,NUC)
```

For the PDATA parameter for IEADMPxx and IEAABDxx, specify PDATA=(ALLPDATA).

Note: PDATA is not an option for member IEADMRxx.

If you cannot specify ALLPDATA, at a minimum, include PSW, REGS, SA, JPA, SPLS, and SUBTASKS. Refer to the IBM manual *MVS Initialization and Tuning Reference* for the release of z/OS being used.

If you are unable to change the SYS1.PARMLIB member, issue an operator command to change the dump options.

The following steps guide you in issuing operator commands to change dump options:

1. Issue the command `DISPLAY DUMP,OPTIONS` to list the dump options currently in effect.
2. Issue the command `CHNGDUMP SET` to change the options.
3. Issue the `CHNGDUMP DEL` or `CHNGDUMP RESET` command to reset the options after recreating the dump.

If the correct dump options are specified in one of the SYS1.PARMLIB members, change the ddname in the JCL for the DTF to reference the ddname corresponding to that member.

For further information on changing dump options, refer to the *MVS System Commands* IBM manual for the z/OS being used.

IBM Connect:Direct IUI ABEND Dumps

About this task

When the IBM Connect:Direct abends, a system dump might be required for problem resolution. To obtain a dump of the TSO address space, allocate a SYSMDUMP DD to an appropriately defined data set, and run the ISPF with the TEST option. To send a dump of the IUI to a data set, allocate SYSMDUMP with `DISP=MOD` to account for the two dumps that IBM Connect:Direct produces.

To recreate an ABEND of the IUI to produce a dump, perform the following steps:

Procedure

1. Ensure that the SYSMDUMP DD is allocated.
2. Reinvoke ISPF with the ISPF TEST command.
3. To allocate a SYSMDUMP DD, type the following TSO command where filename is a valid DSN name that has `LRECL=4160`, `BLKSIZE=0`, and `RECFM=FBS`.

```
TSO ALLOC F(SYSMDUMP) DA(filename) MOD
```

The system will determine blocksize, and will prevent short blocks. Generally, 110 cylinders on a 3390 device is enough space for an IUI dump.

4. When the ABEND message is displayed, press ENTER to produce a dump. Two dump messages (and dumps) are produced. Press PF3/END to bypass the dump.

DGADBATC Batch Dumps

For DGADBATC batch dumps, ensure the JCL that executes DGADBATC contains a SYSMDUMP DD statement. Refer to [“IBM Connect:Direct Data Transmission Facility \(DTF\) Dumps”](#) on page 415 to ensure that the appropriate dump options are specified for the corresponding ddname in the JCL for DGADBATC.

VSAM File Dumps

If a problem occurs when using a IBM Connect:Direct VSAM file, you might need a dump of the VSAM file for problem resolution. To dump a Connect:Direct for z/OS VSAM file, use the IBM Access Method Services (IDCAMS) PRINT command. If you send an IDCAMS PRINT of the TCQ, include the TCX file.

Refer to the sample job stream in the following example for creating a Connect:Direct for z/OS VSAM dump.

```
//XXXXXX JOB (1111),PROGRAMMER,NOTIFY=TSOID,CLASS=D,MSGCLASS=X,
//          MSGLEVEL=(1,1)
//PRINT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
PRINT INDATASET(DUMP1.ZOS.VSAM1)
DUMP
```

To copy a VSAM file to tape, use the IDCAMS REPRO command. You must include the DCB parameters, RECFM=VB, DSORG=PS for the data set created on the tape. In the LRECL and BLKSIZE parameters for the dataset created, you must reflect the size specified for the RECORDSIZE parameter used when you defined the file.

For example, if the IBM Connect:Direct statistics file was initially defined with RECORDSIZE (4089 4089), copy the IBM Connect:Direct statistics file by typing the following DCB parameters in the DD statement in the JCL.

```
//OUTDD DD DSN=TAPE.STAT.FILE,
//          DCB=(LRECL=4089,BLKSIZE=4089,RECFM=F,DSORG=PS)
```

Suppressing Dumps for Specific ABEND Codes

About this task

Use the following procedure if you want to suppress the dump for specific ABEND codes.

Procedure

1. Add the ABEND code for which you want to suppress dumps to the ABEND codes list defined in the ABEND.CODES.NODUMP parameter.
2. Set the ABEND.RUNTASK parameter to ABEND.CODES.NODUMP. Following is an example:

```
ABEND.CODES.NODUMP => (SX37 SX13 U0728 SXD9 S9FC)
ABEND.RUNTASK => ABEND.CODES.NODUMP
```

Resolving the Problem

If Support cannot resolve the problem during, it is documented as a case and is assigned a reference number. Refer to this number when forwarding documentation or calling support.

Severity Level	Definition
Severity 1	Production system is down; requires immediate attention. For example, excessive abnormal termination.
Severity 2	A major problem or question; the product operates but is severely restricted. For example, an incorrect response from a frequently used command.
Severity 3	A non-critical issue is encountered with the product, but the majority of functions are still usable. For example, an incorrect response from an infrequently used command.
Severity 4	A minor problem or question that does not affect product function. For example, the text of a message is misspelled.

All cases are prioritized based on severity.

Escalating a Problem Resolution

If our normal support cycle does not produce the results you require or your issue has changed in severity, you can escalate the case. To escalate a case, contact the technician responsible for your problem.

Reviewing IBM Connect:Direct Messages and Sense Codes

As you research a problem, note any messages issued by IBM Connect:Direct. Look specifically for messages displayed by the API through the IUI or batch interface. If an error occurred in a file transfer, look at messages for any Processes executing at that time for both nodes.

Also, check for messages in the JES log for z/OS.

IBM Connect:Direct messages contain short and long text to explain the error. Connect:Direct for z/OS displays the short text when an event occurs. The following information is displayed for each message if you request the long text:

- Module issuing the message
- Short and long message texts
- System action as a result of the message situation
- User response to correct the situation

Note: For z/OS, press PF1 to display the longer explanation if the message is issued in an ISPF panel.

IBM Connect:Direct sense codes address connection errors involving the netmap and the DGADETLU (alias of DMGNETLU) module. The following is an example of a sense code display containing both the long and short text. For additional information, see *Messages in IBM Connect:Direct for z/OS User Guide*.

```
Connect:Direct MESSAGE DISPLAY                                DATE => 2010.07.22
                                                            TIME => 09:53

MESSAGE ID==> SENS4022
MODULE      ==> DMGNETLU
=====
SHORT TEXT==> LUNAME does not match name of node in NETMAP
LONG TEXT:
  LINE 1 ==> The LUNAME or the remote node does not match the LUNAME
  LINE 2 ==> that is associated with the symbolic node name in the
  LINE 3 ==> C:D Network Map file. See messages SVTR014I and SVTR021I.
  LINE 4 ==>
  LINE 5 ==>
  LINE 6 ==> SYSTEM ACTION: The session is terminated unless NETMAP.CHECK
  LINE 7 ==> is specified in the INIT parms with the WARN parameter.
  LINE 8 ==>
  LINE 9 ==>
  LINE 10==> RESPONSE: The Adjacent.Node entry in the Network Map must be
  LINE 11==>                updated to reflect the proper LUNAME, OR the remote
  LINE 12==>                node should call in with correct LUNAME.

COMMAND ==> _____ ENTER 'DIR' TO DISPLAY THE DIRECTORY
```

Interactive Use of the IBM Connect:Direct Message Facility

Display the long text message through the IUI by either of the following methods:

- Use the SELECT MESSAGE command in the command line interface.
- Type M at the CMD prompt and press Enter.
- Access the PRIMARY OPTIONS MENU and type the MSG option at the CMD prompt.
- Type =M from any IBM Connect:Direct screen at the CMD prompt in the ISPF Interface and press Enter. The long text for the current message is displayed.

Displaying an ABEND Message

About this task

Some user ABEND messages are stored in the IBM Connect:Direct message file. To access them, insert the ABEND message ID in place of a IBM Connect:Direct message ID.

Note: System ABEND messages appear in the JES log for z/OS. They do not appear in the IBM Connect:Direct message file. Refer to the appropriate operating system manuals if you require further details.

For example, to display the description of user ABEND U0075, perform the following:

Procedure

1. Access the PRIMARY OPTIONS MENU in the IUI.
2. Select the MSG option.
3. Select option 1.
4. Type U0075 on the message ID line and press Enter.

```
Connect:Direct MESSAGE DISPLAY                                DATE => mm.dd.yyyy
                                                            TIME => hh:mm

MESSAGE ID==> U0075
MODULE    ==> DMINIT
=====

SHORT TEXT==> Connection services initialization failure.
LONG TEXT:
  LINE 1 ==> An error has occurred while initializing VTAM connection
  LINE 2 ==> services. ESTAE output should give a return code and error
  LINE 3 ==> flag.
  LINE 4 ==>
  LINE 5 ==> Verify the applid does not have a password associated with it
  LINE 6 ==> in the LOCAL.NODE definition of the Network Map.
  LINE 7 ==>
  LINE 8 ==> SYSTEM ACTION: ABEND the initialization of the DTF.
  LINE 9 ==>
  LINE 10==> RESPONSE: Verify that the APPLID is varied active and there is
  LINE 11==>                not a VTAM password specified in the Network Map.
  LINE 12==>

COMMAND   ==> _____ ENTER 'DIR' TO DISPLAY THE DIRECTORY
```

Examine Output from Select Commands

Problem determination and resolution is often as simple as gathering output from the IBM Connect:Direct SELECT commands and examining the output for obvious errors. For additional information, see the *IBM Connect:Direct for z/OS User Guide*. A description of the commands follows:

Command	Description
SELECT NETMAP	Displays or prints the definitions of nodes with which IBM Connect:Direct can communicate.
SELECT PROCESS	Displays or prints information about IBM Connect:Direct Processes in the TCQ. You can use this command before or during IBM Connect:Direct Process execution.
SELECT STATISTICS	Displays or prints statistics from the statistics log. Use this command only after executing a IBM Connect:Direct Process.
SELECT TASK	Selects, displays, or prints the list of all active users, Processes, and tasks within IBM Connect:Direct.
SELECT TYPE	Displays or prints type records from the Type file.

SELECT STATISTICS Command

The SELECT STATISTICS command is one of the most useful commands for problem determination. This section provides a brief overview of the SELECT STATISTICS command. For more detailed information, refer to the *IBM Connect:Direct for z/OS User Guide*.

If an error occurred during a file transfer, issue the SELECT STATISTICS command on both the PNODE and SNODE to review the statistics for the Process. Process statistics are stored in the IBM Connect:Direct statistics log. You can request them by either using the IUI Select Statistics (SS) screen, if applicable, or issuing the IBM Connect:Direct SELECT STATISTICS command.

The statistics log records the following types of information for IBM Connect:Direct Processes:

- Function requested (COPY, RUN TASK, RUN JOB, SUBMIT)
- Process name and number
- Start and stop times, and date of the function
- Completion code
- Messages associated with the Process
- Location and ID of the user requesting IBM Connect:Direct services
- Sending and receiving file names
- Amount of data sent and received
- Security violations
- All IBM Connect:Direct WTO messages, allocation information, and mount requests

Note: WTO messages are created during IBM Connect:Direct Process execution to document the execution steps and are stored in the IBM Connect:Direct statistics file. This type of message is an excellent debugging tool for determining a IBM Connect:Direct Process execution failure.

The optional parameters associated with the SELECT STATISTICS command enable you to define the search criteria and the form in which the report is presented.

IBM Connect:Direct IUI Select Statistics Screen

The Select Statistics screen provides a convenient, easy method for issuing the SELECT STATISTICS command. Use the fields to specify statistics selection criteria. The selection criteria enable you to determine what records to select from the statistics log, limit the statistics to a certain period of time, or limit the statistics to a certain Process. The selection criteria also allow IBM Connect:Direct to select the requested statistics from the file more efficiently.

The SELECT STATISTICS screen displays general IBM Connect:Direct Process information.

- To display the requested information in a formatted report, type a D in the CMD field and press Enter.
- To print the formatted report, type a P in the CMD field and press Enter. The output is sent to your TSO sysout.
- For a summary report display, type an S in the CMD field and press Enter.
- To view your statistics file in an unformatted display, type FIL or F in the CMD field and press Enter.
- The SELECT STATISTICS screen also enables you to exclude certain types of information from being displayed. Type a Y in any of the following fields and press Enter.
 - MEMBERS - Statistics generated for each partitioned data set (PDS) member
 - WTO - WTO (Write to Operator) messages
 - WTO - All statistics other than WTO messages

When statistics are requested by Support, do not exclude any information. Set all the fields listed to N. Support must receive all information to solve the problem.

Verify File Attributes

If a problem occurs during a file transfer, verifying the accuracy of the file attributes can often resolve the problem. Review the following file attribute parameters for the file being transmitted when the error occurred. Check both the input and output files specified in your IBM Connect:Direct Processes:

- Logical record length (LRECL)
- Block size (BLKSIZE)
- Record format (RECFM)
- File organization (DSORG)
- File disposition (DISP)
- Unit containing the file (UNIT)
- Volume serial number (VOLSER)
- Storage for allocating new files (SPACE)
- Optional processing code (OPTCD)
- Length of keys used in file (KEYLEN)
- Number of blocks or tracks to search for available space (LIMCT)
- SMS options and ACS rules

These parameters are described in detail at the [Connect:Direct Process Language help](#).

Common Errors

This section describes the following types of common errors:

- Signon and IUI/API errors
- Connect:Direct DTF session establishment errors
- Connect:Direct DTF out-of-storage ABENDS
- Allocation and open errors
- Transmission errors
- Operator interface errors

For each type of error, information on probable causes, actions to take, and data to collect is provided.

Note: For information on initialization and license key errors, see Initialization Errors in the *IBM Connect:Direct for z/OS Configuration Guide* that deals with errors that can occur when you start up. For errors related to security, see [Troubleshooting Security Errors](#).

Signon and IUI/API Errors

Signon errors keep you from accessing IBM Connect:Direct. IUI/API problems prevent you from successfully submitting a Process or executing a command.

This section describes the following types of signon and IUI/API errors:

- ISPF signon failures can be caused by a variety of problems. Errors signing onto the IUI through the ISPF interface can include problems related to VTAM, TCP/IP, ISPF, or security. Refer to [Troubleshooting Security Errors](#) for a description of common security errors.
- IUI/API IBM Connect:Direct session failures occur when the IUI/API cannot establish a session with the DTF. A session failure usually means you will be unable to sign on to IBM Connect:Direct.
- SELECT command errors occur while issuing SELECT STATISTICS, SELECT PROCESS, SELECT NETMAP, SELECT USER, SELECT TYPE, and SELECT TASK commands from the IUI, batch, or operator interface.

Note: You can diagnose most IUI/API problems by running an API-to-DTF session trace. See information on the NDMCMDS trace in [IBM Connect:Direct Automatic Traces](#).

To avoid insufficient space being allocated for the temporary data set upon signon as well as SB37 ABENDs, the administrator can define global signon defaults so that users do not have to individually alter their signon default values to increase their allocation. For more information, see [“Global Signon Defaults”](#) on page 181.

Condition: Signon to IUI Denied

Your signon to IUI is denied with the message *Error during ACB open*.

Error	Cause	Action	Collect
SVTB002I	The interactive applid that is used for signon is not active, is in an unacceptable state, or is not correctly defined to VTAM.	<p>Review both the short text and long text messages. Allocate NDMCMDS to display error messages to your terminal. NDMCMDS shows all actual commands issued to the DTF, including resolution of symbolics. It can be particularly helpful to debug IBM Connect:Direct commands through the IUI or through the batch interface if you are having signon problems, syntax errors, and so forth.</p> <ul style="list-style-type: none"> For Connect:Direct for z/OS, type the following on your command line: <p>TSO ALLOC FI(NDMCMDS) DA(*)</p>	<ul style="list-style-type: none"> NDMCMDS output Applid status display Applid definitions Network map

Condition: Signon to IUI Denied

Your signon to IUI is denied with the message *USER attempted SIGNON with TIMEOUT=NO*. Only an administrator can change the TIMEOUT value and only on their own signon panel. Non-ADMIN users cannot sign on with the inactivity timer disabled.

Note: The inactivity timer indicates the number of minutes a user can remain inactive without communicating with the DTF. This timer is active if an administrator has specified a value in the TCP.API.TIMER global initialization parameter.

Error	Cause	Action	Collect
SAFA020I	The user, who does not have administrative authority, attempted to change the TIMEOUT parameter value to NO from an IUI signon panel.	Log off and log on again. Change the TIMEOUT parameter to YES to complete the signon process.	<ul style="list-style-type: none"> Statistics file

Condition: Signon to IUI Denied - No Error Message

Your attempt to sign on to the IUI is denied, but no error message is displayed.

Error	Cause	Action	Collect
None	Your TSO or TSS profile specifies the NOWTPMSG option, which inhibits some error output from being displayed to the terminal.	Review both the short text and long text messages. For z/OS, change to the WTPMSG option by typing TSO PROF WTPMSG. With the WTPMSG option, error messages are displayed at the terminal. Retry the operation (sign on).	<ul style="list-style-type: none"> None

Condition: Signon Denied - IBM Connect:Direct not Active

Your signon is denied.

Error	Cause	Action	Collect
SVTB004I SCIA011I	The IBM Connect:Direct you are signing on to is not active.	<p>Review both the short text and long text messages. Ensure that IBM Connect:Direct has completed initialization before attempting a signon. Allocate NDMCMDS to display additional information about the session failure. NDMCMDS shows all actual IBM Connect:Direct commands issued to the DTF, including resolution of symbolics. It can be particularly helpful to debug IBM Connect:Direct commands through the IUI or through batch interface if you are having signon problems, syntax errors, and so forth.</p> <p>For z/OS, check to see that the network map is specified correctly on the ISPF menu and that the network map is correctly loaded.</p> <p>Try to sign on through the batch interface to isolate the problem.</p>	<ul style="list-style-type: none"> NDMCMDS output Network map IBM Connect:Direct initialization parameters For z/OS the ISR@PRIM panel

Condition: Signon Denied - Users Exceeds Limit

Signon is denied.

Error	Cause	Action	Collect
STAA004I	The number of interactive users on IBM Connect:Direct has reached the limit set in the MAXUSERS parameter.	Review both the short text and long text IBM Connect:Direct messages. Check the MAXUSERS parameter in the IBM Connect:Direct initialization parameters data set. If it is commented out, the default is six users. Report this error to your IBM Connect:Direct administrator, and determine whether you need to increase the value of this parameter. If the TCP.API.TIMER is not being used, consider using this global initialization parameter to terminate orphaned sessions (some JAI users do not sign off, they just drop the connection).	<ul style="list-style-type: none"> IBM Connect:Direct initialization parameters

Condition: TCP/IP API session terminated

Users who are logged on to the DTF via TCP/IP may have their sessions dropped and see the message *TCP/IP SEND error; Connection Lost*. If the user presses F1 to see the help for this message, the following text is displayed for message SVTC006I:

```
MSGID ==> SVTC006I
MODULE ==> DMTSTSND
```

TCP/IP SEND error; Connection Lost

An error has been detected while sending a command via TCP/IP. The TCP/IP connection has been lost. A possible cause is Connect:Direct has been shutdown and all TCP/IP connections have been closed.

System Action: Normal processing can not continue.

Response: Logoff and attempt to signon again.

In addition, SAFA019 signoff errors are recorded in statistics along with matching user IDs and the message, *TCP API session LOGOFF has been forced*. To reduce the number of active sessions and stay below the limit set in the MAXUSERS global initialization parameter, the administrator can use the TCP.API.TIMER global initialization parameter to specify the maximum time of session inactivity a TCP/IP-connected IUI or API session waits before timing out and exiting. When a user has not done anything with the IUI to cause it to communicate with the DTF and the timer period expires, the DTF drops the session, and can then detect and recover lost sessions without having to be recycled.

Error	Cause	Action	Collect
SVTC006I SAFA019	The TCP.API session has been inactive for the period specified by the TCP.API.TIMER global initialization parameter.	<ul style="list-style-type: none"> A TSO/ISPF/IUI user should log on to the DTF they lost connection with. A program or system that was using the API should sign on again. 	<ul style="list-style-type: none"> Statistics file—check the user and completion code (comp code) associated with the SAFA019I completion message (comp msg). On the SELECT STATISTICS screen, type Y in the CHANGE EXTENDED OPTS field and on the next screen, type SO (signoff) as a Record Type.

Condition: SELECT Command Issued Successfully - No Output Produced

The SELECT command is issued successfully and completes with a successful return code and message, but no output is produced.

Error Messages					
SAFF000I	SAFF014I	SAFK000I	SAFL000I	SAFL010I	SCBB000I
SCBL000I	SCBO000I	SCBP000I	SCBQ000I	SCBX000I	SOPA000I
SOPA011I	SOPE000I	SOPS001I			

Cause	Action	Collect
It is likely that IBM Connect:Direct is having trouble allocating the temporary data set that contains the output from the SELECT command.	Review both the short text and long text messages. For Connect:Direct for z/OS, specify a UNIT and VOLSER for the temporary data set, and SPACE information. You can specify UNIT and VOLSER on the SIGNON DEFAULTS panel of the IUI or use the TEMPDSN parameter on your signon command for batch. Note that a UNIT type of VIO is not acceptable.	<ul style="list-style-type: none"> List of the SELECT commands that produce output and those not producing output Any error messages

Condition: SELECT Commands - No Output Available

SELECT commands return with no output and a message indicating no output was available from the command.

Error	Cause	Action	Collect
SOPA010I SOPB012I	For the SELECT PROCESS and SELECT STATISTICS commands, it is likely that the userid issuing the command is defined as a general user by the stage 2 security exit, indicating that the user is only allowed to see the command output for Processes submitted by that same userid.	Review both the short text and long text messages. Check to see that the userid is defined with the ability to select Process/statistics for Processes not submitted by that userid.	<ul style="list-style-type: none"> Authorization for the userid Statistics file

IBM Connect:Direct DTF Session-Establishment Errors

IBM Connect:Direct DTF session-establishment errors prevent a successful connection between two IBM Connect:Direct systems. This section explains the most common causes of DTF session-establishment errors, actions to take, and the types of data you need to collect to troubleshoot the error.

Condition: Cannot Establish a Session with Another IBM Connect:Direct Session

The table describes the four possible causes and the courses of action to take.

Error	Cause	Action	Collect
SVTM026I SVTM045I SVTM053I SVTM104I	The links that connect the two IBM Connect:Direct systems are not active, or an error has occurred on the links.	Review both the short text and long text messages. If you are unable to determine the problem, check the RPLERRCK DD for possible clues. Use your network management software to determine the status of the links used for system-to-system communication. Reactivate the links.	• None
SVTM026I	The cross-domain resource definition for the remote IBM Connect:Direct system is not active.	Review both the short text and long text messages. If you are unable to determine the problem, check the RPLERRCK DD for possible clues. Use your network management software to determine the status of the IBM Connect:Direct cross-domain manager and cross-domain resource definitions used in communicating with other IBM Connect:Direct locations. Reactivate the cross-domain resource manager or cross-domain resources	• None
SNAS0801 SVTM026I	The VTAM applid for the remote IBM Connect:Direct system is not active.	Review both the short text and long text IBM Connect:Direct messages. If you are unable to determine the problem, check the RPLERRCK DD for possible clues. Use your network management software to determine the status of the applid for the remote IBM Connect:Direct system. Ensure the remote IBM Connect:Direct has initialized. Reactivate the VTAM applid, or initialize the remote IBM Connect:Direct. system.	• None

Error	Cause	Action	Collect
SCCS028I STAA005I	The maximum number of secondary sessions is reached on the secondary IBM Connect:Direct system.	<p>Review both the short text and long text messages. If you are unable to determine the problem, check the RPLERRCK DD for possible clues. Determine the number of active VTAM sessions for the secondary location. Use your network management software to issue D NET,ID=applid,E (VTAM applid) at the secondary location site. Ensure that the maximum number of secondary sessions is sufficient for your requirements.</p> <p>To avoid the situation that causes the STAA05I error, set the global initialization parameter, <u>"TCP.CONNECT.TIMEOUT= 0 number of seconds"</u> on page 487.</p>	<ul style="list-style-type: none"> • None

IBM Connect:Direct DTF Out-of-Storage ABENDS

DTF out-of-storage ABENDS may occur during heavy IBM Connect:Direct activity or during phases when the DTF has run for a long period of time. This chapter explains the most common causes of DTF out-of-storage ABENDs, actions to take, and the types of data to collect.

Condition: Out-of-Storage ABEND Occurs in the DTF

The following table describes two possible causes and associated actions to take.

Error	Cause	Action	Collect
user ABEND U0500 user ABEND U0501	If this condition occurs only during heavy IBM Connect:Direct activity, you may need to modify the initialization parameters or the DTF REGION parameter.	Review both the short text and long text messages. Limit the number of Processes that can run at one time using the MAXPRIMARY, MAXSECONDARY, and MAXPROCESS initialization parameters. Also, examine the MAXSTGIO initialization parameter to determine if it can be decreased. The REGION size in the DTF JCL may need to be increased to allow more concurrent Processes.	<ul style="list-style-type: none"> • TSubpool that is growing • Dump taken after controlled tests when all DTF activity has ended • The IBM Connect:Direct log • IBM Connect:Direct initialization parameters • IBM Connect:Direct STC (started task) JCL • Source for any user exits

Error	Cause	Action	Collect
System ABEND Sx0A System ABEND Sx78	If this condition appears to be "storage creep" and occurs after the DTF is active for a long time (not necessarily running many Processes immediately), you can take several actions.	<ul style="list-style-type: none"> See Possible Actions to troubleshoot this problem. 	<ul style="list-style-type: none"> Dump taken after controlled tests after all DTF activity ended The IBM Connect:Direct log IBM Connect:Direct initialization parameters IBM Connect:Direct STC JCL Source for any user exits

Possible Actions

Use the following list to help you troubleshoot DTF out-of-storage ABENDs:

- Examine all RUNTASK programs; ensure that for every file opened, a CLOSE and a FREEPOOL is also done.
- Examine any user exits for GETMAIN (or STORAGE OBTAIN) macros ; verify that FREEMAIN (or STORAGE RELEASE) macros are issued for each of those areas.
- Examine any RUNTASK programs for GETMAIN (or STORAGE OBTAIN) macros; ensure FREEMAIN (or STORAGE RELEASE) macros are issued for each area.
- Try to pinpoint the type of Processes or other DTF activity that causes the problem:
- Does this occur only during a COPY?
- Does this occur when a specific Process is run? What does the Process do?
- Does this occur when a certain command is issued? What is the command?

Note: For the next two suggestions, you can issue an MVS DUMP command for the Connect:Direct for z/OS address space to help in your investigation. Make sure that the SDATA includes the following parameters:

RGN,LSQA,SUM,PSA,GRSQ,SQA,SWA,TRT

See the IBM MVS System Commands manual for the release of z/OS in use.

- If you cannot determine which Process, command, or other activity is causing the storage creep, run a typical batch of Processes or commands that runs when the ABEND occurs. Before the out-of-storage ABEND occurs, go to the ADMIN MD panel and QUIESCE Connect:Direct for z/OS. For example, if the ABEND usually occurs after 10 hours of activity, quiesce after about 8 hours.
- If you did determine that a certain Process or command causes the problem, submit that Process or issue the command several times (the number of times depends on how long it takes before you get to the ABEND). For example, if it occurs after the Process runs 100 times, run it 90 times in your tests. Get a dump of the DTF address space after all DTF activity is finished.

Allocation and Open Errors

Allocation and open errors involve the source or destination files. This chapter describes errors in which the allocation or opening of a file fails, and the action to take and types of data you need to collect to troubleshoot the error.

Condition: Allocating a User File Fails

Use the following table to troubleshoot this error.

Note: IBM Connect:Direct initialization parameters (ALLOC.CODES and ALLOC.RETRIES) determine which allocation errors, if any, cause a Process that fails on an allocation error to be retried.

Error	Cause	Action	Collect
SDAA001I SDAA004I SDAA005I SDAA048I	IBM Connect:Direct received an error while allocating a file or data set.	<p>Review both the short text and long text messages. Check the SYSLOG, console, or IBM Connect:Direct statistics for the text of the SDAA004I and SDAB005I messages. The SDAA004I message contains the allocation parameters used by IBM Connect:Direct. If an error exists, the ERR=nnnn field of the SDAB005I message contains non-zeroes, and error text follows. Use the IBM Connect:Direct message facility to look up the error, which has a format of SDEnnnnI, where nnnn is the number in the ERR field. The operating system dynamic allocation routine returns the ERR value.</p> <p>Note: IBM Connect:Direct initialization parameters (ALLOC.CODES and ALLOC.RETRIES) determine which allocation errors, if any, cause a Process that fails on an allocation error to be retried.</p>	<ul style="list-style-type: none"> • IBM Connect:Direct allocation string (found in the WTO records in the statistics file) • IBM Connect:Direct Process involved • SDAA004I message output • SDAB005I message output

Condition: TCQ File Below Defined Threshold Value

Use the following table to troubleshoot this problem.

Error	Cause	Action	Collect
SPQL003I TCQ file is now below the defined threshold of &VAR.	The number of VSAM file CIs used has reached the defined threshold.	Reduce the number of Processes in the TCQ, or increase the size of the TCQ. Refer to “Configure the TCQ” on page 381 for more information.	<ul style="list-style-type: none"> • None

Transmission Errors

Transmission errors include consistency problems within communication components that can occur during Process execution. The errors can occur within communication components such as VTAM, IBM Network Control Program (NCP), or links.

This section lists possible transmission errors, error messages, probable causes, actions to take, and data to collect to troubleshoot an error.

Condition: Error During Process Execution Initiation

Error	Cause	Action	Collect
SVTM041I (SNASYNC1: Session abnormally terminated)	The session or link was lost before Process execution began.	Review both the short text and long text messages. Use your network management software to determine the status of the link, cross-domain definitions, and applids used in the system-to-system communication. Activate the link, cross-domain definitions, or applids as required, and restart the Process	<ul style="list-style-type: none"> • None
SVTM053I (Session acquire failure)	A protocol error occurred within the IBM Connect:Direct system.	Review both the short text and long text messages. Also, request a session manager trace and an RPL trace.	<ul style="list-style-type: none"> • Output from session manager • Output from RPL traces

Condition: Unrecoverable Error Occurs while a Process Executes

When an unrecoverable send or receive error occurs within the system-to-system session while a Process executes, the three likely causes of the problem are detailed in the following table. You may need a backup copy of the file if a file I/O error caused a send or receive error. If the error is temporary, retrying the Process might clear up the difficulty.

Error	Cause	Action	Collect
SVTM045I	An I/O error within the primary or secondary node causes IBM Connect:Direct to send a negative response to the other location.	<p>Review both the short text and long text messages. Accompanying messages indicate the type of error that caused the send or receive session to fail. Check your network management software for VTAM sense codes, then find the reason the sense code was issued. Correct the problem if possible and retry the Process.</p> <p>In some cases, you need a VTAM buffer or an I/O trace of the error. You may need a backup copy of the file if a file I/O error caused send or receive error. If the error is temporary, retrying the Process might clear up the difficulty.</p>	<ul style="list-style-type: none"> • Output from the VTAM buffer or I/O trace
SVTB020I, followed by a U4095 ABEND	A IBM Connect:Direct system shuts down with either the IMMEDIATE or FORCE parameter specified on the STOP CD command.	Review both the short text and long text messages. Restart IBM Connect:Direct.	<ul style="list-style-type: none"> • None

Error	Cause	Action	Collect
SVTM042I SVTM043I SVTM044I SVTM045I SVTM046I SVTM047I SVTM048I SVTM049I VTAM sense code 0870 VTAM sense code 800A	An error occurs within one of the communication components (VTAM, NCP, or link)	Review both the short text and long text messages. The communication component containing the error issues error messages. Various VTAM and NCP definitions are incompatible with IBM Connect:Direct operations. Refer to <i>Selecting RU Sizes</i> in the <i>IBM Connect:Direct for z/OS Configuration Guide</i> for more information.	<ul style="list-style-type: none"> • VTAM definitions • NCP definitions

Operator Interface Errors

Operator interface errors occur while you are using the operator interface to issue commands to IBM Connect:Direct. You can find more information on the operator interface in the *IBM Connect:Direct for z/OS Facilities Guide*.

This section lists possible operator interface errors, error messages, possible causes, actions to take, and data to collect to troubleshoot an error.

Task Busy Message

Use the following table to troubleshoot the problem when you receive a Task Busy message after issuing an Operator Interface command.

Error	Cause	Action	Collect
IEE342I Modify rejected - task busy	An error in the MCS.SIGNON parameter in the IBM Connect:Direct initialization parameters.	Ensure that the MCS.SIGNON parameter reflects a valid userid-password combination with IBM Connect:Direct operator authority and that the network map is correctly specified on the MCS.SIGNON parameter. Remove comments from this parameter.	<ul style="list-style-type: none"> • IBM Connect:Direct initialization parameters • SYSLOG output

User Not Authorized Messages

Use the following table to troubleshoot authorization errors you receive when you issue operator interface commands.

Error Messages					
SCBB001I	SCBC030I	SCBD001I	SCBE001I	SCBF001I	SCBF063I
SCBF064I	SCBG001I	SCBH001I	SCBI001I	SCBJ001I	SCBK005I
SCBL001I	SCBN001I	SCBO001I	SCBP001I	SCBR002I	SCBS001I
SCBT005I	SCBU003I	SCBV001I	SCBW001I	SCBX001I	SCBY001I
SCPA008I	SFIA002I	SFIA003I	SRJA014I	SRTA008I	SSUB100I

Cause	Action	Collect
The userid attempting to issue the operator interface commands is not authorized to issue them.	<p>Review both the short text and long text messages. Check the userid specified in the MCS.SIGNON parameter of the IBM Connect:Direct initialization parameters file. Determine whether that userid has the authority to issue the command. If you believe it does, run a security trace to determine why the user cannot issue the command.</p> <p>See Security Traces for information on how to run the trace.</p>	<ul style="list-style-type: none"> • IBM Connect:Direct initialization parameters • Security trace

Diagnostic Tools

You can perform some problem isolation and diagnostics by running traces. A trace is a sequential recording of program events during execution. Trace output is useful only as a diagnostic tool for the Support.

When running traces for Support, consider the following information:

- Limit IBM Connect:Direct activity while running traces.
- Trace the simplest case possible.
- Disable traces upon completion. They generate considerable overhead.
- Use traces judiciously. They are for diagnostic purposes only and there is potential security exposure.

Security Traces

Support uses a security trace to debug security problems. The trace shows:

- Fields from the security control block
- Messages
- Return codes from the security system itself such as IBM RACF, CA-TOP SECRET, and ACF2
- Data set names, if verifying data sets
- Userids (passwords will not displayed as encrypted or plain text)

Note: To prevent a remote node's security from discovering and then using Signon dummy passwords to gain access to a primary node, you can use the initialization parameter, "[REMOTE.DUMMY.PASSWORD = \[YES | INTERNAL \]](#)" on page 473.

Starting a Security Trace for the First Time in the DTF

About this task

Note: Once you have completed this procedure, go to [Turning a Security Trace On and Off in the DTF After Initial Setup](#) when you need to start a security trace. You do not have to reinitialize IBM Connect:Direct.

Procedure

1. Specify TRACE=DEBUG as a parameter in your security exit source and reassemble and link-edit the exit.
2. For Connect:Direct for z/OS in the IBM Connect:Direct startup JCL, allocate a DD for SECURITY, either to SYSOUT or to a data set on DASD with attributes of RECFM=VBA, LRECL=121, and BLKSIZE=125 or greater.

To be able to turn security tracing on and off using DEBUG settings, add the DEBUG= parameter to the global initialization parameter data set, and specify 00100000 (or xx1xxxxx where x is any hex value 0-F).

Note: If you specify Separate trace per task tracing to be done for security traces, all output for the security trace will be routed to the task DD, for example, Rnnnnnnnn.

To turn security tracing off, you can then specify xx0xxxxx using the IBM Connect:Direct MODIFY command to specify the debug setting.

3. If IBM Connect:Direct is active, stop IBM Connect:Direct. Restart it with the modified JCL startup job, and recreate the problem.

Turning a Security Trace On and Off in the DTF After Initial Setup

Use the MODIFY command to turn tracing on and off by specifying DEBUG settings (see [Debug Settings](#)).

Starting a Security Trace in the IUI (ISPF)

You can start a security trace to debug security problems in the IUI.

Procedure

1. To allocate a security trace for the IUI, enter the following command:

```
TSO ALLOC FI(APISECUR) DA(*) /* FOR ZOS */
```

Note: As an alternative, to send the information to a preallocated data set with attributes of RECFM=VBA, LRECL=121, and BLKSIZE=125, you can enter the follow command:

```
TSO ALLOC FI(APISECUR) DA(' $h1q.APISECUR.TRACE ')OLD
```

2. Log on to the DTF and carry out any commands you want traced. Trace information is written to the screen immediately upon signon. Sign off when finished.
3. To stop the security trace, issue the following command:

```
TSO FREE FI(APISECUR) /* FOR ZOS */
```

IBM Connect:Direct Function Traces

IBM Connect:Direct contains various internal traces for diagnosing problems and recording events. Based on the trace specified, the IBM Connect:Direct trace output is directed to various ddnames. You can enable these traces using one of the following methods:

- Modify the IBM Connect:Direct startup job stream to include the trace files and ddnames for trace output and add the DEBUG=nnnnnnnn parameter to the initialization parameter data set. The traces are turned on during IBM Connect:Direct initialization and continue running until turned off by the MODIFY command or until IBM Connect:Direct is terminated.
- Issue the IBM Connect:Direct MODIFY command to set DEBUG bits. The trace starts when you issue the MODIFY command. See [IBM Connect:Direct MODIFY Command](#).
- Reduce the amount of trace information by restricting a trace to a Process (MODIFY Debug = parameter) or a specific node (NODE.TRACE.ON parameter). See [IBM Connect:Direct MODIFY Command](#) for more information about these parameters.

Debug Settings

Use the following debug settings with the DEBUG=nnnnnnnn initialization parameter, the BITS.ON=X'nnnnnnnn' and BITS.OFF=X'nnnnnnnn' MODIFY command parameters, and the NODE.TRACE.BITSON=(node name,X'nnnnnnnn') and NODE.TRACE.BITSOFF=(node name,X'nnnnnnnn') MODIFY command parameters.

For each debug bit turned on for a trace, you must allocate the equivalent DD names in the Output DD column to the IBM Connect:Direct started task. If you do not specify these DD names in the started task JCL of IBM Connect:Direct, you must allocate them using either the DYN (batch) or DYN (IUI) option

described in this section. The one exception is separate trace per task (Rnnnnnnn), which is dynamically allocated by IBM Connect:Direct as required. This trace output is directed to SYSOUT.

The following table shows the available function traces for Connect:Direct for z/OS, with their respective DEBUG settings, and the DD names (or filenames) used for output. Specify these bits using hexadecimal notation, for example, X'80' plus X'10' would result in X'90' while X'08' plus X'04' would result in X'0C'.

DEBUG Setting	Trace Type	Output DD
80000000	COPY Routine and RUN TASK trace	RADBDD01
10000000	Full TPCB/SYMBOLICS from DMCBSUBM	DMCBSUBM
08000000	Session manager trace	RADBDD05
04000000	Separate trace per task (Example: "R0000005" to trace TASK 5)	Rnnnnnnn
02000000	API session trace	RADBDD07
01000000	DMGCBSub trace	RADBDD08
00400000	TCQSH from DMCOPYRT	DMCOPYRT
00200000	Make each SVC dump unique	N/A
00100000	SECURITY trace control	SECURITY
00040000	GETMAIN/FREEMAIN trace	RADBDD16
00008000	I/O buffer trace	RADBDD21
00004000	WTO all dynamic allocation parameters	RADBDD22
00002000	IBM Connect:Direct/Plex traces	
	ACTION queue manager trace	CDPLXACT
	CKPT queue manager trace	CDPLXCKP
	TCQ queue manager trace	CDPLXTCQ
	STATS queue manager trace	CDPLXSTA
	First REQUEST queue manager trace	CDPLXREQ
	Second and subsequent REQUEST queue manager trace. For example, "CDPLXR03" traces the third queue manager.	CDPLXRnn
	JOIN queue manager trace	CDPLXJOI
00001000	Workload Balancing trace	CDPLXWLB
00000800	zIIP-related trace	CDZIIP
00000100	In-storage tracing only Note: The size of this in-storage table is controlled by the TRACE.BUFFER initialization parameter.	N/A
00000080	RPL trace - long Note: To avoid generating excessive output when you use this trace with a large value for the V2.BUFSIZE initialization parameter, use the short RPL trace.	RPLOUT
00000040	RPL trace - short	RPLOUT
00000020	Version 2 session trace	RADBDD33

DEBUG Setting	Trace Type	Output DD
00000008	Logon exit trace	RADBDD35
00000004	Logon Processor trace	RADBDD36
00000002	SCIP exit trace	RADBDD37
00000001	SNMP trap trace	SCTRAPDD

Displaying DEBUG Settings

Use the INQUIRE DEBUG command to display the current DEBUG settings for each adjacent node in the Netmap.

The INQUIRE DEBUG command has the following format and parameter.

Label	Command	Parameter
(optional)	INQUIRE DEBUG	WHERE (SERVER=server name)

The parameter for the INQUIRE DEBUG command is:

Parameter	Description
WHERE (SERVER=server name)	This parameter is optional. This parameter specifies the IBM Connect:Direct/Server whose DEBUG settings you want to view. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter. If you omit this parameter in a IBM Connect:Direct/Plex environment, the DEBUG settings for the IBM Connect:Direct/Manager are displayed. You cannot use this parameter in a IBM Connect:Direct/Stand-alone Server. See Debug Settings for a descriptions of these settings.

Using the INQUIRE DEBUG Command from the Batch Interface

About this task

To use the INQUIRE DEBUG command from the batch interface:

Procedure

1. Place your command in a batch job stream.
2. Submit the job while IBM Connect:Direct is running.

Note: You must set the fifth character of the DGADBATC output parameter specification to Y to print the result of the command that is in the temporary data set.

3. Verify the results.

The following figure shows a partial sample report:

```

=====
CD.CHI          *INQ  DEBUG/QUIESCE*  DATE: mm.dd.yyyy  TIME: hh:mm:ss
SYSTEM INITIALIZED ----- (0000)----- mm.dd.yyyy  hh:mm:ss
=====

DEBUG           => '00200000'
QUIESCE         => No
TCQ DSN         => PROD1.CD.CHI.TCQ
TCX DSN         => PROD1.CD.CHI.TCX
TCQ Threshold   => No
TCQ File        0% Full. Max.# CI:      1000 # Used CI:      0

NODE TABLE    => NODE NAME      QUIESCE  DEBUG
NODE ENTRY     1 => CD.DAL        Yes      B0BB0BB0
NODE ENTRY     2 => CD.DAL.CSGA   Yes
NODE ENTRY     3 => CD.DAL.CSGB   Yes
NODE ENTRY     4 => CD.DAL.LU0    Yes
NODE ENTRY     5 => CD.DAL.LU62   Yes
NODE ENTRY     6 => CD.DAL.UDT    Yes

```

Issuing the **INQUIRE DEBUG** Command through the IUI

To display the DEBUG settings from the IUI:

1. Select option INQ from the Connect:Direct Administrative Options Menu.

The Inquire DTF Internal Status screen is displayed.

2. If you want to view the DEBUG settings for a specific IBM Connect:Direct/Plex member, type the server name in the Server field. If you want to view the DEBUG settings for a IBM Connect:Direct/Manager, leave the Server field blank.

Leave the Server field blank in a IBM Connect:Direct/Stand-alone Server.

3. Select the IDBG option.

4. Press ENTER.

The current DEBUG settings are displayed.

DEBUG Initialization Parameter

Various settings on the DEBUG=nnnnnnnn initialization parameter turn on a specific trace option or any combination of options. In the syntax for the DEBUG initialization parameter, nnnnnnnn represents the DEBUG setting in hexadecimal.

You can place trace DD statements in the system without slowing down IBM Connect:Direct performance if you do not turn on the trace by specifying the DEBUG parameter.

For problems with SNA connections, use the following four function traces merged into a single output file:

- Session manager trace
- Separate trace per task trace
- Long RPL trace
- COPY routine trace

Note: If the problem occurs during file transfer or session establishment of node connections, run the trace on both the sending and receiving nodes.

IBM Connect:Direct MODIFY Command

The IBM Connect:Direct MODIFY command yields the same types of traces as the DEBUG initialization parameter. Unlike the initialization parameter, however, the MODIFY command does not require you to bring down and restart the DTF. The trace starts when you issue the MODIFY command to turn on the trace bits, provided the DD is allocated.

Note: When JES data sets are dynamically allocated, specify FREE=CLOSE to ensure that the DD is deallocated and tracing stopped when you close the trace DD using the MODIFY command.

You can issue the MODIFY command through the batch interface or interactively through the IUI. See [Issuing the MODIFY Command Through the Batch Interface](#) or [Issuing the MODIFY Command through the IUI](#). To see the current DEBUG setting, see [Displaying DEBUG Settings](#).

The MODIFY command has the following format and parameters. None of these parameters are required.

Label	Command	Parameters
(optional)	MODIFY	BITS.OFF = X'nnnnnnnn'
		BITS.ON = X'nnnnnnnn'
		CLOSE = ddname
		DDNAME = (ddname, nn)
		DEBUG = nnnnnnnn
		DYN (batch) = 'dynamic allocation string' Note: Use an equal sign before and quotes around the dynamic allocation string.
		DYN (IUI) dynamic allocation string Note: Do not use an equal sign before or quotes around the dynamic allocation string in the IUI screen unless you are entering it on the Command Line Interface..
		INITPARMS = YES
		MODDIR.TRACE = YES
		NODE.TRACE.BITSOFF=(node name, X'nnnnnnnn')
		NODE.TRACE.BITSON=(node name, X'nnnnnnnn')
		NODE.TRACE.ON = (node name, debug bits)
		NODE.TRACE.OFF = node name
		WHERE(SERVER = server name)
		ZIIP = NONE EXTCOMP SSLTLS ALL PROJECT

The following table describes the parameters of the MODIFY command.

Parameter	Description
BITS.OFF = X'nnnnnnnn'	Turns individual trace bits off. Refer to Debug Settings for the 'nnnnnnnn' value.
BITS.ON = X'nnnnnnnn'	Turns individual trace bits on. Refer to Debug Settings for the 'nnnnnnnn' value.
CLOSE = ddname	This parameter specifies the DD name that is closed in the IBM Connect:Direct DTF.
DDNAME = (ddname, nn)	This parameter specifies a DD name related to a requested trace. All trace information generated as a result of the BITS.ON setting is directed to the DDNAME indicated in the parameter list, based on the IBM Connect:Direct TASKID number nn. This DDNAME provides a consolidated trace of all activity associated with the task. The DDNAME format is R0000nn, where nn is the TASKID.
DEBUG = nnnnnnnn	Replaces the system-wide debugging bits with the specified debug bits. Refer to Debug Settings to see all possible debug bit values.

Parameter	Description
DYN = 'dynamic allocation string' (batch)	<p>This parameter specifies that dynamic allocation is invoked in the DTF using a specified allocation string. Use an equal sign before and quotes around the dynamic allocation string in the batch DYN parameter.</p> <p>You must use an equal sign before and quotes around the dynamic allocation string in the Command Line Interface for the IUI.</p>
DYN dynamic allocation string (IUI)	<p>This parameter specifies that dynamic allocation is invoked in the DTF using a specified allocation string. Do not use an equal sign before or quotes around the dynamic allocation string in the MODIFY DYN field on the MODIFY COMMAND screen.</p> <p>Note: If you issue the MODIFY command in the Command Line Interface (CLI), you must use an equal sign before and quotes around the dynamic allocation string.</p>
INITPARMS	Refreshes initialization parameters qualified to be modified dynamically while IBM Connect:Direct is running.
MODDIR.TRACE=YES	Requests a module trace.
NODE.TRACE.BITSOFF = (node name, X'nnnnnnnn')	Turns individual trace bits off for a specified node. The node name is the 1–16 character name of the node on which the trace runs. The * and ? wildcard characters are supported when specifying the node name. Refer to Debug Settings for the 'nnnnnnnn' value.
NODE.TRACE.BITSON = (node name, X'nnnnnnnn')	Turns individual trace bits on for a specified node. The node name is the 1–16 character name of the node on which the trace runs. The * and ? wildcard characters are supported when specifying the node name. Refer to Debug Settings for the 'nnnnnnnn' value.
NODE.TRACE.ON = (node name, debug bits)	<p>Requests a trace run on one or more specified nodes. The node name is the 1–16 character name of the node on which the trace runs. The * and ? wildcard characters are supported when specifying the node name. The debug bits are the 8-character DEBUG bits setting. See the BITS.OFF = X'nnnnnnnn' parameter for a listing of debug bits.</p> <p>The trace runs until turned off by the NODE.TRACE.OFF= parameter.</p>
NODE.TRACE.OFF = (node name)	<p>Turns off a trace set by the NODE.TRACE.ON parameter.</p> <p>The node name is the 1–16 character name of the node on which the trace is running. The * and ? wildcard characters are supported when specifying the node name.</p>
WHERE(SERVER=server name)	<p>This parameter specifies which IBM Connect:Direct/Plex member the MODIFY command applies to. The server name parameter is the 1–8 character name assigned to a IBM Connect:Direct/Server by the CDPLEX.SERVER initialization parameter.</p> <p>If this parameter is omitted, the MODIFY command applies to the IBM Connect:Direct/Manager.</p>
ZIIP = <u>NONE</u> EXTCOMP SSLTLS ALL PROJECT	This parameter specifies the setting for the zIIP exploitation feature (ZEF). Refer to “zIIP Exploitation Feature” on page 404 and “ZIIP = NONE EXTCOMP SSLTLS ALL PROJECT” on page 496 .

Issuing the MODIFY Command Through the Batch Interface

To use the MODIFY command from the batch interface:

Procedure

1. Place the command in the batch job stream.
2. Submit the job while IBM Connect:Direct is running.
3. Verify the results.

Examples of MODIFY Command through the Batch Interface

The following example turns on the short RPLOUT trace:

```
MODIFY BITS.ON = X'00000040'
```

The following example turns off the short RPLOUT trace:

```
MODIFY BITS.OFF = X'00000040'
```

The following example invokes dynamic allocation in the DTF to the allocated DDNAME RPLERRCK:

```
MODIFY DYN = 'DD=RPLERRCK'
```

The following job turns on the merged COPY routine and DMGCBSub traces.

```
//TRACEON JOB (1111),'TRACES',NOTIFY=JSMITH,CLASS=0,  
//          REGION=1024K,MSGCLASS=X  
//STEP01   EXEC PGM=DGADBATC,PARM='YYSLYYY'  
//STEPLIB  DD DSN=$CD.SDGALINK,DISP=SHR  
//DMPUBLIB DD DISP=SHR,DSN=JSMITH.CNTL  
//          DD DISP=SHR,DSN=JSMITH.PROCESS.LIB  
//DMMSGFIL DD DISP=SHR,DSN=JSMITH.CD1.MSG  
//DMNETMAP DD DISP=SHR,DSN=JSMITH.CD1.NETMAP  
//DMPRINT  DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//NDMLLOG  DD SYSOUT=*  
//SYSIN    DD *  
SIGNON NETMAP=JSMITH.CD1.NETMAP USERID=(JSMITH)  
MODIFY BITS.ON=X'81000000'  
SUBMIT PROC=ACCTSEPT  
SIGNOFF
```

The following job turns off the trace.

```
//NOTRACE JOB (1111),'TRACES',NOTIFY=JSMITH,CLASS=0,  
//          REGION=1024K,MSGCLASS=X  
//STEP01   EXEC PGM=DGADBATC,PARM='YYSLYYY'  
//STEPLIB  DD DSN=$CD.SDGALINK,DISP=SHR  
//DMPUBLIB DD DISP=SHR,DSN=JSMITH.CNTL  
//          DD DISP=SHR,DSN=JSMITH.PROCESS.LIB  
//DMMSGFIL DD DISP=SHR,DSN=JSMITH.CD1.MSG  
//DMNETMAP DD DISP=SHR,DSN=JSMITH.CD1.NETMAP  
//DMPRINT  DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//NDMLLOG  DD SYSOUT=*  
//SYSIN    DD *  
SIGNON NETMAP=JSMITH.CD1.NETMAP USERID=(JSMITH)  
MODIFY BITS.OFF=X'81000000'  
SIGNOFF
```

See *IBM Connect:Direct for z/OS User Guide* for a description of the batch interface, and the installation and administration guide for a description of the MODIFY command.

Issuing the MODIFY Command through the IUI

About this task

To use the MODIFY command features through the IUI:

Procedure

1. Request option MD from the Connect:Direct Administrative Options Menu to access the MODIFY COMMAND screen.

node.name		MODIFY COMMAND		14:34	
CMD ==>					
Server ==> _____ 00000000 (Current DEBUG Settings)					
MODIFY	DEBUG	==>	_____	(nnnnnnnn)	
MODIFY	BITS.ON	==>	_____	(nnnnnnnn)	
MODIFY	BITS.OFF	==>	_____	(nnnnnnnn)	
MODIFY	DDNAME	==>	_____	(ddname,nn)	
MODIFY	CLOSE	==>	_____	(ddname)	
MODIFY	MODDIR.TRACE	==>	_____	(YES)	
MODIFY	DYN	==>	_____		
MODIFY	SESSIONS	==>	(Quiesce or Resume)	NODE	==> _____
MODIFY	NODE.TRACE.ON	==>	(_____)		
MODIFY	NODE.TRACE.OFF	==>	_____		
MODIFY	NODES	==>	_____	(YES)	
MODIFY	INITPARMS	==>	_____	(YES)	
MODIFY	ZIIP	==>	_____	(ALL NONE EXTCOMP SSLTLS PROJECT)	

2. Type values in the appropriate fields. See the MODIFY command parameter descriptions in [IBM Connect:Direct MODIFY Command](#).

Note: Do not type an equal sign or quotes in the MODIFY DYN field. However, if you issue the MODIFY command on the CLI, you must type an equal sign before and quotation marks around the dynamic allocation string.

3. When you have completed your entries, press ENTER.

Note: If you need to modify trace settings for more than one node, you can use the MODIFY NODES screen which displays the Trace and Debug settings for all of the Netmap adjacent nodes. See [Modify Trace and Debug Settings Through the Modify Nodes Screen](#) for more information.

Examples of MODIFY Commands through the IUI

This section shows examples of MODIFY commands for Connect:Direct for z/OS.

The following MODIFY commands set the bits to turn on a short send/receive trace and to dynamically allocate the ddname RPLOUT.

```
MODIFY DYN='DD=RPLOUT,DSN=A985467.PRINT,DISP=SHR,FREE=CLOSE'
MODIFY BITS.ON=X'00000040'
```

After running the trace, the following MODIFY commands close the ddname RPLOUT and turn off the short send/receive trace.

```
MODIFY BITS.OFF=X'00000040'
MODIFY CLOSE=RPLOUT
```

Modify Trace and Debug Settings Through the Modify Nodes Screen

Perform the following procedure to modify trace and debug settings through the Modify Nodes screen.

Procedure

1. Request option MD from the Connect:Direct Administrative Options Menu to access the Modify Command screen.
2. Type YES in the MODIFY NODES field.
3. Press **ENTER**.

The Modify Nodes screen is displayed.

MODIFY NODES (Unsaved Changes)				Row 1 of 1054
CMD ==>				SCROLL ==> PAGE
(CMD commands: CLEAR DEBUG/TRACE QUIESCE RESUME; CANCEL END REFRESH SAVE)				
(SEL commands: C=Clear Q=Quiesce R=Resume. Overtyping Debug setting.)				
SEL	Node Name	Quiesce	Debug	Chg
---	-----	-----	-----	---
	CD.PR.D.CSGA	OFF	11111111	
	CD.PR.D.CSGB	OFF	11111111	
	CD.PR.D.CSGD	OFF	11111111	
	CD.PR.D.CSGE	OFF	11111111	
	CD.PR.D.CSGF	ON	11111111	Y
	CD.PR.D.CSGG	OFF	11111111	
	CD.PR.D.VIPA	OFF	11111111	
	CD.TST	OFF	FFFFFFFF	
	CD.TST. %%%%	OFF	FFFFFFFF	
	CD.TST.CSGA	ON		Y
	CD.TST.CSGB	OFF	FFFFFFFF	
	CD.TST.CSGD	OFF	FFFFFFFF	
	CD.TST.CSGG	OFF	FFFFFFFF	
	CD.TST.CTCA1	OFF	FFFFFFFF	
	CD.TST.CTCA2	OFF	FFFFFFFF	
	CD.TST.CTCA3	OFF	FFFFFFFF	

4. Use one or both of the following methods:
 - Issue a Debug or Trace command on the CMD line.
 - Issue SEL commands or overtype the Debug field for specific nodes.

For more information, see [“Modify Nodes Screen” on page 179](#).

5. Type SAVE in the CMD field and press **ENTER** to save your changes.

IBM Connect:Direct Automatic Traces

You can activate some internal IBM Connect:Direct traces without using the DEBUG bits. Activate these traces during normal IBM Connect:Direct operation. Others traces are useful mostly for problem diagnosis. You can activate each of these automatic IBM Connect:Direct traces by having a DD statement in the IBM Connect:Direct startup JCL or allocated in a TSO session, except for NDMCMDS, which is allocated in the JCL for DGADBATC or through the IUI. For more information, see [DD Statements in Startup JCL](#).

Note: To avoid generating excessive output, review [Security Traces](#).

Some of the most useful automatic traces are:

- DMVSOPEN contains information related to the allocation of the target data set.
- ESTAE captures information on I/O errors; VTAM connection errors; ABEND control blocks; open and close errors; TCQ/TCX errors on add, update, and so forth; and statistics file write errors. Provided in the basic DGAJCONN JCL member.
- CDESTAE contains various I/O errors from the statistics facility.
- RPLERRCK captures VTAM send and receive errors. Provided in the basic DGAJCONN JCL member.

- NDMCMDS shows all actual IBM Connect:Direct commands issued to the DTF, including resolution of symbolics. It can be particularly helpful to debug IBM Connect:Direct commands through the IUI or through batch interface if you are having signon problems, syntax errors, and so forth.
- NDMLOG lists all initialization parameters read from the INITPARM data set including obsolete parameters, which are indicated by SITA995I messages, and all modules, along with the last date on which they were modified, and related fix numbers. Provided in the basic DGAJCONN JCL member.
- CDLOG is a chronological log of IBM Connect:Direct events listing all master, console, programmer, and operator messages, and information on failed tests.

DD Statements in Startup JCL

IBM Connect:Direct provides two members in the JCL file:

- DGAJCONN—Contains the minimum set of DDs to run IBM Connect:Direct. You may want to use this set of DD statements as the base and add specific DD statements from the DGAJCONX member to fit your tracing needs.
- DGAJCONX—Contains all possible DDs to cover stand-alone servers, IBM Connect:Direct/Plex systems, automatic traces, and various DEBUG output. Most of the DD statements in this JCL startup jobstream have been discussed earlier in this chapter.

The DD statements in the DGAJCONX JCL can be broken down into the following types:

- Minimal DD statements included in the DGAJCONN JCL
- DD statements for general operations
- DD statements for running automatic traces.
- DD statements for running detailed function traces in conjunction with the DEBUG initialization parameter in a stand-alone IBM Connect:Direct system
- DD statements for running detailed function traces in conjunction with the DEBUG initialization parameter in a IBM Connect:Direct/Plex system



CAUTION: The diagnostic DD statements associated with the DEBUG bits cause much more I/O and allocating them can consume significant CPU resources. To avoid degrading performance in your production environment, be sure to follow the guidelines in [Diagnostic Tools](#).

The following DD statements are included in both the DGAJCONN and DGAJCONX JCL:

DDNAME	Function
STEPLIB	IBM Connect:Direct SDGALINK.
DMPUBLIB	IBM Connect:Direct Process library
USRINFO	Standard display from User exits
NDMLOG	Automatic trace to list all initialization parameters read from the INITPARM data set including obsolete parameters, which are indicated by SITA995I messages, and all modules, along with the last date on which they were modified, and related fix numbers.
ESTAE	Automatic trace to capture I/O errors, VTAM connection errors, ABEND control blocks, open and close errors, TCQ/TCX errors on adds and updates, and statistics file write errors.
RPLERRCK	Automatic trace to capture VTAM/TCP send and receive errors.

The following DD statements run general functions in all systems:

DDNAME	Function
SYSTEM	Standard MVS

DDNAME	Function
NSXOUT	SNA NSXEXIT
CDDUMPR ¹	IBM Connect:Direct DUMP command output
ADRIOXLG	Sysprint for DSS exit
SECURITY ¹	Security trace if Stage 2 exit has TRACE=DEBUG specified
CDSECURI	Stage1 Security Trace
DMPRINT ¹	Submit command Output
APITRACE ¹	Stage2 API Trace, which consists primarily of IUI and batch interface activity
AXUNIQ	Error Recovery
¹ Can generate excessive output. Use judiciously.	

The following DD statements are used to run automatic traces and perform related functions.

Note: To handle B37 (out of space) conditions when tracing to disk, code the DD statement with FREE=CLOSE, which will close and deallocate the dataset. The DD statement will not be reallocated and tracing will stop. If the DD statement does not include FREE=CLOSE and a B37 condition occurs, the dataset is closed and reopened. Tracing continues and all old trace data to this DD is lost. When FREE=CLOSE is coded for JES datasets, closing the DD makes the JES entry available immediately for printing or purging without affecting the other JES datasets. You could specify FREE=CLOSE in the DD statement and then, using the ISPF MD dialog, periodically close the DD and reallocate it with FREE=CLOSE. This would enable you to purge old JES entries without having to reinitialize IBM Connect:Direct.

DDNAME	Function
DEVTRACE ¹	Traces UCB open and close activity.
DMGEVENT	Traces Event Services activity.
DMVSOPEN	Formats the allocation block.
CDESTAE	Diagnostics on various I/O errors from the statistics facility. Allocate this name by including in the IBM Connect:Direct startup JCL.
IGWTRACE ¹	Traces PDSE program objects load to unload activity.
LOSTOUT	Lost term exit trace.
NDMAPI	Used for diagnostics on session errors with the API.
NDMCMDS	IUI/Application Program Interface (IUI/API) commands passed to IBM Connect:Direct. Also use it for diagnostics on session errors with the API.
CDCMDS	IUI/Application Program Interface (IUI/API) commands passed to IBM Connect:Direct. Also use it for diagnostics on session errors with the API.
CDLOG ¹	A chronological log of IBM Connect:Direct events listing all master, console, programmer, and operator messages, and information on failed tests.
¹ Can generate excessive output. Use judiciously.	

The following output DD statements are used to run detailed function traces and perform related functions in a stand-alone system.

Output DD	Trace Type	DEBUG Setting
RADBDD01	COPY Routine and RUN TASK trace	80000000

Output DD	Trace Type	DEBUG Setting
DMCBSUBM	Full TPCB/SYMBOLICS from DMCBSUBM	10000000
RADBDD05	Session manager trace	08000000
Rnnnnnnn	Separate trace per task (Example: "R0000005" to trace TASK 5) Note: DDs are dynamically allocated by IBM Connect:Direct as required.	04000000
RADBDD07	API session trace	02000000
RADBDD08	DGADCBSU trace	01000000
DMCOPYRT	TCQSH from DMCOPYRT	00400000
N/A	Make each SVC dump unique	00200000
SECURITY	SECURITY trace control	00100000
RADBDD16	GETMAIN/FREEMAIN trace	00040000
RADBDD21	I/O buffer trace	00008000
RADBDD22	WTO all dynamic allocation parameters	00004000
CDZIIP	System z Integrated Information Processor (zIIP)-related trace	00000800
RPLOUT	RPL trace - long Note: To avoid generating excessive output when you use this trace with a large value for the V2.BUFSIZE initialization parameter, use the short RPL trace.	00000080
RPLOUT	RPL trace - short	00000040
RADBDD33	Version 2 session trace	00000020
RADBDD35	Logon exit trace	00000008
RADBDD36	Logon Process or trace	00000004
RADBDD37	SCIP exit trace	00000002
SCTRAPDD	SNMP trace	00000001

The following output DD statements are used to run detailed function traces and perform related functions in a IBM Connect:Direct/Plex environment:

Output DD	Trace Type	DEBUG Setting
CDPLXACT	ACTION queue manager trace	00002000
CDPLXCKP	CKPT queue manager trace	
CDPLXTCQ	TCQ queue manager trace	
CDPLXSTA	STATS queue manager trace	
CDPLXREQ	First REQUEST queue manager trace	
CDPLXRnn	Second and subsequent REQUEST queue manager trace. For example, "CDPLXR03" traces the third queue manager.	
CDPLXJOI	JOIN queue manager trace	
CDPLXWLB	Workload Balancing trace	00001000

Global Initialization Parameters

Initialization parameters supply values for various IBM Connect:Direct functions. IBM Connect:Direct processes these parameters during initialization.

Global initialization parameters apply to a IBM Connect:Direct/Stand-alone Server or each member of a IBM Connect:Direct/Plex environment.

In a IBM Connect:Direct/Plex environment, the local initialization parameters of a member can override some global initialization parameters. See [Local Initialization Parameters](#) for more information about local initialization parameters.

In all cases, initialization parameters are kept in a PDS (Partitioned Data Set) and may not come from a sequential file.

Note: In a IBM Connect:Direct/Plex environment, you can only override initialization parameters allowed in the local initialization parameters file using the PARM= keyword in the EXEC statement at system startup. In a IBM Connect:Direct/Stand-alone Server environment, however, you can override global initialization parameters with the PARM= keyword in the EXEC statement.

Handling Return Codes

If you receive a return code of 4 when you stop IBM Connect:Direct, review the NDMLOG for information regarding obsolete and soon-to-be obsolete parameters. Once you remove these parameters, you avoid these warning messages at start-up. These messages are numbered: SITA995I, SITA988I, and SITA989I.

ABEND.CODES.NODUMP=(ABEND code list)

The ABEND code list parameter specifies up to 16 system or user ABENDs. This parameter does not apply to RUN TASK ABENDS unless you specify ABEND.CODES.NODUMP for the ABEND.RUNTASK initialization parameter.

When a task ABENDs, IBM Connect:Direct searches this list. If the ABEND is found, the dump is suppressed. Use this parameter to prevent dumps on ABEND codes for which you do not need dump information.

Specify system ABEND codes in the list using an S followed by three characters such as SB37.

Specify user ABEND codes in the list using a U followed by four numeric digits such as U4030.

Specify a wildcard masking character in the code using X. Place the X in any position following either the S or U, such as SX37 or U40XX.

Note: It is not necessary to include the following ABEND codes in the ABEND.CODES.NODUMP list because they are suppressed by default: S00C (reason code 4 only), S047, S13E, S222, S422, SA03, U1024, U1025, U1028, SX13, SX37.

Modifiable through MODIFY INITPARMS command: YES.

ABEND.RUNTASK = (DUMP | ABEND.CODES.NODUMP)

This parameter specifies whether to refer to ABEND.CODES.NODUMP for ABEND codes when suppressing dumps.

Value	Description
DUMP	System dumps when a RUNTASK ABENDs. This value is the default.
ABEND.CODES.NODUMP	An ABEND dump is generated instead of a system dump. Searches the ABEND.CODES.NODUMP list for a match. If the ABEND code is found, the dump is suppressed. If the ABEND code is not found, the system dump continues.

Modifiable through MODIFY INITPARMS command: YES

ALLOC.CODES = (allocation errors)

This parameter specifies allocation errors for which IBM Connect:Direct retries the Process step as specified in the ALLOC.RETRIES and ALLOC.WAIT initialization parameters. These allocation errors are the Dynamic Allocation Interface Routine (DAIR) codes generated by the dynamic allocation function (SVC99) of the operating system, such as z/OS. The following is an example of an IBM error reason code (0210) shown in the IBM Connect:Direct SDAB005I message.

SDAB005I - ERR=0210,INFO=0000,REQUESTED DATA SET NOT AVAILABLE. ALLOCATED TO ANOTHER JOB.

You can find common dynamic allocation error codes in the IBM Connect:Direct MSG file by enclosing the code in the message SDExxxxI skeleton, where xxxx is the error reason code. The variable SDAB005I message short text and description come from the *MVS Programming Authorized Assembler Services Guide* (general publication number SA22-7608-nn), which is accessible on IBM's website. For ERR codes not found in the IBM Connect:Direct MSG file, see the *MVS Programming Authorized Assembler Services Guide*.

IBM Connect:Direct defined the following error codes to enable retries for specific failures.

Value	Description
PDSR	This is an internally generated IBM Connect:Direct code, that unlike the others described in this section, is not an allocation code. When two or more Processes attempt to write to the same PDS at the same time, only one can write successfully to the PDS. The remaining Processes have error messages with this PDSR code and are retried in the same way as the other codes discussed in this section.
DSNR	This option is similar to the PDSR allocation code but works on non-PDS data sets. When you attempt to send or receive a non-PDS data set, but cannot because the data set is in use, IBM Connect:Direct retries the Process (specified by the retry option) for the number of times specified, if you have coded the DSNR option.
GDGR	This option indicates that if the ENQ fails (as described for the GDGENQ parameter) for GDG data sets, the Process is retried.
TAPR	This option indicates that a tape Process is retried if the number of tape Processes specified by MAX.TAPE is reached.

Value	Description
ARCH	<p>This option indicates that the data set is archived, and IBM Connect:Direct retries based upon the value specified for the ALLOC.RETRIES initialization parameter. You must use the DGAXARCL allocation exit for this option to work.</p> <p>Note: It is not necessary to use the IBM Connect:Direct DMGALRCL exit to avoid a common allocation error that ties up the IBM task input/output table (TIOT) via the enqueue on SYSZIOT. See the discussion on the sample allocation exit, DGAXARCL (alias of DMGALRCL), in Sample Allocation Exits, for more detailed information on recall processing of migrated and archived data sets.</p>

Allocation retries are controlled by the PNODE and supported for z/OS and VM/ESA platforms only. The following table lists the default allocation error codes and their meanings.

Code	Definition
020C	Exclusive use of shared file
0210	Allocated to another job
0218	Volume not mounted
0220	Volume not available
0234	One device required
0068	VM minidisk already linked read-only, if transferring with VM
0069	VM minidisk already linked read-write, if transferring with VM
006A	VM minidisk already linked read-write and read-only, if transferring with VM

Modifiable through MODIFY INITPARMS command: YES

ALLOC.MSG.LEVEL = INFO | WARN | SEVERE

This parameter specifies the severity of dynamic allocation messages to display. By setting the level, you can suppress unwanted messages. ALLOC.MSG.LEVEL values correspond to levels defined by Dynamic Allocation and are described in the *IBM Authorized Assembler Services Guide* (GC28-1467).

Value	Description
INFO	This value indicates that all informational messages are displayed.
WARN	This value indicates that only warning messages are displayed.
SEVERE	This value indicates that only severe messages are displayed.

Modifiable through MODIFY INITPARMS command: NO

ALLOC.RETRIES = number of retries

This parameter specifies the number of IBM Connect:Direct retries of an allocation failure. The default is 20.

Modifiable through MODIFY INITPARMS command: YES

ALLOC.WAIT = hh:mm:ss

This parameter specifies the amount of time that IBM Connect:Direct waits between retries of an allocation failure. The default is 00:03:00.

Modifiable through MODIFY INITPARMS command: YES

ALLOCATION.EXIT = modname

This parameter specifies the name of the user-written interface that communicates with Connect:Direct for z/OS. You can invoke the allocation exit prior to any allocation activity by IBM Connect:Direct, thereby allowing the exit program to examine and modify information that IBM Connect:Direct uses during the allocation Process, such as the data set name (DSN) and destination name. In addition, you can set parameters to terminate a copy step before allocation takes place.

For sample exits, see the \$CD.SDGASAMP library. The default is no allocation exit.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX = NO | YES

This parameter indicates whether or not IBM Connect:Direct initializes as a IBM Connect:Direct/Plex operation.

Value	Description
NO	IBM Connect:Direct does not initialize as a IBM Connect:Direct/Plex operation.
YES	IBM Connect:Direct initializes as a IBM Connect:Direct/Plex operation.

Modified through MODIFY INITPARMS command: NO

CDPLEX.MAXSERVER = maximum number of servers | 4

This parameter specifies the maximum number of servers the IBM Connect:Direct/Plex Manager will manage. The acceptable range is 1–32. The default value is 4.

For more information, see *Storage Requirements in a IBM Connect:Direct Plex Environment* in the *IBMConnect:Direct for z/OS Configuration Guide*.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.TIMER = 5 | number of minutes

This parameter specifies the time-out value for XCF communications in minutes. The valid range is 0, 5–99. Zero (0) indicates that no time-out is set for XCF communications.

Modified through MODIFY INITPARMS command: NO

CDPLEX.WLM.GOAL = (NO | YES, exitname)

This parameter specifies whether the IBM Workload Manager (WLM) Goal Mode queries are used for balancing IBM Connect:Direct/Plex Process workload in a sysplex.

Value	Description
NO	No Goal Mode queries are made. Process workload is balanced without determining system capacity information. This is the default value.
YES	Goal mode queries are made. A warning message is issued when a WLM query is made that shows a server is running on a system that is NOT in Goal Mode.
exitname	If you want to implement your own method for using WLM query information, you can create a WLM user exit and specify the name of the exit. You can find a sample user exit in \$CD.SDGASAMP library called DGAXWLMX .

Modifiable through MODIFY INITPARMS command: yes

CHECK.CERT.EXPIRE = NO | YES

This parameter indicates whether or not IBM Connect:Direct will check the validity of certificates.

This parameter is only effective if Connect:Direct Secure Plus has been enabled by specifying SECURE.DSN = Connect:Direct Secure Plus parameter filename. For more information, see [Certificate Validity Check Used by Connect:Direct Secure Plus](#).

Value	Description
NO	IBM Connect:Direct does not check the validity of certificates.
YES	IBM Connect:Direct checks the validity of certificates whenever IBM Connect:Direct is started. This is the default value.

Modifiable through MODIFY INITPARMS command: YES

CHECK.CERT.EXPIRE.TIME = 00:00:00 | HH:MM:SS

This parameter specifies the time of day that IBM Connect:Direct will check the validity of certificates. The default, 00:00:00, indicates that IBM Connect:Direct will perform its validity check at midnight. You can enter any value between 00:00:00 and 23:59:59. This parameter is only effective if CHECK.CERT.EXPIRE = YES and Connect:Direct Secure Plus has been enabled by specifying SECURE.DSN = Connect:Direct Secure Plus parameter filename.

Modifiable through MODIFY INITPARMS command: YES

CHECK.CERT.EXPIRE.WARN.DAYS = 30 | nnn

This parameter specifies the number of days prior to the certificate expiration date that a warning message will be issued. IBM Connect:Direct. The default is 30 days. You can enter any value between 1 and 365. This parameter is only effective if CHECK.CERT.EXPIRE = YES and Connect:Direct Secure Plus has been enabled by specifying SECURE.DSN = Connect:Direct Secure Plus parameter filename.

Modifiable through MODIFY INITPARMS command: YES

CKPT = nK | nM

This parameter enables automatic checkpointing of eligible files if no CKPT keyword is specified on the IBM Connect:Direct COPY statement.

See the CKPT.MODE initialization parameter for further details of automatic checkpointing. K means thousands of bytes; M means millions of bytes.

Valid values of n are:

- 0–2147483K
- 0–2147M

Setting a value of 0 means no automatic checkpointing.

The default is 10M.

IBM Connect:Direct uses the value specified, rounded to the nearest block boundary, to determine when a checkpoint is taken. The CKPT specification on the IBM Connect:Direct Copy statement always overrides the CKPT initialization parameter value.

Modifiable through MODIFY INITPARMS command: YES

Note: For sequential files, do not specify a CKPT value less than:

$$\text{BLKSIZE} * \text{NCP} * 10 * \# \text{ stripes}$$

where NCP is the number of buffers for reading data from or writing data to a sequential data set using BSAM and # stripes refers to striped extended-format data sets. For more information, see [Problems Involving Checkpoints](#) .

CKPT.DAYS = number of days

This parameter specifies the number of days that checkpoint records stay in the Checkpoint file before automatic deletion during IBM Connect:Direct initialization. The records can be left in the Checkpoint file if transmission is interrupted and the Process is deleted without being restarted. The default is 4.

Modifiable through MODIFY INITPARMS command: YES

CKPT.MODE = (RECORD | BLOCK BLOCK | RECORD PDS | NOPDS NOPDS | PDS VSAM | NOVSAM VSAM | NOVSAM)

This parameter enables you to control checkpointing in both record and block level transfers.

Note: This parameter does not apply to TCP/IP or LU6.2 connections. See Process Queuing and Recovery for an explanation of when checkpointing occurs.

Subparameter	Description
<u>RECORD</u> BLOCK	Refers to transferring physical sequential (PS) files. It determines whether checkpointing is allowed when the CKPT parameter is specified on the COPY statement. If a record level transfer is taking place, BLOCK does not allow record level checkpointing, even if you have coded the request on the COPY statement. If you are performing a block level transfer, BLOCK enables checkpointing requests coded on the COPY statement. RECORD enables record level checkpointing.
<u>BLOCK</u> RECORD	Refers to transferring PS files. It determines what type of checkpointing occurs when automatic checkpointing is in effect. You enable automatic checkpointing by specifying a value in the CKPT parameter in the initialization parameters. If you use the CKPT parameter, you do not have to request checkpointing on each COPY statement. If you specify RECORD for this parameter, both record level and block level automatic checkpointing occur, depending on the mode of transfer for each copy. Use BLOCK to prevent automatic checkpointing on a record level transfer.
<u>PDS</u> NOPDS	Refers to transferring partition data sets (PDS). Because checkpointing information is sent with each member of a PDS, this subparameter specifies whether checkpointing is allowed with PDS transmission if a request is coded on the COPY statement. NOPDS prevents checkpointing on the PDS, even if you requested it on the Copy statement. PDS enables PDS transmissions to be checkpointed.
<u>NOPDS</u> PDS	Refers to transferring PDS. It determines what type of checkpointing occurs when automatic checkpointing is in effect. To enable automatic checkpointing, specify a value in the CKPT parameter in the initialization parameters. If you use the CKPT parameter, you do not have to request checkpointing on each COPY statement. If you specify PDS, all PDS transmissions are automatically checkpointed. NOPDS prevents automatic checkpointing of PDS transmission.
<u>VSAM</u> NOVSAM	Specifies whether checkpointing takes place for VSAM files when the checkpoint parameter is specified in the COPY statement.
<u>VSAM</u> NOVSAM	Specifies whether automatic checkpointing takes place for VSAM files.

Modifiable through MODIFY INITPARMS command: YES

COMPRESS.EXT = ALLOW | DISALLOW

This parameter specifies whether extended compression will be allowed on a global basis.

Note: If COMPRESS.NETMAP.OVERRIDE=ALLOW is specified, the COMPRESS.EXT value specified in the netmap ADJACENT.NODE entry will override the value specified for this parameter in the global initialization parameters.

Value	Description
ALLOW	Extended compression is allowed. This value is the default.
DISALLOW	Extended compression is not allowed.

Modifiable through MODIFY INITPARMS command: Yes.

COMPRESS.NEGO.FAIL = STEP | PROCESS

This parameter specifies how a Process is handled when negotiations fail (for the system that has control of the Process).

Value	Description
STEP	When negotiations fail, the STEP fails with RC=8, and modal logic may be used to determine how to proceed. This value is the default.
PROCESS	When negotiations fail, the Process fails and is placed in HO HE (Held in Error).

Modifiable through MODIFY INITPARMS command: Yes.

COMPRESS.NETMAP.OVERRIDE = ALLOW | DISALLOW

This parameter specifies whether netmap entries for compression control (on a node-by-node basis) will override the global compression control settings in the initialization parameters.

Value	Description
ALLOW	The netmap compression control settings are allowed to override the initialization parameters. This value is the default.
DISALLOW	The netmap compression control settings are not allowed to override the initialization parameters.

Modifiable through MODIFY INITPARMS command: Yes.

COMPRESS.STD = ALLOW | DISALLOW

This parameter specifies whether standard compression will be allowed on a global basis.

Note: If COMPRESS.NETMAP.OVERRIDE=ALLOW is specified, the COMPRESS.STD value specified in the netmap ADJACENT.NODE entry will override the value specified for this parameter in the global initialization parameters.

Value	Description
ALLOW	Standard compression is allowed. This value is the default.
DISALLOW	Standard compression is not allowed.

Modifiable through MODIFY INITPARMS command: Yes.

CONFIRM.COLD.START = YES | NO

This parameter indicates whether an operator has to confirm a COLD start of the Transmission Control Queue (TCQ) and Statistics files.

Value	Description
YES	Issues a Write to Operator with Reply (WTOR) prompt to force the operator to confirm the request for a COLD start before executing the COLD start.
NO	Performs a COLD start without requiring operator confirmation.

Modifiable through MODIFY INITPARMS command: NO

CRC = (OFF | ON, YES, No)

This parameter specifies whether cyclic redundancy checking (CRC) is performed for TCP/IP connections. The CRC parameter is a 32 bit checksum used by IBM Connect:Direct to detect network data corruption which occasionally occurs in IP data networks. If network data corruption is detected, IBM Connect:Direct will stop the Process execution and restart the Process from the last checkpoint record. This provides an extra layer of data integrity when transmitting data over an IP network. You cannot enable CRC checking when running Connect:Direct Secure Plus.

Parameter	Description
<u>OFF</u> ON	This parameter specifies whether CRC is ON or OFF for the node. This first positional parameter establishes the default for the node.
<u>YES</u> No	This parameter specifies whether overrides are allowed. You can perform overrides in a Process statement or with a network map parameter.

Modifiable through MODIFY INITPARMS command: YES

CTCA = NO | YES

This parameter specifies whether the channel-to-channel adapter (CTCA) driver is loaded at IBM Connect:Direct initialization.

Value	Description
NO	The CTCA driver is not loaded at initialization. This value is the default.
YES	The CTCA driver is loaded at IBM Connect:Direct initialization.

Modifiable through MODIFY INITPARMS command: NO

CTCA.TIMER = number of seconds

This parameter specifies the number of seconds for a SEND/RECEIVE to wait before a timeout on the connection occurs. The valid range is from 60–300. The default is 180 seconds.

Modifiable through MODIFY INITPARMS command: NO

DATEFORM = (MDY | DMY | YMD | YDM)

This parameter specifies how dates are displayed and input. Dates are displayed with a 4-digit year format unless a 2-digit year format is specified. You can use periods or back slashes (/) to separate the month, day, and year values.

This parameter applies only to Gregorian dates.

Value	Description
MDY	<p>This value indicates that dates are displayed or input in one of the following formats:</p> <ul style="list-style-type: none"> • MMDDYYYY • MM/DD/YYYY • MM.DD.YYYY • MMDDYY • MM/DD/YY • MM.DD.YY
DMY	<p>This value indicates that dates are displayed or input in one of the following formats.</p> <ul style="list-style-type: none"> • DDMMYYYY • DD/MM/YYYY • DD.MM.YYYY • DDMMYY • DD/MM/YY • DD.MM.YY
YMD	<p>This value indicates that dates are displayed or input in one of the following formats.</p> <ul style="list-style-type: none"> • YYYYMMDD • YYYY/MM/DD • YYYY.MM.DD • YYMMDD • YY/MM/DD • YY.MM.DD
YDM	<p>This value indicates that dates are displayed or input in one of the following formats.</p> <ul style="list-style-type: none"> • YYYYDDMM • YYYY/DD/MM • YYYY.DD.MM • YYDDMM • YY/DD/MM • YY.DD.MM

Modifiable through MODIFY INITPARMS command: NO

DEBUG = nnnnnnnn

Turns on a specific trace option or any combination of options, where nnnnnnnn represents a debug setting in hexadecimal.

By default, this initialization parameter is set to 00200000 to make each SVC dump unique. You can modify DEBUG= settings using the MODIFY command. See [IBM Connect:Direct MODIFY Command](#).

See [Debug Settings](#) for a complete listing of the DEBUG settings, the trace types produced, and the ddnames used for output.

Modifiable through MODIFY INITPARMS command: NO

DEFAULT.PERMISS = (text_permissions | 644, binary_permissions | 755)

This parameter specifies the default file permissions for HFS files that are created by Connect:Direct. If not specified, Connect:Direct defaults to 644 for text files and 755 for binary files.

To enable the permission setting for HFS files that use DEFAULT.PERMISS, set the UNIX System Services UMASK to 000 by using the runtime environment variable, _EDC_UMASK_DFLT. Otherwise, Connect:Direct uses a default UMASK of 022 that changes the write permissions for GROUP and OTHER.

To set the environment variable, define the _EDC_UMASK_DFLT=000 variable in a RECFM=VB type file and allocate the ENVIRON DD in the Connect:Direct startup JCL. For example: //ENVIRON DD DISP=SHR, DSN=\$CD.ENVIRON(TZ).

Value	Description
text_permissions	Default file permissions for text files.
binary_permissions	Default file permissions for binary files.

Permission values contain 3 digits. The first digit indicates the owner's file permissions, the second digit indicates the owner's group's file permissions, and the third digit indicates the file permissions for all others. The following are permission values:

- 0 - No file access is allowed
- 1 - Execute access is allowed
- 2 - Write access is allowed
- 3 - Write and Execute access is allowed
- 4 - Read access is allowed
- 5 - Read and Execute access is allowed
- 6 - Read and Write access is allowed
- 7 - Read, Write, and Execute Access is allowed

Modifiable through MODIFY INITPARMS command: NO

DESC.CRIT = (descriptor code)

This parameter specifies the descriptor code used for critical write-to-operator (WTO) messages. Messages that go to the critical route code are, for example, disastrous session errors or critical ABENDs. You can specify as many as 16 codes. The default of DESC.CRIT = (2) specifies immediate action is required.

Modifiable through MODIFY INITPARMS command: YES

DESC.NORM = (n,n,...)

This parameter specifies the descriptor code used for normal (WTO) messages. You can specify as many as 16 codes.

DESC.NORM = () specifies that no descriptor code is assigned. The default is no descriptor code.

Modifiable through MODIFY INITPARMS command: YES

DESC.TAPE = (n,n,...)

This parameter specifies the descriptor code for the tape pre-mount message used by Connect:Direct for z/OS. (For more information, see the IBM Connect:Direct for z/OS Facilities Guide.) You can specify as many as 16 codes. The default is 2.

Modifiable through MODIFY INITPARMS command: YES

DSNTYPE = YES | NO

This parameter indicates whether the DSNTYPE will be propagated from the source file (to be used as the default destination file DSNTYPE) or whether it must be coded within the Process. DSNTYPE=YES must be specified in the receiving node's initialization parameters for the receiver to perform the propagation of DSNTYPE from the source file.

Value	Description
Yes	Indicates that the DSNTYPE of the source file (FROM DSN) will be used to create the new output file (TO DSN) if it cannot be obtained from any other COPY statement parameter. The DSNTYPE will be propagated only when the output data is allocated as DISP=NEW or does not exist. A COPY statement parameter that influences DSNTYPE is required when copying to different DSN types, for example, PDSE to PDS or PDS to PDSE.
No	Indicates that the DSNTYPE of the source file (FROM DSN) will not be propagated to the new output file (TO DSN). The DSNTYPE must be supplied by a COPY statement parameter, or else it will take the system default (PDS for partitioned organization or BASIC for sequential organization).

Modifiable through MODIFY INITPARMS command: NO

ECZ.COMPRESSION.LEVEL = 1 | n

This parameter determines the level of compression. The valid value range is 1–9. The default is 1, which usually provides sufficient compression. The data goes through the compression code the number of times indicated by the value specified for the parameter.



CAUTION: Compression consumes significant CPU resources. To avoid degrading performance in your production environment by changing the global, default settings for the extended compression parameters, you should: (1) Review [“Using Extended Compression” on page 403](#) to view test results that describe how changing the global, default values affects performance, and (2) Review Testing the Effects of Changing Values for Extended Compression Parameters in *IBM Connect:Direct for z/OS User Guide* for information on using the DGASACMP offline utility to perform tests to determine whether changing the default values of the extended compression parameters at the global level or by overriding them at the Process level will significantly improve your system performance.

Modifiable through MODIFY INITPARMS command: YES

ECZ.MEMORY.LEVEL = 4 | n

This parameter identifies how much virtual memory is allocated to maintain the internal compression state. This memory is above the 16 megabyte line. The valid value range is 1–9. The default is 4. Level 1 requires the least memory (1 KB); level 9 requires the most memory (256 KB).

Compression consumes significant CPU resources. For more information, see the caution for the parameter, [“ECZ.COMPRESSION.LEVEL = 1 | n” on page 456](#).

Modifiable through MODIFY INITPARMS command: YES

ECZ.WINDOWSIZE = 13 | nn

This parameter determines the size of the compression window or history buffer. This memory is above the 16 megabyte line. The valid values are 8–15. The default is 13. Size 8 uses 1 KB of memory, whereas Size 15 requires 128 KB of memory.

Compression consumes significant CPU resources. For more information, see the caution for the parameter, “[ECZ.COMPRESSION.LEVEL = 1 | n](#)” on page 456.

Modifiable through MODIFY INITPARMS command: YES

ESF.WAIT = hh:mm:ss

This parameter specifies the maximum amount of time that IBM Connect:Direct waits before checking for ESF-submitted Processes. When that time expires, IBM Connect:Direct retrieves any Processes submitted through the ESF.

The default is 00:03:00.

Modifiable through MODIFY INITPARMS command: YES

EXPDT = (TT,DD,TD,DT) (if multiple values EXPDT = TT | DD | TD | DT | ALL | NONE (if only one value)

This parameter specifies IBM Connect:Direct system defaults for propagating the expiration date from the FROM data set to a NEW data set. The following table lists the valid keywords for the EXPDT parameter and coding conventions.

Value	Meaning	Result
TT	tape-to-tape	Propagate the expiration date if the data set on the sending side and the data set on the receiving side are both on tape.
DD	DASD-to-DASD	Propagate the expiration date if the data set on the sending side is on DASD and the data set on the receiving side is also on DASD.
TD	tape-to-DASD	Propagate the expiration date if the data set on the sending side is on tape and the data set on the receiving side is on DASD.
DT	DASD-to-tape	Propagate the expiration date if the data set on the sending side is on DASD and the data set on the receiving side is on tape.
ALL		Always propagate the EXPDT from data sets on all device types to data sets on all device types (works only for DASD and tape).
NONE		Never propagate the expiration date of the sending data set to the receiving data set. This value is the default.

If you specify multiple values, enclose them in parentheses and separate them by a comma. If you code a single value, you do not need to enclose them in parentheses. If you code ALL or NONE, you cannot code any other keyword.

The receiving side determines whether or not IBM Connect:Direct propagates the expiration date. If the sending side specifies ALL in its initialization parameter, but the receiving side specifies NONE, the EXPDT is not propagated. Therefore, if the copy is from SNODE to PNODE, the PNODE side makes the determination; if the copy is from PNODE to SNODE, the SNODE side determines if the EXPDT is propagated.

IBM Connect:Direct overrides the EXPDT initialization parameter in a Process when the following conditions occur:

- If you code an EXPDT or RETPD parameter for the receiving side (TO side) in the Process, IBM Connect:Direct uses that EXPDT or RETPD and ignore the initialization parameter EXPDT.
- If you code an EXPDT or RETPD for the sending side (FROM side) in a Process and not for the receiving side, IBM Connect:Direct uses the EXPDT in the Process, according to the EXPDT initialization parameter setting on the receiving side.
- If you do not specify the EXPDT in the Process and the input (FROM) data set is on DASD, IBM Connect:Direct obtains the EXPDT from the DSCB. If the input data set is on tape and the tape is SL or

AL (Standard or ASCII), IBM Connect:Direct uses the tape label. When IBM Connect:Direct dynamically allocates the data set on the receiving side, EXPDT is used, according to the initialization parameter EXPDT setting on the receiving side.

When you transfer a data set with no associated EXPDT, the following occurs:

If an input data set does not have an EXPDT, and the EXPDT is to be propagated, then the dynamic allocation string for the output data set specifies LABEL = EXPDT = 00000. DASD data sets are considered to not have an EXPDT if the DSCB EXPDT is 00000. Tape data sets are considered to not have an EXPDT if the HDR1 label contains 00000 for the EXPDT. When a data set is allocated with LABEL = EXPDT = 00000, the tape header label or the DASD DSCB contains zeroes for the EXPDT on the output data set. If you have a tape management system or DASD management system, their databases can reflect a different EXPDT than the tape label or DASD DSCB, depending upon the defaults on the receiving side.

Modifiable through MODIFY INITPARMS command: YES

EXTENDED.RECOVERY = NO | YES

This parameter specifies whether IBM Connect:Direct Extended Recovery is used.

Value	Description
NO	IBM Connect:Direct Extended Recovery is not used. This is the default value.
YES	IBM Connect:Direct Extended Recovery is used.

Extended recovery is supported in the IBM Connect:Direct/Stand-alone Server and IBM Connect:Direct/Plex environments.

Modifiable through MODIFY INITPARMS command: NO

FASP = (NO | SSP , NO | SSP)

The first positional parameter is used for when this IBM Connect:Direct Server is the PNODE and the second is for when this IBM Connect:Direct Server is the SNODE. The default of NO indicates that the FASP via SSP support is globally set to FASP=NO. IBM Connect:Direct Processes should not attempt to use the FASP transport unless overridden by the NETMAP or Process. SSP indicates that the IBM Connect:Direct Processes should request the FASP transport when in session with SSP unless override by the NETMAP or Process.

Modifiable through MODIFY INITPARMS command: No

FASP.FILESIZE.THRESHOLD = nnn | nK | nM | nG

The default of 1G defines the threshold to limit the use of the FASP transport to specific file sizes. If the estimated file size is less than this threshold, IBM Connect:Direct will proceed as FASP=NO for that file transfer. This value is represented as number of bytes and 1K equals 1024. The sending IBM Connect:Direct Server will apply the threshold using the PNODE's setting.

Modifiable through MODIFY INITPARMS command: Yes

FASP.BANDWIDTH = nnn | nK | nM | nG

This is an optional FASP configuration parameter that specifies how much bandwidth each transfer can use. The default is taken from the license and is negotiated with the remote to the smaller of the values. The value cannot exceed the value defined in the license and SSP will ensure the smaller is used. This value is represented as number of bits and 1K equals 1000.

Modifiable through MODIFY INITPARMS command: Yes

FASP.POLICY = FAIR | FIXED | HIGH | LOW

This is an optional FASP configuration parameter that specifies the 'fairness' of each transfer. The default is FAIR.

Modifiable through MODIFY INITPARMS command: Yes

FIPS = NO | YES

This parameter specifies whether Connect:Direct FTP+ will place System SSL in to FIPS mode. It is a global, non-refreshable initialization parameter that applies to the TLS protocol. If FIPS=YES, this parameter is effective only if the following have been met:

- System SSL requirements to enable and run in FIPS mode have been met. For more information on meeting the System SSL requirements for FIPS mode, see the [System SSL Programming Guide](#).
- Connect:Direct Secure Plus has been enabled by specifying the SECURE.DSN initialization parameter.

Once FIPS mode is requested and successfully set, Connect:Direct FTP+ will not attempt to switch to non-FIPS mode. To switch to non-FIPS mode, update the FIPS initialization parameter to FIPS=NO and restart Connect:Direct FTP+. If Connect:Direct FTP+ is in FIPS mode, any attempt to use the SSL protocol will result in a failure.

Modifiable through MODIFY INITPARMS command: NO

GDGALLOC = GENERATION | DSNNAME

This parameter specifies whether IBM Connect:Direct allocates GDG data set by generation or data set.

Note:

Do not specify both GDGALLOC=GENERATION and GDGENQ=NO. IBM Connect:Direct will change GDGENQ=NO to GDGENQ=YES.

Value	Description
GENERATION	<p>IBM Connect:Direct allocates by generation, so the allocation is DATA.SET.NAME(xx), where (xx) is the relative generation such as (+1), (-3), (0), and so on. Use for SMS GDG files so that duplicate GDG generations are not created. GDGALLOC=GENERATION logic supports sites that use both SMS and non-SMS GDG data sets. The GDGENQ parameter is ignored and IBM Connect:Direct handles the ENQ logic. IBM Connect:Direct performs an exclusive SYSTEMS ENQ on QNAME NDMGDG and uses the base GDG name as the RNAME.</p> <p>This exclusive ENQ causes a serialization within any IBM Connect:Direct on the system. First, IBM Connect:Direct issues a LOCATE to find the current absolute generation number. Next, IBM Connect:Direct performs an exclusive SYSTEM ENQ on QNAME SYSDSN and uses the data set name returned from LOCATE as the RNAME. If successful, IBM Connect:Direct then performs a shared SYSTEM ENQ on QNAME SYSDSN and uses the base GDG name as the RNAME. Dynamic allocation (SVC 99) is performed.</p> <p>If the data set name allocated is not the same as what the LOCATE returned, the process ends with an SDEGDGRI error and the file is deleted. The two SYSDSN ENQs are not released until the end of the COPY step. The NDMGDG ENQ on the base is released (DEQueued) after the allocation is performed.</p> <p>GENERATION is the default value.</p>

Value	Description
DSNAME	<p>IBM Connect:Direct allocates by data set name (i.e., DATA.SET.NAME.G0000V00). If GDGENQ=YES parameter is coded, IBM Connect:Direct handles the ENQ logic. IBM Connect:Direct performs an exclusive SYSTEMS ENQ on QNAME NDMGDG and uses the base GDG name as the RNAME.</p> <p>This exclusive ENQ causes a serialization within any IBM Connect:Direct on the system. First, IBM Connect:Direct issues a LOCATE to find the current absolute generation number. Next, IBM Connect:Direct performs an exclusive SYSTEM ENQ on QNAME SYSDSN and uses the data set name returned from LOCATE as the RNAME. If successful, IBM Connect:Direct then performs dynamic allocation (SVC 99) on the data set name that LOCATE returned.</p> <p>If the ENQ is not successful, the process ends with an SDEGDGRI error and allocation does not occur. The SYSDSN ENQ is not released until the end of the COPY step. The NDMGDG ENQ on the base is released (DEQueued) after the allocation is performed.</p>

If you code GDGALLOC = GENERATION, then for new non-SMS managed files, you must use one of the IBM-approved methods of supplying DCB attributes. For example, you could use one of these methods:

- Code a model DSCB in the Process DCB=(model DSCB data set name).
- Use an existing data set with the attributes desired for the new GDS data set DCB=(cataloged data set name) in the Process.
- Have a model DSCB defined for the generation data group (GDG).
- Use the LIKE=(cataloged data set name) parameter in the Process.

If you fail to use an approved method when creating new GDG data sets by generation, you will receive an allocation error of 048C.

Modifiable through MODIFY INITPARMS command: YES

GDGENQ = YES|NO

This parameter specifies whether or not IBM Connect:Direct uses ENQ on the data set and on the base GDG before allocation to see if another address space or task has this data set or base GDG allocated. This condition applies to output GDG data sets only.

Note:

Do not specify both GDGALLOC=GENERATION and GDGENQ=NO. IBM Connect:Direct will change GDGENQ=NO to GDGENQ=YES.

Value	Description
YES	<p>IBM Connect:Direct performs an ENQ on the entire data set (with the G0000V00 appended to the base GDG name) if the Process is copying to an output GDG data set. The ENQ fails if any job in the system has this data set allocated (including IBM Connect:Direct). Another ENQ is done for the GDG base. This ENQ fails if a different job (excluding IBM Connect:Direct) has any generation of this GDG allocated. If either ENQ fails, the Process is retried according to the ALLOC.CODES, ALLOC.RETRIES, and ALLOC.WAIT initialization parameters. If GDGR is not specified in the ALLOC.CODES, then the Process is not queued for retry.</p> <p>The ENQs are not done for data set names coded in the Process as data.set.GnnnnVnn, only for data sets that specify a relative generation, such as (+1) (0) or (-1).</p>
NO	The GDGENQ parameter is disabled and no ENQ is done for GDG copies. NO=default.

Modifiable through MODIFY INITPARMS command: YES

IMMEDIATE.SHUTDOWN = I | R | (I, nnn | 60) | (R, nnn | 60)

This parameter determines how an immediate shutdown issued through the STOP CD command is executed.

Value	Description
I	<p>The immediate shutdown waits for all Run Task programs to complete before shutting down IBM Connect:Direct. This value is the default</p> <p>nnn <u>60</u></p> <p>This value specifies the number of seconds that IBM Connect:Direct waits after starting the SHUTDOWN before it forces the DTF to terminate. If the wait time expires and IBM Connect:Direct forces a shutdown, the following message is displayed to the operator:</p> <p>SITB999I SHUTDOWN IMMEDIATE timed out.</p> <p>nnn=0 indicates that IBM Connect:Direct does not have a time limit within which it must shut down the DTF.</p>
R	<p>The immediate shutdown commands functions as a runtaskimm shutdown. It terminates all executing Run Task Processes and shuts down IBM Connect:Direct.</p> <p>nnn <u>60</u></p> <p>This value specifies how many seconds IBM Connect:Direct waits after starting the SHUTDOWN before it forces the DTF to terminate. If the wait time expires and IBM Connect:Direct forces a shutdown, the following message is displayed to the operator:</p> <p>SITB999I SHUTDOWN IMMEDIATE timed out.</p> <p>nnn=0 indicates that IBM Connect:Direct does not have a time limit within which it must shut down the DTF.</p>

Modifiable through MODIFY INITPARMS command: NO

INITPARM.BACKUP

A new non-refreshable, global INITPARM is added to determine the SMF record type number and the recording interval, as well as to turn off recording the new SMF record, license permitting.

Note: If the SMF record id of 133 is being used by other applications, you must specify a suitable record id with this initialization parameter.

For X09 licenses, the allowed syntax is:

```
SESSION.HIGHWATER.SMF = (133 | 128-255, 1-60)
```

The first number is the SMF record type. The second number is the recording interval in minutes.

INITPARM.BACKUP = member

Specifies the name of the partitioned data set member that contains the backup of the global initialization parameter file. It is created when IBM Connect:Direct successfully initializes. In case IBM Connect:Direct cannot start up successfully after initparm updates have been applied, you can restore the initparms using this member and then restart IBM Connect:Direct.

Note: The PDS that is used for the INITPARMS in the PARM= of the JCL is the data set used to hold the backup. Ensure that the PDS is large enough to hold both the directory entry (with ISPF type stats) and this backup member.

Modifiable through MODIFY INITPARMS command: NO

INVOKE.ALLOC.EXIT = SEND|RECV|BOTH

Determines whether to invoke the allocation exit upon sending a file, receiving a file, or both sending and receiving a file.

Value	Description
SEND	Invokes the allocation exit upon sending a file.
RECV	Invokes the allocation exit upon receiving a file.
BOTH	Invokes the allocation exit upon both sending and receiving a file. This is the default value.

Modifiable through MODIFY INITPARMS command: YES

INVOKE.ALLOC.EXIT.ON.RESTART = YES|NO

Indicates whether to invoke the allocation exit on restart of a previously failed Process.

Value	Description
NO	This value indicates whether to invoke the allocation exit on restart of a previously failed Process.
YES	This value indicates whether to invoke the allocation exit on restart of a previously failed Process. This is the default value.

Modifiable through MODIFY INITPARMS command: YES

INVOKE.SPOE.ON.SNODEID = NO|YES

This parameter indicates whether to invoke Secure Point-of-Entry when a user codes SNODEID = parameter on the PROCESS.

Value	Description
NO	IBM Connect:Direct does not invoke Secure Point-of-Entry when a user codes SNODEID = parameter on the PROCESS.
YES	IBM Connect:Direct invokes Secure Point-of-Entry when a user codes SNODEID = parameter on the PROCESS.

Modifiable through MODIFY INITPARMS command: NO

MAX.AGE = (nnn , * | *(nnn) | ALL | ALL(nnn) | status_type | status_type (nnn) , list)

This parameter specifies how many calendar days to wait before purging a Process. You can also use this parameter to purge only Processes with a specific status or define a different time to wait for each status type.

Note: IBM Connect:Direct software does not automatically delete Processes when you specify 0.

The following table explains the values.

Value	Description
nnn	The number of days to wait before purging a Process. The maximum value is 32767. After the specified number of days as passed, the Process is deleted. If nnn=0, no MAX.AGE processing is performed.
* (asterisk)	Purge all Processes.

Value	Description
ALL	Purge all Processes with Hold queue error status types (HE, HO, HP, HS, RA, and RH).
status_type	Purge only specified status types (HE, HO, HP, HS, PR, RA, RH, HC, HI, HR, WC, WT, or WX).
list	Specify a list by separating entries with a comma.

Following are the queue status values you can select for automatic removal from the TCQ.

Parameter	Description
ALL	This value indicates a request for all of the queue types for the MAX.AGE parameter. If you specify ALL and another queue type, ALL is ignored.
HE	This value indicates to automatically remove held for error (HE) status values. This parameter is the default.
HO	This value indicates to automatically remove held for operator (HO) status values.
HP	This value indicates to automatically remove held due to Process error (HP) status values.
HS	This value indicates to automatically remove held for suspension (HS) status values.
RA	This value indicates to automatically remove held for restart due to allocation error (RA) status values.
RH	This value indicates to automatically remove restart held (RH) status values.

The order of precedence for MAX.AGE subparameters is:

1. Any specified status types take precedence over the ALL subparameter and the wildcard (*) subparameter.
2. The ALL subparameter (purge types HE, HO, HP, HS, RA, and RH) takes precedence over the wildcard (*) subparameter.
3. The wildcard (*) subparameter (purge all valid status types) takes the last precedence.

An example of the MAX.AGE parameter follows.

```
MAX.AGE=(10,*,ALL(4),HO(20),HI(0))
```

The following table explains the values in the example.

Value	Description
10	The default number of days before a Process is removed from the Process queue.
*	All Process types are removed. Because no waiting period is specified for *, the default of 10 days is used.
ALL(4)	Processes with Hold queue error status types (HE, HO, HP, HS, RA, and RH) are removed after 4 days in the Hold queue. Note that the 4-day waiting period for Processes with these status types overrides the 10-day default waiting period.
HO(20)	Processes with an HO status type are purged after 20 days. As a result, they are not purged when ALL or wildcard (*) Processes are purged.
HI(0)	Processes with an HI status type are not eligible for purge because zero is specified as the waiting period. They are never automatically purged from the Hold queue.

In the following example, Processes in the PR queue are aged after one day. Processes in any other queue or status are not aged:

```
MAX.AGE = (0,PR(1),*(0))
```

Modifiable through MODIFY INITPARMS command: YES

MAX.AGE.TOD = time

This parameter specifies when to automatically purge a Process queue. If omitted, the queue is purged at midnight and at IBM Connect:Direct initialization. You can use any valid IBM Connect:Direct time format for the TIME parameter.

In the following example, the Process queue purge runs at 2:30 p.m.

```
MAX.AGE.TOD=14:30
```

Modifiable through MODIFY INITPARMS command: YES

MAXPRIMARY = number of PNODE sessions

A PNODE session is started when initiating a Process to one or more SNODEs. This parameter specifies the maximum number of PNODE sessions that can be active concurrently. The range is from 2 to 512. The default is 6.

Note: We recommend you code a MAXPROCESS value of 150 or less for performance reasons. If you code MAXPROCESS, the value of this parameter can be up to the MAXPROCESS. There is no advantage to setting it higher than MAXPROCESS. If you do not code a MAXPROCESS, MAXPRIMARY + MAXSECONDARY becomes the default for MAXPROCESS. In that case, the combined values of MAXPRIMARY and MAXSECONDARY should equal 150 or less.

Modifiable through MODIFY INITPARMS command: NO

MAXPROCESS = number of executing PNODE and SNODE Processes

This parameter specifies the maximum number of executing PNODE and SNODE Processes allowed at one time. The value allowed is between 2 and 1024, inclusive. The default is the value of MAXPRIMARY + MAXSECONDARY.

Modifiable through MODIFY INITPARMS command: NO

Note: Although 1024 is the maximum valid value for this parameter, MAXPROCESS should be set to 150 or less.

Note: If you are using IBM Connect:Direct for z/OS based on a license for the Simultaneous Session metric, you should set the MAXPROCESS value to be equal to or less than the number of Simultaneous Session entitlements included with your license. You must obtain entitlements sufficient to cover the highest number of sessions that are or have been simultaneously in existence across all instances or copies of the software you have installed. Using the MAXPROCESS parameter allows you to control and limit the number of Simultaneous Sessions used with each copy of IBM Connect:Direct for z/OS.

MAXRETRIES = number of retries

This parameter specifies the maximum number of retries that is made to start a node-to-node session. If IBM Connect:Direct cannot start the session, any Processes destined for the secondary node are placed in the timer queue for retries (TI RE). After all retries are exhausted, they go into the HO WC (hold queue, waiting connection). The range for MAXRETRIES is from 0–512. The default is 7. For related information, see the WTRETRIES initialization parameter.

Modifiable through MODIFY INITPARMS command: YES

MAXSECONDARY = number of SNODE sessions

A SNODE session is started when receiving a Process from a PNODE. This parameter specifies the maximum number of SNODE sessions that can be active concurrently. The range is from 2 to 512. The default is 6.

Note: We recommend you code a MAXPROCESS value of 150 or less for performance reasons. If you code MAXPROCESS, the value of this parameter can be up to the MAXPROCESS. There is no advantage to setting it higher than MAXPROCESS. If you do not code a MAXPROCESS, MAXPRIMARY + MAXSECONDARY becomes the default for MAXPROCESS. In that case, the combined values of MAXPRIMARY and MAXSECONDARY should equal 150 or less.

Modifiable through MODIFY INITPARMS command: NO

MAXSTGIO = maximum storage used for sequential data set transfers for system-determined NCP, maximum I/O storage for user-specified NCP)

This parameter specifies the maximum amount of storage used for BSAM sequential data set transfers. MAXSTGIO has two positional parameters that limit the total I/O buffer size in different circumstances. The first is used to limit it when the system determines the number of channel programs (NCP). The second is used to limit it when you specify the NCP in the COPY statement. (The larger the value, the better the I/O performance for sequential file transfers. However, the larger the value, the larger the REGION size that may be required for the DTF.) The valid value range for both parameters is 60000–8,388,608 (60K–8M). The default value is 1048576 (1M).

- IBM Connect:Direct uses these parameters to limit the number of buffers/channel programs used for sequential I/O and calculates the NCP by dividing the MAXSTGIO value by the block size of the data set being transferred. The number of channel programs specified can range from 0 (to have the system determine the value) to 255. For more information on how IBM Connect:Direct processes sequential data sets using BSAM, see [How to Improve BSAM Data Transfer Rates](#).

For example, if you specify the default of 1 MB for MAXSTGIO, the following number of channel programs/buffers are allocated for data sets with the block sizes listed in the following table. Also listed is the amount of storage required for buffers for this transfer, which is a product of the block size and NCP.

BLKSIZE	Number of Channel Programs/Buffers	Storage Used for Transfer
80	255	20,400
4,080	192	783,360
6,400	128	819,000
27,998	32	895,936

- The data sets in the above table did not take advantage of striping or Large Block Interface (LBI) support, which affect BSAM sequential data set transfer rates. In addition, the method for determining NCP did not vary—the NCP values listed above were all system-determined. You can also specify the NCP by using the second positional parameter of the MAXSTGIO initialization parameter.
- If you specify a large value for MAXSTGIO, be sure to review the REGION size specified for the DTF. The region size must be large enough to accommodate the maximum number of sequential transfers that could take place at any one time, multiplied by the value coded for MAXSTGIO, plus the normal amount of region that the DTF requires.

Modifiable through MODIFY INITPARMS command: YES

MAX.TAPE = number of tape Processes | NONE

This parameter specifies the maximum number of tape Processes that are allowed to start in a node.

Value	Description
number of tape Processes	<p>The numeric range is 0–32767. The default is 10.</p> <p>When this limit is reached, one of two events can occur to any Processes that try to allocate a tape unit:</p> <ul style="list-style-type: none"> • The Processes end with a return code of 8 and a SDETAPRI message • If the ALLOC.CODES initialization parameter includes the code TAPR, the Processes are placed on the timer retry queue. They are then retried the number of times specified in the ALLOC.RETRIES parameter, at the interval specified in the ALLOC.WAIT parameter.
NONE	This value indicates that the node does not perform tape Processing.

If you specify NONE, any Process that tries to copy from or to a tape on this node ends with a return code of 8 and a SDETAPRI message. You can still copy from or to tapes on the SNODE.

Modifiable through MODIFY INITPARMS command: NO

MAXUSERS = number of users

This parameter specifies the maximum number of interactive users and batch users that can sign on to IBM Connect:Direct at any one time. When this limit is reached, no other users are allowed to sign on. The range for MAXUSERS is from 2–512. The default is 6.

Modifiable through MODIFY INITPARMS command: NO

MCS.CLIST = console operator CLIST library file name

This parameter specifies the file name of the CLIST library of the z/OS console operator. This parameter is required for use of the console operator interface. No default exists.

Note: Multiple Operate CLIST data sets can be concatenated using the SDAGOPLS DD statement in the IBM Connect:Direct JCL. For more information, see the IBM Connect:Direct *Facilities Guide*.

Modifiable through MODIFY INITPARMS command: NO

Note: Multiple Operate CLIST data sets can be concatenated using the SDAGOPLS DD statement in the Connect:Direct JCL. For more information, see the IBM Connect:Direct *Facilities Guide*.

MCS.SIGNON = (SIGNON USERID = (user ID,password) NETMAP = network map [optional parameters])

This parameter specifies the console Signon command of the operator for the Operator interface.

Keyword	Subparameter	Description
USERID	user ID	<p>The user ID of the console operator.</p> <p>Password associated with the user ID of the console operator.</p>
NETMAP	network map	The VSAM file that identifies all valid IBM Connect:Direct nodes and applids in the network.
	optional parameters	Optional parameters associated with the SIGNON command. These parameters are described in <i>Managing Sessions</i> in the <i>IBM Connect:Direct for z/OS User Guide</i> .

You must specify the SIGNON USERID and NETMAP keywords.

This parameter is required for installations that use the console operator interface. You can specify all the parameters allowed on the SIGNON command here. There is no default value.

If a signon without a password occurs in a stage1 exit, the authority is inherited from the TSO user ID used for the signon. If a signon with a password occurs in a stage1 exit, the authority of the user ID in the signon command is used.

Modifiable through MODIFY INITPARMS command: NO

MULTI.COPY.STAT.RCD=not set | CT | MC | M2

This parameter creates statistics records for files copied using the DGADSI0X I/O exit and is particularly useful with the wildcard feature, which can produce large numbers of files. When this parameter is set, a message is sent to the Console each time the DGADSI0X I/O exit copies a file, regardless of what other types of statistics records are being generated. For more information on the DGADSI0X I/O exit, see *Utility Programs in IBM Connect:Direct for z/OS User Guide*.

By default, this initialization parameter is not set. Use the following table to determine the conditions under which to use each setting.

Value	Conditions for Use	Description
Not set	<ul style="list-style-type: none"> DGADSI0X I/O exit is not used. Information for individual files is needed. 	No file completion statistics records are created and no console messages are produced.
CT	<ul style="list-style-type: none"> DGADSI0X I/O exit is used. Pre-existing customer exits detect files using CT records. 	Produces a Copy Termination type of statistics record.
MC	<ul style="list-style-type: none"> DGADSI0X I/O exit is used. Control Center is installed and monitoring the IBM Connect:Direct server. 	Produces a PDS Member Copy type of statistics record, which replaces the member name with the file name provided by the ADRDSSU utility.
M2	<ul style="list-style-type: none"> DGADSI0X I/O exit is used. None of the other conditions apply. 	Produces a “multiple copy” type of statistics record, which shows the file name, current return code, bytes processed as reported by the IBM ADRDSSU utility, and normal Step information.

Modifiable through MODIFY INITPARMS command: YES

NETMAP.CHECK = NO | (ALL | TCP, ALL | BOTH | NODENAME, FAIL | WARN | PASS)

This parameter defines the communication types that perform NETMAP checking, the verification to perform, and the action to take if the node does not exist. This parameter is ignored for CTCA connections.

Value	Description
NO	This value indicates that the IBM Connect:Direct node attempting to establish a session with this IBM Connect:Direct node need not be defined in the network map at this node. This feature is convenient when another IBM Connect:Direct node initiates contact the majority of the time.

Value	Description
<u>ALL</u> TCP	<p>ALL enables NETMAP checking for all communication types except for TCP/IP.</p> <p>TCP enables NETMAP checking for TCP/IP communication.</p> <p>Note: If you code NETMAP.CHECK = TCP, you must provide a network map entry for each TCP/IP node. The adjacent node entry must specify the logical node name, port number, TCP/IP address, and a session type of TCP. For example:</p> <p>ADJACENT.NODE=((UNIX.DALLAS,5555,199.5.5.5,TCP) ENVIRONMENT = UNIX)</p>
<u>ALL</u> BOTH NODENAME	<p>ALL or BOTH (for SNA) enables verification on both the logical node name and APPLID/LUNAME.</p> <p>ALL or BOTH (for TCP) enables verification on both the logical node name and IP address.</p> <p>NODENAME enables verification on the logical node name or TCP Alias names.</p>
<u>FAIL</u> WARN PASS	<p>FAIL indicates that access to the system is denied.</p> <p>WARN indicates that access is allowed, but a warning message is issued.</p> <p>PASS indicates that access is allowed without any warning message being issued.</p>

You must define all three parameters to require that the IBM Connect:Direct node establishing a session with this IBM Connect:Direct node be defined in the network map of this node under certain conditions.

To enable NETMAP checking for all communication types, you must code the NETMAP.CHECK parameter for each. Following is an example.

```
NETMAP.CHECK=(ALL,ALL,FAIL)
NETMAP.CHECK=(TCP,NODENAME,WARN)
```

- The first entry for NETMAP.CHECK causes IBM Connect:Direct to check all communication types, except for TCP, for both NODENAME and APPLID/LUNAME.
- The second NETMAP.CHECK entry checks TCP nodes for NODENAME only. If the node does not exist, IBM Connect:Direct issues a warning message but permits access.

Modifiable through MODIFY INITPARMS command: NO

NETMAP.CHECK.ON.CALL= YES | NO

Indicates how IBM Connect:Direct handles a HOLD=CALL Process at submit time.

Value	Description
NO	This value specifies that the PNODE permits a submit of a Process with HOLD=CALL, even if the SNODE entry is not in the network map of the PNODE. However, the TCP/IP.DEFAULT entry must be in the network map of the PNODE. This value is the default.
YES	This value specifies that the PNODE does not allow a submit of a Process with HOLD=CALL, if the SNODE entry is not in the network map of the PNODE (even if a TCP/IP.DEFAULT entry exists in the network map). If you specify Yes, you must define the SNODE in the PNODE network map.

If you specify Yes, you must define the SNODE in the PNODE network map.

Modifiable through MODIFY INITPARMS command: YES

NODE.QUIESCE.OFF = NODENAME

This parameter indicates that all processing in the specified node is not suspended until a Quiesce is issued with the MODIFY command. You can specify this node-level parameter for as many nodes as required. Node-level initparms are processed in the order specified.

The node name parameter is the 1–16 character local node name specified in the network map of the affected node. You can also specify a partial node name with the * and ? wildcard characters. If you specify * anywhere in the node name other than the very end, you must enclose the node name in single quotes. For example, the following parameter sets Quiesce to OFF on all node names that contain DETROIT.

```
NODE. QUIESCE. OFF = '*DETROIT*'
```

Refer to [Suspending and Resuming Quiesce and Trace Settings](#) for how to suspend normal operations by setting SESSIONS to Q (Quiesce) with the MODIFY command.

Modifiable through MODIFY INITPARMS command: NO

NODE.QUIESCE.ON = NODENAME

This parameter indicates that all processing in the specified node is suspended and no new processing is permitted until a Resume is issued with the MODIFY command. You can specify this node-level parameter to suspend processing for as many nodes as required. Node-level initparms are processed in the order specified.

The node name parameter is the 1–16 character local node name specified in the network map of the affected node. You can also specify a partial node name with the * and ? wildcard characters. If you specify * anywhere in the node name other than the very end, you must enclose the node name in single quotes. For example, the following parameter suspends processing on all node names that contain MIAMI.

```
NODE. QUIESCE. ON = '*MIAMI*'
```

If the parameter is issued on an SNODE to quiesce processing with a PNODE, the session with the PNODE is established. However, as soon as the PNODE node name is determined, the session is terminated. No processing of data occurs.

Use this parameter if you want to suspend processing on a node because of problems, but want other nodes to continue processing. You can also use it if you know that a node will be down for some time.

Refer to [Suspending and Resuming Quiesce and Trace Settings](#) for how to resume normal operations by setting SESSIONS to R (Resume) with the MODIFY command.

Modifiable through MODIFY INITPARMS command: NO

NODE.TRACE.OFF = NODENAME

Turns off tracing for a specified node. You can specify this node-level parameter for as many nodes as required. Node-level initparms are processed in the order specified.

The node name parameter is the 1–16 character local node name specified in the network map of the affected node. Use the * and ? wildcard characters to specify the node name. If you specify * anywhere in the node name other than the very end, enclose the node name in single quotes. For example, the following parameter disables tracing on all node names that contain DALLAS.

```
NODE.NODE.OFF = '*DALLAS*'
```

You can modify NODE.TRACE.OFF= settings using the MODIFY command. See [IBM Connect:Direct MODIFY Command](#).

Modifiable through MODIFY INITPARMS command: NO

NODE.TRACE.ON = (NODENAME,nnnnnnnn)

Turns on a specific trace option or any combination of options for a specified node, where nnnnnnnn represents a debug setting in hexadecimal. You can specify this node-level parameter for as many nodes as required. Node-level initparms are processed in the order specified.

The node name parameter is the 1–16 character local node name specified in the network map of the affected node. You can use the * and ? wildcard characters to specify the node name. If you specify * anywhere in the node name other than the very end, you must enclose the node name in single quotes. For example, the following parameter enables tracing on all node names that contain DENVER and sets the debug bits to 80000000.

```
NODE.TRACE.ON = ('*DENVER*',80000000)
```

You can modify NODE.TRACE.ON= settings using the MODIFY command. See [IBM Connect:Direct MODIFY Command](#).

See [Debug Settings](#) for a complete listing of the DEBUG settings, the trace types produced, and the ddnames used for output.

Modifiable through MODIFY INITPARMS command: NO

NON.SWAPABLE = YES | NO

This parameter specifies whether IBM Connect:Direct is marked as non-swappable.

When NON.SWAPABLE = YES, IBM Connect:Direct is not swapped out during periods of no activity. The default value is YES.

Note: When IBM Connect:Direct is running as a IBM Connect:Direct/PLEX or when CTCA has been initialized, NON.SWAPABLE is forced to YES. Otherwise, this keyword controls the setting.

Modifiable through MODIFY INITPARMS command: NO

PDSE.SHARING = YES | NO

This parameter indicates if the zOS PDSE sharing feature is supported by IBM Connect:Direct. The keyword, PDSESHARING in the IGDSMSxx member in SYS1.PARMLIB, which defines the level of PDSE sharing across subsystems of a sysplex, has two possible values:

- NORMAL sharing allows users to share a PDSE only at a data set level.
- EXTENDED sharing allows users to share a PDSE at both a data set level and member level.

For more information on PDSE sharing, including requirements, and test scenarios and results for both the normal and extended mode, refer to the , which you can find at [IBM Redbook Partitioned Data Set Extended \(PDSE\) Usage Guide](#).



CAUTION: Because PDSE sharing allows multiple users to open the same PDSE member for output, some operations may destroy directories and create data integrity problems. The last user to issue the STOW macro, which replaces an entry on the directory, gets their update permanently applied to the member. To ensure that updates are not lost, keep this consideration in mind so that users can take the appropriate steps.

When you initialize IBM Connect:Direct, the IGWLSHR callable service is used to verify that the operating system can support PDSE sharing. If the operating system cannot support this PDSE sharing level, the SITA641W error message

Level of PDSE.SHARING is not supported

will display and initialization will continue as if the PDSE.SHARING parameter had been specified as NO.

Value	Description
YES	IBM Connect:Direct will support PDSESHARING as defined by the operating system. Note: You must specify SHR as the disposition (DISP) keyword so that you can share a particular data set with other jobs. All other dispositions – OLD, NEW, RPL, and MOD – will continue to serialize the PDSE to ensure that multiple IBM Connect:Direct processes cannot share the same PDSE.
NO	IBM Connect:Direct will not support PDSE sharing. If multiple processes attempt to open the same PDSE for output at the same time, the processes terminate with one of the following error messages: <ul style="list-style-type: none">• SDEPDSRI – PDS already open for output by IBM Connect:Direct.• SDE0210I – Requested data set not available. Allocated to another job. In addition, the processes are placed in the appropriate queue and retried according to any initialization parameters specified.

Modifiable through MODIFY INITPARMS command: YES

PDSENQ = YES | NO

This parameter specifies whether or not IBM Connect:Direct serializes access of output PDSEs for simultaneous directory updates from IBM Connect:Direct and ISPF EDIT or the IBM linkage editor.

Value	Description
YES	An ISPF/IEWL ENQUEUE (or RESERVE if the device is shared) is issued to serialize access for output PDS data sets opened with DISP = SHR. Use of PDSENQ does not prevent simultaneous directory updates from batch jobs or other sources that do not issue the same enqueues. If the ENQ or RESERVE fails, DATA SET IN USE allocation error is issued. The Process retries later if 0210 is specified in the ALLOC.CODES initialization parameter. Because the ISPF editor only enqueues a member of a PDS when an ISPF SAVE of the member is issued, a user can be in edit on the member from a PDS to which IBM Connect:Direct is trying to copy. If you attempt to save a member being edited at the same time that IBM Connect:Direct is copying to that PDS, the TSO/ISPF session hangs until the IBM Connect:Direct COPY operation is complete. When the IBM Connect:Direct COPY operation completes, the ISPF SAVE command executes, thereby overlaying the member that are recently copied by IBM Connect:Direct.
NO	An ISPF/IEWL ENQUEUE is not issued.

Modifiable through MODIFY INITPARMS command: YES

PROCESS.RETENTION = YES | NO

This parameter saves a PNODE Process to the PR queue and retains the completed Process for a period of time. Retaining a Process allows you to troubleshoot problems with a completed Process or with the use of symbolics.

Because a Process will be stored on the TCQ longer, increase the TCQ space before activating this parameter. (For more information on enlarging the TCQ, see *Planning the Installation* in *IBM Connect:Direct for z/OS Configuration Guide*.) A Process remains in the PR queue until you delete it using the DELETE command or it is removed when space is needed or based on the values defined in the MAX.AGE and MAX.AGE.TOD parameters. Use the TCQ.THRESHOLD parameter to identify when to delete Processes in the PR queue, to make space available for newly submitted Processes.

Changing the parameter to NO will not delete Processes from the PR queue. It will prevent more Processes from being added to the PR queue.

Value	Description
YES	After a PNODE Process is complete, it is moved to the PR queue to allow you to troubleshoot a problem with a Process or to view PNODE Processes that have executed.
NO	Completed PNODE Processes will not be added to the PR queue. This is the default.

SNODE Processes cannot be moved to the PR queue.

The TCQ.THRESHOLD parameter is the auto-deletion threshold for the PR queue.

Modifiable through MODIFY INITPARMS command: NO

PRTYDEF = Process priority

This parameter specifies the default priority for Processes submitted to IBM Connect:Direct. If you do not specify priority on the Process statement, IBM Connect:Direct uses the default priority when placing the Process on the TCQ. The priorities range from zero to 15, with 15 the highest priority. The default is 10. This parameter is not valid for LU6.2 and TCP/IP flows.

Modifiable through MODIFY INITPARMS command: YES

QUIESCE = YES | NO

This parameter specifies whether or not IBM Connect:Direct holds Processes from execution.

Value	Description
YES	No DTF-to-DTF sessions are started, but you can establish interactive sessions. Any Process to be executed is placed in the WAIT queue. Refer to Suspending and Resuming Quiesce and Trace Settings for information on how to resume normal operations by setting SESSIONS to R (Resume) with the MODIFY command.
NO	IBM Connect:Direct does not hold Processes from execution.

In a IBM Connect:Direct/Plex environment, this parameter applies to all IBM Connect:Direct/Servers.

Note: When you initialize IBM Connect:Direct for the first time with allocation to a TCQ created by DGADTQFX it is recommended that you specify YES for the QUIESCE parameter. After you delete any unwanted Processes from the TCQ, DTF activity can be resumed using the Modify command.

Modifiable through MODIFY INITPARMS command: NO

QUIESCE.NODE = node name

If you are using the QUIESCE.NODE parameter, you are encouraged to use the NODE.QUIESCE.ON parameter as it provides equivalent functionality with enhanced capabilities. The QUIESCE.NODE parameter is still supported.

This parameter indicates that all processing in the specified node is suspended and no new processing is permitted until a Resume is issued with the MODIFY command. You can specify this parameter up to 15 times, which suspends processing on up to 15 nodes.

The node name parameter is the 1–16 character local node name specified in the network map of the affected node. You can also specify a partial node name followed by an asterisk (*). For example, the following parameter suspends processing on all node names that begin with NODE.CHICAGO.

```
QUIESCE.NODE = NODE.CHICAGO*
```

If the parameter is issued on an SNODE to quiesce processing with a PNODE, the session with the PNODE is established. However, as soon as the PNODE node name is determined, the session is terminated. No processing of data occurs.

Use this parameter if you want to suspend processing on a node because of problems, but want other nodes to continue processing. You can also use it if you know that a node will be down for some time.

Refer to [Suspending and Resuming Quiesce and Trace Settings](#) for how to resume normal operations by setting SESSIONS to R (Resume) with the MODIFY command.

Modifiable through MODIFY INITPARMS command: NO

REMOTE.DUMMY.PASSWORD = [YES | INTERNAL]

This parameter controls whether Signon and Process Start is authorized for remote nodes if a dummy password was specified during Signon.

Value	Description
YES	Any remote can specify a dummy password to obtain Signon and Process Start authorization on the local node.
INTERNAL	Requires the remote node to have the Adjacent Node attribute of INTERNAL (see Trusted Node Security) for the authorization to occur.

Modifiable through MODIFY INITPARMS command: NO

REQUEUE = YES | NO

This parameter specifies whether to requeue Processes which ABEND, such as an x37, or with a return code greater than 4, or to allow any subsequent steps to run, or go to Process termination.

Value	Description
YES	Places the Process in the hold queue if it did not end with any of the errors listed above but ABENDED with a return code greater than 4 and one of the following is true: <ul style="list-style-type: none">• The Process or SUBMIT command has REQUEUE = YES• Neither the Process nor the SUBMIT command has REQUEUE specified, but REQUEUE = YES is specified in the initialization parameters• The data set on the PNODE side is a tape data set YES is the default value.
NO	Executes the remaining steps in a Process following a failed COPY STEP, but the failed COPY STEP is not requeued. If REQUEUE is specified on a PROCESS or SUB statement, it overrides the initialization REQUEUE specification.

This parameter is only effective if checkpointing is in use. REQUEUE only applies to the PNODE, or submitting side that has Process control.

REQUEUE is not effective under any of the following conditions:

- SHUTDOWN IMMEDIATE is requested
- Session error caused the Process to terminate

YES places the Process in the hold queue if it did not end with any of the errors listed above but ABENDED with a return code greater than 4 and one of the following is true:

- The PROCESS or SUBMIT command has REQUEUE = YES
- Neither the PROCESS nor the SUBMIT command has REQUEUE specified, but REQUEUE = YES is specified in the initialization parameters
- The data set on the PNODE side is a tape data set

If a dynamic allocation error occurs, the Process goes to ALLOCATION RETRY. When the specified number of allocation retries is exhausted and if REQUEUE = YES is specified, the Process is placed in the hold queue with a status of HO RA (HO = Held by Operator; RA = Held for Restart Due to Allocation Error).

If the Process is ABENDED, the status on the hold queue is HE (hold/error). If the Process received a return code greater than 4, the status is RH (restart/held).

Modifiable through MODIFY INITPARMS command: YES

RESET.ORIGIN.ON.SUBMIT = YES | NO

This parameter resets the originating node.

Parameter	Description
YES	The originating node is set to the node where the submit is issued. This action applies to all connection types.
NO	The originating node is not set to the node where the submit is issued.

This parameter only affects Processes submitted to the SNODE that use SNODEID or SUBMIT. If you use this parameter, both the sending and receiving nodes must use this parameter. Also test its impact, especially if you or your trading partner use Secure Point-of-Entry (SPOE). If you use SPOE and apply this parameter, you may need to update AUTHFILE entries for user ID/node combinations used by SPOE.

Modifiable through MODIFY INITPARMS command: YES

REUSE.SESIONS = YES | NO

Enables you to control the use of the sessions initiated by the local node. When you select a Process for execution between two nodes, control of the session is negotiated. If only one DTF has work destined for the other DTF, then the DTF with work to process controls the session. If they both have work to process, then the one with the higher priority work controls the session. This negotiation takes place at the completion of each Process. It is possible for the local DTF to initiate a session and be significantly delayed in utilizing that session based on the workload of the partner DTF.

Value	Description
YES	The previously allowed negotiation takes place as described.
NO	The remote DTF is not allowed to utilize the sessions established by the local DTF. (IBM Connect:Direct does not allow Processes that are waiting for an eligible session to run when an SNODE session becomes available.)

Modifiable through MODIFY INITPARMS command: YES

ROUTCDE.CRIT = (route code)

This parameter specifies the route code used for critical WTO messages. You can specify as many as 16 codes. Suppress these messages by typing 0 for the route code.

ROUTCDE.CRIT = (2,8) specifies master console information and teleprocessing control.

ROUTCDE.CRIT = (8,11) specifies teleprocessing control and programmer information. This value is the default.

Modifiable through MODIFY INITPARMS command: YES

ROUTCDE.NORM = (route code)

This parameter specifies the route code used for normal WTO messages. You can specify up to 16 codes. You can suppress these messages by typing 0 for the value.

ROUTCDE.NORM = (2,11) specifies master console information and programmer information.

ROUTCDE.NORM = (11) specifies programmer information. This value is the default.

Modifiable through MODIFY INITPARMS command: YES

ROUTCDE.TAPE = (route code)

This parameter specifies the route code used for the tape mount message issued by Connect:Direct for z/OS. You can specify as many as 16 codes. A specification of 0 suppresses the tape mount message.

ROUTCDE.TAPE = (3,5,11) specifies tape pool, tape library, and programmer information.

ROUTCDE.TAPE = (5,11) specifies tape library and programmer information. This value is the default.

Note: If ROUTCDE.TAPE = 0 is coded and TAPE.PREMOUNT = NO, then the z/OS system mount processing at allocation time holds an ENQ on SYSZTIOT until mount is satisfied.

Modifiable through MODIFY INITPARMS command: YES

RUN.JOB.EXIT = modname

This parameter specifies the name of the IBM Connect:Direct module responsible for user read/write control of job streams. The module name can be from 1–8 characters long. The first character must be alphabetic. No default exists for this parameter.

Specify one of the following interface programs to use with Connect:Direct for z/OS:

- RUN.JOB.EXIT = DGAXACRJ (CA-ACF2)
- RUN.JOB.EXIT = DGAXRACJ (IBM RACF and CA-TOP SECRET)

Sample programs are provided as part of the Connect:Direct for z/OS sample library. They may not meet the normal security requirements of an installation. Modify them accordingly.

You must define a user on all nodes involved in Process execution.

Modifiable through MODIFY INITPARMS command: NO

RUNJOBID = USER | CD

This parameter specifies security environment in force for IBM Connect:Direct RUNJOB Processes.

Value	Description
USER	This value specifies that the Process runs under the ID of the user. This value is the default.
CD	This value specifies that the Process runs under the IBM Connect:Direct DTF ID.

Modifiable through MODIFY INITPARMS command: NO

RUN.TASK.EXIT = modname

This parameter specifies the name of the module responsible for verifying that a user is authorized to run a specified program in the DTF address space. The modname can be from 1–8 characters; the first character must be alphabetic. No default exists for this parameter.

Specify one of the following interface programs to use with Connect:Direct for z/OS:

- RUN.TASK.EXIT = DGAXACFT (CA-ACF2)
- RUN.TASK.EXIT = DGAXRACT (for IBM RACF and CA-TOP SECRET)
- RUN.TASK.EXIT = DGAXSAFT (CA-ACF2 with SAF enabled)

Sample programs are part of the Connect:Direct for z/OS sample library. They may not meet the normal security requirements of an installation. Modify them accordingly.

You must define a user on all nodes involved in Process execution.

Modifiable through MODIFY INITPARMS command: NO

RUNTASK.RESTART = YES | NO

This parameter determines whether a RUN TASK program executes at restart if IBM Connect:Direct is unable to determine whether the program has run.

Value	Description
YES	RUN TASK program executes at restart if IBM Connect:Direct is unable to determine whether the program has run
NO	RUN TASK program does not executes at restart if IBM Connect:Direct is unable to determine whether the program has run

This parameter corresponds to the node where the RUN TASK step executes. For example, if the RUN TASK step is running on the SNODE, the coding of the RUNTASK.RESTART parameter on the SNODE determines whether the RUN TASK program executes at restart.

Modifiable through MODIFY INITPARMS command: YES

S+CMD.ENFORCE.SECURE.CONNECTION = YES | NO

This parameter specifies whether Connect:Direct Secure Plus commands are accepted from the IBM Connect:Direct client API on nonsecure connections.

Value	Description
YES	IBM Connect:Direct will enforce a secure connection, that is, it will not accept Connect:Direct Secure Plus commands from a client API on a nonsecure connection.
NO	IBM Connect:Direct accepts Connect:Direct Secure Plus commands from a client API on a nonsecure connection.

Modifiable through MODIFY INITPARMS command: NO

SECURE.DSN = filename

This parameter specifies the Connect:Direct Secure Plus parameters file.

Note: To enable FIPS mode, you must specify a filename for this parameter.

Modifiable through MODIFY INITPARMS command: NO

SECURE.SSL.PATH.PREFIX = prefix

This parameter specifies the prefix location of the key database that contains the certificates for the SSL protocol. Use this parameter if you are using SSL security with the IBM Connect:Direct Connect:Direct Secure Plus and you are operating in a CD/Plex environment.

Modifiable through MODIFY INITPARMS command: NO

SECURITY.EXIT | SECURITY

This parameter specifies the name of the IBM Connect:Direct exit which performs security checking. A sample security exit, DGAMGSAF, is provided in the \$CD.SDGASAMP library. You can modify this exit if it does not meet your security requirements.

Value	Description
module name	A name 1–8 alphanumeric characters long, with the first character alphabetic.
DATASET	Specifies that the exit is invoked only for file security; the IBM Connect:Direct Authorization Facility is used for access (signon) security.
ALL	Specifies that the exit is invoked for file and access security.
PSTKT	Indicates that the local DTF IBM RACF security is defined to accept IBM RACF PassTicket passwords. For more information, see Generating IBM RACF PassTickets .
OFF	Specifies that no security exists; all requests are valid.

If you do not specify the SECURITY.EXIT parameter or it is commented out of the initialization parameters file, customized security is not performed and the IBM Connect:Direct Authorization Facility (AUTH file) is used.

For the first installation of Connect:Direct for z/OS, specify SECURITY.EXIT = OFF until a security exit is installed.

A user must be defined on all nodes involved in Process execution.

The default is the IBM Connect:Direct Authorization Facility.

Note: You can also code this parameter as SECURITY=

Modifiable through MODIFY INITPARMS command: NO

SECURITY.NOTIFY = YES | NO | HOLD

This parameter specifies whether IBM Connect:Direct sends a message to users informing them of security failures on Processes they have submitted.

Value	Description
YES	Specifies that IBM Connect:Direct sends a message to users informing them of security failures on Processes they have submitted. If you set the SECURITY.NOTIFY initialization parameter to YES and you specify NOTIFY = %USER or NOTIFY = user ID on the Process statement, a security failure sends a TSO notification to the user specified in the NOTIFY parameter of a SUBMIT or PROCESS statement.
NO	Specifies that IBM Connect:Direct does not send a message to users informing them of security failures on Processes they have submitted.
HOLD	Specifies that IBM Connect:Direct places Processes in the Hold queue with a status of HE if the other node returns an error during performance of security checking.

The following scenarios could occur with this parameter:

- SECURITY.NOTIFY = NO and a Process has NOTIFY = user ID specified. If a stage 2 security error occurs on the SNODE, the user ID is not notified. The user ID is notified of all other errors or normal completion. All messages and return codes are in the Statistics File.
- SECURITY.NOTIFY = YES and a Process does not specify NOTIFY. The user is not notified of any errors or normal completion. All messages and return codes are in the Statistics File.

- SECURITY.NOTIFY = YES and a Process has NOTIFY = user ID specified. If a stage 2 security error occurs on the SNODE, the user ID is notified. The user ID is also notified of all other errors or normal completion. All messages and return codes are in the Statistics File.

Modifiable through MODIFY INITPARMS command: NO

SERVER.MSU = number|2147483647

For IBM Connect:Direct Premium edition 5655-X01, this parameter specifies Million Service Units (MSU) value of license. The MSU value is used to compute a Value Unit (VU) which is a unit of measure by which the program can be licensed. This value defined by the product license and is obtained from your license.

The acceptable range is 0–2147483647.

Default: 0

Modifiable through MODIFY INITPARMS command: No.

Note: Server must be recycled to update the value.

Note: If you do not specify a value for this parameter, it defaults to 0.

SERVER.TYPE = [PROD | TEST]

This parameter specifies IBM Connect:Direct server type.

Acceptable values are:

Value	Description
PROD	This value indicates that the Server and its licence is intended to be a production copy of the C:D server. This value is default.
TEST	This value indicates that the Server is intended for testing and migration activities. License restriction do not apply to this Server.

Modifiable through MODIFY INITPARMS command: No.

Note: Server must be recycled to update the value.

Note: Your product license defines the server environment support, Production or Test, and its restrictions.

SESSION.HIGHWATER.SMF = (133 | 128-255, 1-60) | NO

This parameter determines the SMF record type number and the recording interval for High Water Mark records. For License permitting the acceptable range is (133 | 128-255, 1-60). The default value is (133 , 60) where the second value is in minutes.

Note:

- If the SMF record id of 133 is being used by other applications, you must specify a suitable record id with this initialization parameter. The value of NO is only valid for license versions X01 and X12. X09 and X11 licenses require the SESSION.HIGHWATER.SMF parameter.
- If message SITA386E or SITA386W appears, it means the default or specified record type is being suppressed by SMF. Refer to D SMF, O display and look for NOTYPE sub parameter of the SYS parameter to see which records are being suppressed.

SNA = YES | NO

This parameter specifies if IBM Connect:Direct initializes with SNA support. You must also specify a valid VTAM APPLID in the local node record of the NETMAP.

Value	Description
YES	IBM Connect:Direct tries to open the VTAM ACB. If it cannot open the ACB, IBM Connect:Direct prompts the operator for the next action.
NO	Only Processes running under TCP/IP or CTCA run or connect to the DTF.

If you change this parameter to SNA = NO after IBM Connect:Direct initializes, you must restart IBM Connect:Direct.

Modifiable through MODIFY INITPARMS command: NO

SNMP = YES | NO

Initializes the SNMP trap agent environment.

Value	Description
YES	Enables the SNMP trap agent environment.
NO	Disables the SNMP trap agent environment.

Modifiable through MODIFY INITPARMS command: YES

SNMP.DSN = data set name | data set name (member)

This parameter identifies the data set used by the SNMP task to initialize the default trap variables and user defined trap triggers. The data set contains the trap events that you want to disable and any trap triggers you define. All traps are enabled by default. A sample data set is installed in the \$CD.SDGASAMP library called DGAXSNMP .

Modifiable through MODIFY INITPARMS command: YES

SNMP.MANAGER.ADDR = hostname | IP address

This parameter determines the TCP/IP address or hostname of the host where the SNMP network manager is initialized. By default, this address is the same as the IBM Connect:Direct TCP/IP address, or the local hostname. In a IBM Connect:Direct/Plex environment, the default is the TCP/IP address for the IBM Connect:Direct Manager. You may specify the IP address as an IPV4 or IPV6 address.

This parameter is required if the SNMP network manager resides on a different host or is required to use a different TCP/IP address.

Modifiable through MODIFY INITPARMS command: YES

SNMP.MANAGER.PORTNUM = port-number

This parameter is the TCP/IP port that is defined for UDP traffic to the SNMP network manager. The default is port number 162. If the defined UDP port number is something other than 162, this parameter is required.

Modifiable through MODIFY INITPARMS command: YES

SNODE.ARCH.RECALL.WAIT = NO | YES

There is no default value for this parameter however the default behavior is as follows.

A Process that allocates a migrated dataset will either:

- WAIT for the dataset to be recalled
- Terminate with an SDEARCHI RC=08

The action taken depends if ARCH is coded in ALLOC.CODES initialization parameter. If ARCH is not included in the ALLOC.CODES then the Process will WAIT for the RECALL to complete. If ARCH is included then the Process will terminate with the SDEARCHI RC=08. The behavior is the same regardless if this IBM Connect:Direct is the PNODE or SNODE.

With SNODE.ARCH.RECALL.WAIT the RECALL behavior will be controlled based on the setting of this parameter when this Connect:Direct is the SNODE.

SNODE.ARCH.RECALL.WAIT=YES (Process waits for the RECALL to be completed)
 SNODE.ARCH.RECALL.WAIT=NO (Process terminates with SDEARCHI RC=08)

STAT.ARCH.CONFIRM = YES | NO

This parameter indicates whether or not IBM Connect:Direct is to have confirmation that the contents of a statistics file pair are archived before erasing them and reusing the file pair to record new information.

Value	Description
YES	<p>Specifies that IBM Connect:Direct requires confirmation before reusing the file. The Connect:Direct for z/OS utilities DGADARRT and DGADARBT provide archive confirmation. You can invoke these utilities from an archive Process or an archive batch job, respectively.</p> <p>If archive confirmation has not occurred at the time a file is to be switched to and therefore erased, IBM Connect:Direct issues a WTOR requesting operator permission to overwrite the file. DTF activity halts until you type a response to the WTOR. An affirmative response causes an immediate file pair switch. A negative response disables the statistics logging function, but the DTF remains active.</p>
NO	<p>Specifies that IBM Connect:Direct erases the file contents at the time of a pair switch regardless of whether indication that the file was archived is received.</p> <p>Note: If you code the STAT.ARCH.CONFIRM parameter as YES, then also specify the STAT.SWITCH.SUBMIT parameter.</p>

Modifiable through MODIFY INITPARMS command: NO

STAT.BUFFER.ESDSDATA | STAT.BUFFER.KSDSINDX | STAT.BUFFER.KSDSDATA

This parameter specifies the number of buffers VSAM allocates for the statistics clusters. IBM Connect:Direct uses the values when generating VSAM access method control blocks (ACBs) for the statistics files. Generating these blocks provides a means of tuning VSAM performance for statistics file access in the DTF. IBM Connect:Direct specifies separate buffers for the index and data components for the key sequenced clusters. Each buffer is the size of the control interval of the specified component.

Note: These buffers are allocated above the 16 megabyte line.

The syntax for this parameter is as follows:

```
STAT.BUFFER.ESDSDATA = number of ESDS data buffers
STAT.BUFFER.KSDSINDX = number of KSDS index buffers
STAT.BUFFER.KSDSDATA = number of KSDS data buffers
```

The defaults are:

- STAT.BUFFER.ESDSDATA = 6
- STAT.BUFFER.KSDSINDX = 6
- STAT.BUFFER.KSDSDATA = 6

Modifiable through MODIFY INITPARMS command: NO

STAT.ERROR = ABEND | DISABLE

This parameter specifies the action of the DTF for certain types of errors which can occur in the Statistics Facility, such as VSAM errors or repeated ABENDs.

Value	Description
ABEND	Specifies that the DTF ABENDs with U3400. This value is the default.
DISABLE	Specifies that the Statistics Facility is disabled but the DTF remains active. The DTF operates normally. However, no statistics records are written.

When an ABEND occurs within the Statistics Facility, an SVC dump is written to a SYS1.DUMPxx data set and recovery is attempted. After five recovery attempts, the DTF ABENDs with U3400 or the Statistics Facility is disabled, depending on the value specified for the STAT.ERROR parameter.

Modifiable through MODIFY INITPARMS command: NO

STAT.EXCLUDE = (record type list)

This parameter specifies what record types to exclude from the statistics log. The system does not pass excluded records to the statistics exit. The 2-character identifiers specify the record types in the list. Refer to [“Statistics Records” on page 318](#) for a complete list of record type identifiers.

You can also selectively exclude using the Statistics exit. See [“Recording Statistics for Specific Record Types” on page 378](#) for more information. You can also turn recording of specific record types on and off during DTF execution using the STATISTICS ON/OFF API command.

The following example excludes PDS member records from the statistics log.

```
STAT.EXCLUDE = (MC)
```

Statistics records are often useful or indispensable in debugging problems. Excluding records from the statistics log makes problem determination difficult. Do not exclude the following record types:

Record Type	Description
CT	Copy Termination
PS	Process Submit
PT	Process Termination
RJ	Run Job
RT	Run Task
SW	Submit within Process
WO	Write to Operator (WTO)

No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: YES

STAT.INIT = WARM | COLD

This parameter specifies whether to erase the contents of the statistics files for the DTF at initialization.

Value	Description
WARM	Specifies that the system does not erase the contents at DTF initialization. In this case, statistics from prior DTF executions are available in the new execution. This value is the default.
COLD	Specifies that the system erases all preexisting records. Only records generated during the current execution are available.

Modifiable through MODIFY INITPARMS command: NO

STAT.QUEUE.ELEMENTS = statistics record queue size

This parameter specifies the size of the queue that holds statistic records to be written.

The range of the STAT.QUEUE.ELEMENTS initialization parameter is 1-9999 elements, with a default value of 1500. A queue that is too small can degrade system performance, whereas a queue that is too large can cause wasteful storage allocation above the 16 megabyte line. In a Connect:Direct Plex environment, the queue element number is calculated by taking the CDPLEX.MAXSERVER value, adding 1 (for the manager) and then multiplying it by the STAT.QUEUE.ELEMENTS value. If this is less than 5000, then 5000 is used. If greater than 10000, then 10000 is used.

When a IBM Connect:Direct task writes a statistic record, it queues the record to be written to the statistics facility asynchronously. The statistics facility then processes the queue and writes the statistics record. This parameter controls the size of this queue. When the queue becomes full, tasks that write a statistics record are held until a slot in the queue becomes available.

You should choose the size of the queue based on how busy IBM Connect:Direct is expected to be during peak-use periods. A lightly loaded system (up to 5 Processes executing concurrently) can run with a small queue, so a queue size of 100 elements may be adequate. However, busier systems require a larger queue (200-1500 elements) to avoid performance degradation caused by queue contention.

There are two size considerations for the statistics queue, the total size and the threshold size. The total size is initially specified by the value of the STAT.QUEUE.ELEMENTS initialization parameter, and can be adjusted if necessary. Each stat queue element takes up 2 KB of space and is allocated above the 16 megabyte line. The threshold size is a value smaller than the total size, and represents a portion of the queue that is reserved for peak-use periods. This helps to improve queue availability under a high workload. The threshold size is calculated automatically, and is usually one-fourth the total size. If the threshold size is inadequate, the total queue size should be increased.

Note: To ensure that an adequate amount of virtual storage above the line is allocated for the queue holding the statistics records in a IBM Connect:Direct/Plex environment, specify REGION=0M on the job card that starts IBM Connect:Direct. For more information about storage requirements, see *Planning the Installation in IBM Connect:Direct for z/OS Configuration Guide*.

Modifiable through MODIFY INITPARMS command: NO

STAT.SNODEID = (NO | YES,NO | YES)

This parameter specifies whether the submitter's ID should be placed in the statistics record. The first subparameter applies to version 1 flows (SNA-LU0) and the second subparameter applies to version 2 flows (SNA-LU6.2/TCP).

Note: STAT.SNODEID affects the user ID (original submitter ID or SNODEID) which is used to execute a SUBMIT within a Process, instead of affecting just the submitter's ID in the statistics record.

A value of YES causes the SNODEID, if present, to be placed in the statistics record.

A value of NO causes the submitter's ID to be placed in the statistics record.

Modifiable through MODIFY INITPARMS command: NO

STAT.SWITCH.SUBMIT = dsn [member]

This parameter enables a site to name a sequential data set or a member of a PDS that contains a Process to be submitted at statistics file pair switch time. Use this feature to submit a Process that archives the statistics file pair that has just filled. Alternatively, the Process can submit a batch job which in turn archives the statistics records.

Note: The STAT.SWITCH.SUBMIT parameter is identical in format to the DSN parameter of the IBM Connect:Direct SUBMIT statement. See the [Connect:Direct Process Language](#) help for information on the SUBMIT statement.

If you code the STAT.ARCH.CONFIRM parameter as YES, then also specify the STAT.SWITCH.SUBMIT parameter.

IBM Connect:Direct internally generates a SUBMIT command to submit the Process, and specifies a single symbolic parameter, &EDSN. The symbolic parameter &EDSN specifies the data set name of the entry sequenced cluster just filled. Therefore, the DTF supplies to the archive Process the name of the ESDS cluster to archive.

You can make archived statistics records available to the SELECT STATISTICS command by copying them to a VSAM entry sequenced cluster, and then use the DGADBKEY utility to recreate the associated index information in a VSAM key sequenced cluster.

No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

STAT.SWITCH.TIME = (hh:mm:ss , ...)

This parameter specifies times of day to perform a statistics file switch. The STAT.SWITCH.TIME is in 24-hour clock format. You can specify up to four times in this parameter. The system initiates a switch whenever one of the named times occurs, regardless of whether the currently active files are full. If you do not specify the STAT.SWITCH.TIME parameter, switching occurs whenever a file pair becomes full or in response to the API command STATISTICS SWITCH.

Modifiable through MODIFY INITPARMS command: NO

STAT.TPREC = (start_time, end_time, snaps_per_hour)

This parameter instructs IBM Connect:Direct to create a statistics record that contains the number of Processes and the amount of data sent and received for a node. You can use the statistics record for load balancing and tracking trends.

Value	Description
start_time	This value indicates the start time for the statistics records creation. The valid format is hh:mm:ss. Valid values are 00:00:00 through 23:59:59.
end_time	This value indicates the time when the statistics record generation ends. The valid format is hh:mm:ss. Valid values are 00:00:00 through 23:59:59.
snaps_per_hour	Valid values are 1–60. If you specify a value of 1, IBM Connect:Direct takes one snapshot per hour. If you specify a value of 60, IBM Connect:Direct takes one snapshot per minute, or 60 per hour. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

STAT.USER = (user ID, [password])

This parameter specifies the security ID under which the statistics log is written and any archive Process or batch job runs. Use this parameter when implementing a stage 2 security exit.

A system task (a separate TCB) does the writing of the statistics files to minimize the impact of statistics logging on the throughput of the DTF. File pair switching and archive Process submission is also done by this task. Such processing is done in the background within the DTF, and therefore, has less impact on other activity. Connect:Direct for z/OS creates this task using the security user ID from the STAT.USER parameter. The system also propagates the user ID to the archive Process and to any batch jobs the archive Process submits.

If your site is running with full stage 1/stage 2 security implemented, it is not necessary to supply the password with this parameter.

Value	Description
user ID	<p>Specifies the security ID that IBM Connect:Direct passes to a security exit. It contains 1–8 characters.</p> <p>Note: Certain IBM Connect:Direct statistics records are written with the STAT.USER ID in their user ID field. For example, the S2 records that contain information about the statistics logging Process are written with this ID. Because user ID is one of the indexed statistics record fields, specifying a unique ID facilitates the rapid retrieval of these records through the SELECT STATISTICS command when the TYPE and USER selection criteria are specified.</p>
password	<p>Specifies the current security password. The security exit uses this parameter to validate the current security password.</p>

If you do not specify this parameter, or if you do not implement the stage 2 security exit, the statistics logging task runs with the security ID of the DTF job, and with the user ID of NDM. In this case, the TP and S2 records are written with NDM in their user ID fields.

Modifiable through MODIFY INITPARMS command: NO

STATISTICS.EXIT = modname | (modname[,MANAGER | SERVER | BOTH])

This parameter specifies the name of the IBM Connect:Direct statistics exit module you can invoke to complement the IBM Connect:Direct statistics gathering functions. Use this program to log IBM Connect:Direct information, perform IBM system management facilities (SMF) functions, and log custom information. To use this feature, you must explicitly code this initialization parameter and supply a module name—the default is no security exit.

For more information on the IBM Connect:Direct sample statistics exits, which are located in the \$CD.SDGASAMP library, see [Sample Statistics Exits](#).

In a IBM Connect:Direct /Plex environment, you can specify whether the Manager, the server(s) or both launch the statistics exit. By specifying the MANAGER keyword, only the Manager region launches the statistics exit. By specifying SERVER, all servers launch the statistics exit. Specifying BOTH causes all regions—servers and the Manager—to launch the statistics exit and for the same records to be processed twice.

In a IBM Connect:Direct /Plex environment, MANAGER is the default entity to launch the exit, but you must still specify the module name.

Modifiable through MODIFY INITPARMS command: NO

STRNO.MSG = number | 10

This parameter specifies the number of strings allowed to message file processing.

The acceptable range is 5–100. The default value is 10.

Modifiable through MODIFY INITPARMS command: NO

SUBMIT.EXIT = modname

This parameter specifies the name of the module responsible for controlling changes to IBM Connect:Direct parameters, such as Process name, priority, class, and secondary node. The module name can be from 1–8 alphanumeric characters long, with the first character alphabetic. No default exists for this parameter.

For more information on the IBM Connect:Direct sample exits, which are located in the \$CD.SDGASAMP library, see [Sample Submit Exits](#).

Modifiable through MODIFY INITPARMS command: NO

SYSOUT = class

This parameter specifies the JES output class for spool output generated during DTF execution. The class must be one character in length. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: YES

TAPE.PREMOUNT = YES | NO | LIST

This parameter specifies whether the Connect:Direct for z/OS tape premount message is displayed.

Value	Description
YES	Tape premount message is displayed.
NO	Tape premount message is not displayed.
LIST	LIST instructs IBM Connect:Direct to list up to 10 tape volume serial number on the SVST000I tape premount message.

Note: If ROUTCDE.TAPE = 0 is coded and TAPE.PREMOUNT = NO, then the z/OS system mount processing at allocation time holds an ENQ on SYSZTIOT until mount is satisfied.

Modifiable through MODIFY INITPARMS command: YES

TAPEIO = BSAM | EXCP

This parameter identifies what the TAPEIO setting was prior to Version 5.0.

The default value is BSAM.

This parameter does one thing prior to Version 5.0 and something different afterwards.

Prior to Version 5.0, this parameter controlled what access method was used by the COPY function to read and write to tape, either BSAM or EXCP. When BSAM was specified, BSAM was used to read and write the tape, NOTE TYPE=REL was used to generate tape checkpoint records, and POINT TYPE=REL was used to set the tape restart point. When EXCP was specified, EXCP was used to read and write the tape, a channel command equivalent to NOTE TYPE=ABS was used to generate tape checkpoint records, and a channel command equivalent to POINT TYPE=ABS was used to set the restart point.

After Version 5.0, BSAM is always used to read and write to tape, NOTE TYPE=ABS is always used to generate new tape checkpoint records, and POINT TYPE=ABS is always used to set the tape restart point from new checkpoint records. The TAPEIO parameter is re-purposed and tells IBM Connect:Direct what the TAPEIO setting was prior to Version 5.0 so that it can switch to using POINT TYPE=REL when restarting using pre-Version 5.0 checkpoint records. This information is critical to the successful functioning of checkpoint restart if a tape is created by a IBM Connect:Direct prior to Version 5.0 and the restart attempted on or after Version 5.0. This includes checkpoint records stored in the local node's checkpoint dataset, as well as checkpoint records stored in a remote node's checkpoint dataset.

Thus, the TAPEIO parameter setting becomes irrelevant and can be removed once there is no possibility of restarting a tape COPY function using IBM Connect:Direct prior to Version 5.0.

Modifiable through MODIFY INITPARMS command: NO

TAPEMOUNT.EXIT = modname

This parameter specifies the name of interface to StorageTek Tape Silo software, which provides status information on the volumes to satisfy a tapemount request. You can invoke the exit prior to a tape VOLSER mount request to automatically cancel the request should any volume not be available for the Silo to process. For more information, see [Tapemount Exit](#).

The module name can be from 1–8 alphanumeric characters long, with the first character alphabetic. No default exists for this parameter.

The IBM Connect:Direct sample exit is named DGAXTAPX and located in the \$CD.SDGASAMP library.

Modifiable through MODIFY INITPARMS command: NO

TCP = OES | NO

This parameter specifies whether the TCP/IP connection modules are loaded during initialization and if so, the type of modules.

Value	Description
YES	Specifies the TCP/IP OpenEdition Sockets Interface support. Note: You must install and run the IBM BPX facility, the series of IBM programs that comprise the OES functionality, before you can transfer files using the TCP = OES support of the IBM Connect:Direct. In addition, IBM RACF sites must install the OMVS security segment before using the OES interface to transfer files.
NO	Causes no modules for the TCP/IP connection to load during initialization.

Modifiable through MODIFY INITPARMS command: NO

FASP.FILESIZE.THRESHOLD = nnn | nK | nM | nG

The default of 1G defines the threshold to limit the use of the FASP transport to specific file sizes. If the estimated file size is less than this threshold, IBM Connect:Direct will proceed as FASP=NO for that file transfer. This value is represented as number of bytes and 1K equals 1024. The sending IBM Connect:Direct Server will apply the threshold using the PNODE's setting.

Modifiable through MODIFY INITPARMS command: Yes

TCP.API.LISTEN = ((addr , port) , (addrn , portn))

Use this parameter to define up to eight different address and port combinations for each server for incoming connection requests.

The TCP.API.LISTEN parameter allows for a list of IP address and port number combinations to support multiple addresses, including IPv6.

For each server in a IBM Connect:Direct Plex environment, override the global initialization parameter by specifying the TCP.API.LISTEN parameter in that server's local initialization parameters. The first address defined in the parameter becomes the local or default address.

There is no default for TCP.API.LISTEN if the parameter is not specified. If the parameter is specified with an address only, the port can default. The default port for TCP.API.LISTEN is 1363.

The syntax for the TCP.API LISTEN parameter is similar to the following example:


```
TCP.API.LISTEN = ( (addr1 , port1) , (addrn , portn) )
```

In the example, addr1 through addrn is specified as either the word ANYADDR or ANYADDR6, a TCP hostname or a specific IP address. Also, port1 through portn is specified as a single port number. The following example demonstrates this specification:

```
TCP.API.LISTEN =(MVSA , 1363)          /* establish single API listen
```

Value	Description
ANYADDR6	When you specify ANYADDR6, 0::0 or (::), both IPv4 and IPv6 connection requests are accepted through this listen. To define a single listen task to accept IPv4 and IPv6 requests, specify the TCP.API.LISTEN parameter as follows. TCP.API.LISTEN=(ANYADDR6,1363)

Modifiable through MODIFY INITPARMS command: NO

TCP.API.TIMER = 00:00:00 | hh:mm:ss

This parameter specifies the maximum time of session inactivity a TCP/IP connected IUI or API session waits before timing out and exiting. The default value of 00:00:00 indicates that no timer is used. The effective range is 00:20:00–02:00:00 (20 minutes–2 hours).

If you are using Control Center to monitor your Connect:Direct for z/OS server, set this value to at least twice the value of the Monitor Rest Time setting in Control Center.

Note: It is recommended that you set the value for the TCP.API.TIMER parameter to at least one minute greater than the wait timeout value set for IBM system management facilities (SMF) to avoid problems when restoring migrated data sets. For information on the TIMEOUT parameter, see Signing On to Connect:Direct for z/OS in *IBM Connect:Direct for z/OS User Guide*.

Modifiable through MODIFY INITPARMS command: YES

TCP.CONNECT.TIMEOUT= 0 | number of seconds

This parameter defines the length of time in seconds that a non-DTF interface (IBM Control Center, Connect:Direct Browser User Interface, TELNET, port scanners etc) will be allowed to linger when an attempt is made to connect to the TCP.LISTEN port rather than the TCP.API.LISTEN port.

The default value is 30.

Modifiable through MODIFY INITPARMS command: YES

TCP.FMH.TIMER=hh:mm:ss

This parameter defines the length of time in hours, minutes, and seconds that the TCP session can be inactive waiting on a IBM Connect:Direct FMH to be received from the remote node before the TCP session is terminated.

The default value is 00:10:00. For a IBM Connect:Direct Run Task step, the timer value does not apply.

The IBM Connect:Direct Process is placed in the Timer Retry queue and retried after the amount of time specified by the WTRETRIES initialization parameter has elapsed. However, if the session times out a second time, the Process is placed in Held in Error status and remains in the Hold queue to give you an opportunity to analyze the problem to prevent this Process from retaining TCP resources.

Care should be taken in specifying these timer values in that IBM Connect:Direct Processes could perform differently than expected due to the session being prematurely terminated.

Use this parameter only when necessary, and when you do, set it to an extremely high value. It is difficult to estimate how much time it takes, for example, for a file to be allocated by means of a manual tape mount. Through trial and error, you should be able to select an appropriate value for this parameter.

Modifiable through MODIFY INITPARMS command: NO

TCP.LISTEN = ((addr , port) , (addrn , portn))

Use this parameter to define up to 64 different address and port combinations for each server for incoming connection requests.

The TCP.LISTEN parameter allows for a list of IP address and port number combinations to support multiple addresses, including IPv6.

For each server in a IBM Connect:Direct Plex environment, override the global initialization parameter by specifying the TCP.LISTEN parameter in that server's local initialization parameters. The first address defined in the parameter becomes the local or default address.

There is no default for TCP.LISTEN if the parameter is not specified. If the parameter is specified with an address only, the port can default. The default port for TCP.LISTEN is 1364.

The syntax for the TCP.LISTEN parameter is similar to the following example:

```
TCP.LISTEN = ( (addr1 , port1) , (addrn , portn) )
```

In the example, addr1 through addrn is specified as either the word ANYADDR or ANYADDR6, a TCP hostname or a specific IP address. Also, port1 through portn is specified as a single port number. The following example demonstrates this specification:

```
TCP.LISTEN = ((MVSA , 1364),          - /* using DNS for address & Default
              (10.20.129.3 , 4100),    - /* specific address, specific port
              (10.20.129.3 , 4101))
```

Value	Description
ANYADDR6	When you specify ANYADDR6, 0::0 or (::), both IPv4 and IPv6 connection requests are accepted through this listen. To define a single listen task to accept IPv4 and IPv6 requests, specify the TCP.LISTEN parameter as follows. TCP.LISTEN=(ANYADDR6,1364)

Modifiable through MODIFY INITPARMS command: NO

TCP.RUNTASK.TIMER = hh:mm:ss

The TCP.RUNTASK.TIMER initialization parameter defines the length of time in hours, minutes, and seconds that the PNODE will wait for the RUN TASK on the SNODE to complete before the TCP session is terminated. The default value of 00:00:00 indicates that no timer is used. This timer value only applies to a Run Task function when the RUNTASK is being performed on the SNODE.

Modifiable through MODIFY INITPARMS command: NO

TCP.SOURCEIP = (IPv4 address or DNS | IPv6 address or DNS)

TCP.SOURCEIP defines the default source IPv4 address and IPv6 address for any outbound IBM Connect:Direct connection.

The first positional parameter assigns a global IPv4 address that Connect:Direct can bind to for outbound Processes. The IPv4 default will be obtained from the Hostname assigned to the TCP/IP task.

The second positional parameter assigns a global IPv6 address that Connect:Direct can bind to for outbound IPv6 Processes. There is no default value and the first IPv6 address defined with the TCP/LISTEN initialization parameter will be used as a default.

Modifiable through MODIFY INITPARMS command:

TCP.SRC.PORTS

This parameter specifies a destination IP address (or multiple addresses) and the source ports associated with the destination addresses. The values in this parameter are loaded into a table that IBM Connect:Direct uses to find a match when establishing a session. It is available only for z/OS nodes using TCP=OES.

The syntax for this parameter is as follows:

```
TCP.SRC.PORTS = (ip.address,port-ranges),(ip.address2,port1,port2), -  
                (ip.address3,port-ranges)  
TCP.SRC.PORTS = (ip.address/submask,port-ranges), ...  
TCP.SRC.PORTS = (ip.address/0xFFFFFFFFFFFFFFFF,ports,ranges)
```

Use a wildcard character [* or 0 (zero)] to define a destination IP address pattern. The wildcards must be in the least significant positions.

You can add an optional subnet mask for the destination IP address, followed by the source port number and/or range of port numbers for the destination IP address.

For IPv4, valid subnet mask values are:

- Dotted quad notation, such as 255.255.0.0
- Hexadecimal notation, such as 0xfffff00

For IPv4 address specification, the ip.address can be fully qualified such as 199.1.1.1, or a generic address such as 199.1.*

You cannot use a subnet mask if you use wildcards in the destination IP address pattern.

Specify the range of source ports from lowest port number to highest port number order. For example, 1025–2000 is valid, whereas 2000–1025 is invalid. The source port numbers must be between 1025–65535, inclusive.

Note: The number of source ports defined must be sufficient to handle the number of concurrent IBM Connect:Direct sessions. If not, performance can be severely affected.

Following is an example.

```
TCP.SRC.PORTS = (199.2.4.*, 5000-5050), -  
                (199.2.4.7, 1376), -  
                (200.200.4.4/255.255.2.4, 2000-2100, 3000-3100), -  
                (138.16.*.*, 2000-2050, 3000-3050, 4001, 4005)
```

For IPv6 specification, the ip.address is specified in IPv6 format with colons. It can be fully qualified such as 1:2:3:4:5:6:7:8 or a generic address 1:2:3:4:*. Generic specification of the IPv6 ip.address cannot use the shortcut specification. For example, 1111:0:0:0:6666:7777:8888 can be specified as 1111::6666:7777:8888 as a fully qualified name. However, 1111::6666:* is not allowed because there is no way to know how many zeros have been eliminated.

For IPv6, valid subnet mask values are:

- Hexadecimal notation, such as 0xFFFFFFFFFFFFFFFFFFFFFFFF00000000

Note: If the specification of an IPv6 address, mask and ports takes more than one line, split at the slash dividing the ip.address from the submask.

The following example demonstrates how to specify an IPv6 address, mask and ports over more than one line:

```
TCP.SOURCE.PORTS=(1111:2222:3333:4444:5555:6666:7777:8888/ -  
0xFFFFFFFFFFFFFFFFFFFFFFFF00000000,02000-03000,03500), -  
(199.1.1.2/255.255.0.0,04000-05000,05500)
```

Modifiable through MODIFY INITPARMS command: NO

TCP.SRC.PORTS.LIST.ITERATIONS = number of scans

This parameter specifies the number of times that IBM Connect:Direct scans the available ports list to attempt a connection before going into a retry state. Use any value between 0 and 255.

Modifiable through MODIFY INITPARMS command: NO

TCP.TIMER = wait time

This parameter specifies the number of seconds that a Process waits on a TCP or UDT data read before the Process is cancelled and put in timer retry status.

The number can range from 0–32767. The default value is 300.

Use a value of 60 or greater to keep Processes from waiting indefinitely because of a lost connection.

This parameter can have the following impacts on Processes:

1. Setting the parameter to 0 causes a Process to become stranded and requires you to manually requeue and restart the Process.
2. Setting this parameter to a high value (such as 1800 [a half-hour] or 3600 [an hour]) to allow long running tasks to complete prevents any shutdowns during that time. Also, long running tasks tie up communications links and associated resources for the time period. System resources are used more efficiently by breaking a task into smaller units, with a RUN JOB submitting the long running Process.

Modifiable through MODIFY INITPARMS command: NO

TCQ = WARM | COLD

This parameter specifies how the TCQ is initialized.

Value	Description
WARM	IBM Connect:Direct uses the TCQ as it exists.
COLD	IBM Connect:Direct reinitializes the TCQ and any Processes left on the TCQ are lost.

Modifiable through MODIFY INITPARMS command: NO

TCQ.THRESHOLD = NO | YES | nn

This parameter specifies how IBM Connect:Direct issues warning messages as the TCQ reaches a defined capacity. Use the TCQ.THRESHOLD parameter to identify when to delete Processes in the PR queue, to make space available for newly submitted Processes.

Value	Description
No	Indicates that a warning message is not produced when control intervals (CIs) reach a certain level on the TCQ file. A message is issued when the TCQ is completely full. This value is the default.

Value	Description
Yes	Indicates that a warning message is issued when the TCQ becomes 90% full. The message is reissued when the percentage changes but remains above 90% full.
nn	Allows you to specify a 2-digit percentage of the number of VSAM file control intervals used on the TCQ file. When the TCQ reaches this percentage, IBM Connect:Direct issues a message. The message is reissued when the percentage changes and remains above the percentage specified here. The range for percentage value is 0 through 99. Note: A value of 0 is the equivalent of the default <u>NO</u> value, that is, no SPQL002I warning message is produced; only when the TCQ is completely full is a message issued.

Processes on the PR queue are automatically deleted on a space-needed basis at non-ESF SUBMIT time, thus not affecting the TCQ.THRESHOLD.

Modifiable through MODIFY INITPARMS command: YES

THIRD.DISP.DELETE = YES | NO

The PNODE setting for the THIRD.DISP.DELETE= parameter controls how IBM Connect:Direct processes the value of the third subparameter of the (TO)DISP= parameter during a COPY operation when an ABEND occurs in the COPY step on the receiving node.

Value	Description
YES	Enables the third disposition delete feature.
NO	Disables this feature.

The following table describes the conditions under which THIRD.DISP.DELETE=YES is enforced and the result it produces.

This processing occurs	When all of these conditions are present
The data set on the receiving node is deleted.	ABEND occurs in the COPY step on the receiving node (SNODE).
	THIRD.DISP.DELETE=YES.
	Third subparameter of the TO(DISP) parameter is not defined, or is defined as DELETE.
	Third subparameter of the TO(DISP) parameter is not defined as KEEP or CATLG.
	The destination output file does not exist, that is, the COPY statement creates a file, which means that if RPL is specified, there is no file to replace.

This processing occurs	When all of these conditions are present
	<p>The DISP parameter of the TO side of the COPY statement is set to one of the following options:</p> <ul style="list-style-type: none"> • DISP=(NEW,CATLG) • DISP=(NEW,CATLG,DELETE) • DISP=(NEW,KEEP) • DISP=(NEW,KEEP,DELETE) • DISP=(RPL,CATLG) • DISP=(RPL,CATLG,DELETE) • DISP=(RPL,KEEP) • DISP=(RPL,KEEP,DELETE)
	<p>The Process is ineligible for REQUEUE because one of the following is true:</p> <ul style="list-style-type: none"> • Checkpointing is turned OFF. • REQUEUE=NO is specified in the Process statement or as an initialization parameter.

When THIRD.DISP.DELETE=NO is set, the following processing occurs:

- IBM Connect:Direct does not apply the third subparameter value, even if it is set to DELETE in the Process itself.
- IBM Connect:Direct uses the value set for the second subparameter of the TO(DISP) parameter as the value for the third subparameter. For example, if the second subparameter is set to CATLG, the third subparameter is treated as if CATLG were specified.



CAUTION: When you create GDG data sets using relative numbering, for example, DATA.SET.NAME(+1), you must specify CATLG as the third subparameter of the TO(DISP) parameter. If you fail to do so, a RETRY failure occurs. For example, you can specify DISP=(NEW,CATLG,CATLG) or DISP=(,CATLG,CATLG).

Modifiable through MODIFY INITPARMS command: YES

TRACE.BUFFER = nnn | 2

This parameter specifies how much space in 64-bit storage is allocated for tracing. By default, two megabytes are allocated for the in-storage wrap trace buffer. To turn off in-storage tracing, specify TRACE.BUFFER=0.

Modifiable through MODIFY INITPARMS command: NO

TRANS.SUBPAS = YES|NO

This parameter specifies whether the submitter password is sent to the receiving node if the receiving node submits within a Process back to the submitting node.

Value	Description
YES	The submitter password is sent to the receiving node
NO	The submitter password is not sent to the receiving node

Modifiable through MODIFY INITPARMS command: YES

UDP.SRC.PORTS

This parameter specifies a destination IP address (or multiple addresses) and the source ports associated with the destination addresses to use when making UDT connections. The values in this parameter are loaded into a table that IBM Connect:Direct uses to find a match when establishing a session.

The syntax for this parameter is as follows:

```
UDP.SRC.PORTS = (ip.address,port-ranges),(ip.address2,port1,port2), -  
                (ip.address3,port-ranges)
```

Use a wildcard character [* or 0 (zero)] to define a destination IP address pattern. The wildcards must be in the least significant positions.

Specify the range of source ports from lowest port number to highest port number order. For example, 1025–2000 is valid, whereas 2000–1025 is invalid. The source port numbers must be between 1025–65535, inclusive.

Note: The number of source ports defined must be sufficient to handle the number of concurrent IBM Connect:Direct sessions. If not, performance can be severely affected.

Modifiable through MODIFY INITPARMS command: NO

UDP.SRC.PORTS.LIST.ITERATIONS = number of scans

This parameter specifies the number of times that IBM Connect:Direct scans the available ports list to attempt a UDT connection before going into a retry state. Use any value between 0 and 255.

Modifiable through MODIFY INITPARMS command: NO

UDT = YES|NO

This parameter specifies whether UDT connection modules are loaded during initialization.



CAUTION: When YES is specified, a JCL override is required for the Manager and Server tasks as described in [“Using the UDT Protocol”](#) on page 502.

Value	Description
YES	Loads modules for the UDT connections during initialization.
NO	Causes no modules for UDT connections to load during initialization.

Modifiable through MODIFY INITPARMS command: NO

UDT.MAXPROCESS = 2 | number of executing UDT Processes

This parameter specifies the maximum number of concurrent UDT processes. Value range 1 to 512. Default is 2.



CAUTION: Although 512 is the maximum valid value for this parameter, you should set UDT.MAXPROCESS to a value much lower than the MAXPROCESS initialization parameter setting. The higher you set the UDT.MAXPROCESS parameter, the greater the negative impact on UDT performance.

Modifiable through MODIFY INITPARMS command: NO

UDT33.LISTEN = ((ipaddr,port)[,(ipaddr,port)...])

Use this parameter to define up to eight different address and port combinations for each server for incoming connection requests.

The UDT33.LISTEN parameter allows for a list of IP address and port number combinations to support multiple addresses, including IPv6.

For each server in a IBM Connect:Direct Plex environment, override the global initialization parameter by specifying the UDT33.LISTEN parameter in that server's local initialization parameters. The first address defined in the parameter becomes the local or default address.

The default port for UDT33.LISTEN is 1366.

Value	Description
ANYADDR6	When you specify ANYADDR6, 0::0 or (:::), both IPv4 and IPv6 connection requests are accepted through this listen. To define a single listen task to accept IPv4 and IPv6 requests, specify the UDT33.LISTEN parameter as follows. UDT33.LISTEN=(ANYADDR6,1364)

Modifiable through MODIFY INITPARMS command: NO

UPPER.CASE = YES|NO

This parameter specifies whether initialization messages sent to the console are displayed in uppercase letters.

Value	Description
YES	Initialization messages sent to the console are displayed in uppercase letters
NO	Initialization messages sent to the console are displayed in uppercase and lowercase letters

Note: You must define this parameter on the execution parameter overrides to ensure that all initialization messages are displayed in uppercase letters.

Modifiable through MODIFY INITPARMS command: NO

V2.BUFSIZE = (maximum transmission buffer size, TCP/IP send/receive buffer size)

The first positional parameter specifies the default maximum buffer size that IBM Connect:Direct uses for LU6.2 and TCP/IP data transmission. Valid values range from 3712–2M.

Note: When running with Secure Plus, the minimum V2.BUFSIZE should be 8K to accommodate certificate information exchanged as part of COPY termination.

The second positional parameter is used to alter the TCP/IP send and receive buffer sizes within TCP/IP. The maximum value is 2M. The minimum value is set by TCPCONFIG in the TCP/IP PROFILE data set. If a value is specified that is lower than the TCPCONFIG value, the TCPCONFIG value is used. The default of the second parameter is double the first parameter, unless this is lower than the value set by TCPCONFIG in the TCP/IP stacks PROFILE data set. For example, TCPCONFIG TCPSENDBFRSIZE 64K TCPRCVBFRSIZE 64K in the TCP/IP PROFILE would set the default to 64 KB. This default value of 64K would be used unless the first parameter is greater than 32 KB, or the second parameter is greater than 64 KB.

The default is V2.BUFSIZE=(32K,128K)

In general terms, the second positional parameter should be at least the same and not less than the first parameter and should be big enough to handle the largest V2.BUFSIZE override from the netmap. A good common practice is to have the second parameter be a multiple of the first parameter and at least twice or more than the first parameter.

Also, note that the first positional parameter can be overridden in the ADJACENT.NODE entry in the netmap using the BUFFER.SIZE parameter. For details, see [BUFFER.SIZE=V2.buffer override](#).

Modifiable through MODIFY INITPARMS command: NO

WTMESSAGE = NO | YES | (YES,nnn)

This parameter specifies whether a WTO message is generated when a Process is placed on the timer retry queue.

Value	Description
YES	Specifies that message SVTM110I is written to the console each time a Process is placed on the timer queue.
NO	Specifies that message SVTM110I is not written to the console each time a Process is placed on the timer queue.
(YES,nnn)	Specifies the number of times the SVTM110I messages are written to the console. Valid value for nnn is 1–512. The default is 1, which specifies that the message is written every time. If you specify 2, every second message is displayed, and so on.

IBM Connect:Direct uses this parameter with the MAXRETRIES initialization parameter when attempting to establish a lost session.

Modifiable through MODIFY INITPARMS command: YES

WTRETRIES = hh:mm:ss | 00:03:00

This parameter specifies the amount of time between attempts to reestablish a node-to-node session. The default is 00:03:00. IBM Connect:Direct uses this parameter with the MAXRETRIES initialization parameter when trying to establish a lost session.

Modifiable through MODIFY INITPARMS command: YES

XCF.NAME = XCF group name

This parameter specifies a unique name for a IBM Connect:Direct/Plex environment, or for a IBM Connect:Direct/Stand-alone Server (including the standby system) using extended recovery. This parameter is not needed in a IBM Connect:Direct/Stand-alone Server that does not use extended recovery.

XCF group names cannot begin with the letters “A” through “J” or with the letters “SYS” because these are reserved by IBM. You cannot modify this parameter through the MODIFY INITPARMS command.

Modifiable through MODIFY INITPARMS command: NO

ZEDC = ON|OFF|PREFERRED

This parameter allows you to enable or disable the support for the zEDC feature.

Value	Description
ZEDC = ON	Enables the support within IBM Connect:Direct for the zEDC accelerator for all record formats. Note: If ZEDC=ON and ZIIP is enabled then ZIIP is preferred.
ZEDC = OFF	Disables the support within IBM Connect:Direct for the zEDC accelerator. Default value.
ZEDC = PREFERRED	Enables the support with Connect:Direct for the zEDC accelerator for all record type with setting zEDC as a preferred compression method.

ZFBA = YES|NO

This parameter allows you to enable or disable the support for the zFBA feature. For more information, see *Using zFBA for File Transfer*.

Value	Description
YES	Enables the zFBA feature within the Connect:Direct Server. Modifiable through MODIFY INITPARMS command : YES
NO	Disables the zFBA feature within the Connect:Direct Server. This is the default value.

ZIIP = NONE | EXTCOMP | SSLTLS | ALL | PROJECT

This parameter enables you to specify the setting for the zIIP Exploitation Feature. For more information, see “zIIP Exploitation Feature” on page 404.

Subparameter	Description
<u>NONE</u>	The default. No CPU time is offloaded to a zIIP and IBM Connect:Direct continues to operate as it has for all versions.
EXTCOMP	Schedules SRBs to offload COPY step extended (ZLIB) compression and decompression CPU time to a zIIP.
SSLTLS	Schedule SRBs to offload COPY step SSL and TLS data encryption and decryption (not the handshake) CPU time to a zIIP.
ALL	Schedules SRBs to offload COPY step SSL and TLS data encryption and decryption (not the handshake) and extended (ZLIB) compression and decompression CPU time to a zIIP.
PROJECT	<p>Same as ALL, except the SRBs are always dispatched to a general purpose (CP) processor, not a zIIP processor. This allows you to project C:D zIIP usage (rather than use the zIIP) when you have a zIIP processor online. If no zIIP processor is online, there is no difference between PROJECT and ALL.</p> <p>Note: The PROJECT setting requires that you set the IEAOPTxx parameter, PROJECTCPU to YES. For more information, see IEAOPTxx (OPT parameters) in MVS Initialization and Tuning Reference (SA22-7592-21 or http://publib.boulder.ibm.com/infocenter/zos/v1r12/index.jsp?topic=/com.ibm.zos.r12.ieae200/ieaopt.htm)</p>

Modifiable through MODIFY INITPARMS command: NO

IBM Connect:Direct System File Initialization Parameters

The following section identifies the initialization parameters for IBM Connect:Direct system files. You must specify each initialization parameter. These initialization parameters are considered global parameters in a IBM Connect:Direct/Plex environment.

AUTHDSN = dsn

This parameter specifies the file name of the IBM Connect:Direct VSAM Authorization file. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

CKPTDSN = dsn

This parameter specifies the file name of the IBM Connect:Direct VSAM Checkpoint/restart file. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

MSGDSN = dsn

This parameter specifies the file name of the IBM Connect:Direct VSAM Message file. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

NETDSN = dsn

This parameter specifies the file name of the IBM Connect:Direct VSAM network map file. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

STAT.ARCH.DIR = archive directory file name

This parameter specifies the data set name of the directory of statistics archive files. Use the directory to maintain information about the files containing archived statistics records. This information includes the date/time range covered by the records in each file, and is useful in locating the archive file containing records for a specific date/time. When you omit this parameter, the archive directory functions are unavailable. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

STAT.DSN.BASE | STAT.FILE.PAIRS

The STAT.DSN.BASE parameter specifies the high-level qualifiers for the statistics files cluster names. Use any valid z/OS data set name qualifiers for this parameter. The high-level qualifier can range from 1–37 characters. The syntax for this parameter is as follows:

```
STAT.DSN.BASE = dsname base
```

The STAT.FILE.PAIRS parameter indicates the number of file pairs to use. You must specify at least two file pairs. The number of file pairs ranges from 2–20. The syntax for this parameter is as follows:

```
STAT.FILE.PAIRS = number
```

The two parameters specify the statistics file pair list. During DTF initialization, IBM Connect:Direct uses these two values to develop the data set names for the statistics files. The low-level qualifier, ESDSnn, is added to the base data set name to form the names of the ESDS clusters. In ESDSnn, nn is the number that identifies the position of the file pair in the list. IBM Connect:Direct uses KSDSnn as the qualifier to form the names of the KSDS clusters.

The following example uses both STAT.DSN.BASE and STAT.FILE.PAIRS to specify the statistics file pair list.

```
STAT.DSN.BASE = CD.STATS          /* STATISTICS DSNAME BASE */
STAT.FILE.PAIRS = 3                /* NUMBER OF PAIRS      */
```

The example in the previous figure generates the following file pair list.

```

CD.STATS.ESDS01      /* FIRST FILE PAIR ... ESDS */
CD.STATS.KSDS01      /* FIRST FILE PAIR ... KSDS */
CD.STATS.ESDS02      /* SECOND FILE PAIR ... ESDS */
CD.STATS.KSDS02      /* SECOND FILE PAIR ... KSDS */
CD.STATS.ESDS03      /* THIRD FILE PAIR ... ESDS */
CD.STATS.KSDS03      /* THIRD FILE PAIR ... KSDS */

```

Modifiable through MODIFY INITPARMS command: NO

TYPEDSN = dsn

This parameter specifies the file name of the IBM Connect:Direct VSAM Type file. No default exists for this parameter.

Modifiable through MODIFY INITPARMS command: NO

Local Initialization Parameters

Local initialization parameters are only used in a IBM Connect:Direct/Plex environment. They only apply to a specific member of the IBM Connect:Direct/Plex. Each IBM Connect:Direct/Plex member must have its own local initialization parameter member with the CDPLEX.MANAGER=YES|NO parameter as the first statement in the member.

You can override any initialization parameter that is allowed in the local initialization parameters member by using the PARM= keyword in the EXEC statement. Local parameters override any global parameter setting; however, the global setting is used if a local parameter is not specified. Some parameters do not apply to all members of the CDPLEX. For example, MAXPROCESS, MAXSECONDARY, and MAXPRIMARY set no process limit for the CDPLEX manager. Likewise, MAXUSER has no purpose for a CDPLEX server, as it does not support API sessions.

Local and Global Initialization Parameters

The following local initialization parameters are also global initialization parameters. See the description for the corresponding Global Initialization Parameter for a description of these parameters:

- “CTCA = NO | YES” on [page 453](#)
- “CTCA.TIMER = number of seconds” on [page 453](#)
- “DEBUG = nnnnnnnn” on [page 454](#)
- “MAXPRIMARY = number of PNODE sessions” on [page 464](#)
- “MAXSECONDARY = number of SNODE sessions” on [page 465](#)
- “MAX.TAPE = number of tape Processes | NONE” on [page 465](#)
- “QUIESCE = YES | NO” on [page 472](#)
- “SECURITY.EXIT | SECURITY” on [page 477](#)
- “SNA = YES | NO” on [page 478](#)
- “STAT.INIT = WARM | COLD” on [page 481](#)
- “STATISTICS.EXIT = modname | (modname[,MANAGER | SERVER | BOTH])” on [page 484](#)
- “TCP = OES | NO” on [page 486](#)
- “TCP.API.LISTEN = ((addr , port) , (addrn , portn))” on [page 486](#)
- “TCP.LISTEN = ((addr , port) , (addrn , portn))” on [page 488](#)
- “TCP.TIMER = wait time” on [page 490](#)
- “TCQ = WARM | COLD” on [page 490](#)
- “TRACE.BUFFER = nnn | 2” on [page 492](#)
- “UDT = YES|NO” on [page 493](#)

- “UDT.MAXPROCESS = 2 | number of executing UDT Processes” on page 493
- “UDT33.LISTEN = ((ipaddr,port)[,(ipaddr,port)...])” on page 493
- “UPPER.CASE = YES|NO” on page 494
- “V2.BUFSIZE = (maximum transmission buffer size, TCP/IP send/receive buffer size)” on page 494
- “ZFBA = YES|NO” on page 496
- “ZIIP = NONE | EXTCOMP | SSLTLS | ALL | PROJECT” on page 496

CDPLEX.INITPARM.BACKUP = member

This parameter specifies the name of the data set member to be used as a backup of the local initialization parameter member if IBM Connect:Direct cannot start up successfully after initparm updates have been applied. To ensure that this backup member contains valid initialization parameters, this member is written to after initialization is completed successfully.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.MANAGER = NO | YES

This parameter indicates whether this IBM Connect:Direct/Plex member is a IBM Connect:Direct/Manager or a IBM Connect:Direct/Server and must be the first statement in all local initialization parameter members. Only one IBM Connect:Direct/Manager is allowed in a IBM Connect:Direct/Plex.

Value	Description
YES	Indicates that this IBM Connect:Direct/Plex member is a IBM Connect:Direct/Manager or a IBM Connect:Direct/Server.
NO	Indicates that this IBM Connect:Direct/Plex member is not a IBM Connect:Direct/Manager or a IBM Connect:Direct/Server. This is the default.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.MSGID = NONE | xx

This parameter specifies two characters that identify the IBM Connect:Direct/Plex member. These characters display after the message number in messages sent to a console operator. They enable the operators to identify the message source.

Following is an example of the message display if the CDPLEX.MSGID value is set to S1. The two character identifier is highlighted in this example.

SVTM055I	S1	SESSION (001) ESTABLISHED WITH	SNODE=SC.DUB.TPYLA2
SVTM055I	S1	SESSION (001) ESTABLISHED WITH	PNODE=SC.DUB.TPYLA2
SVTM036I	S1	PROCESS STARTED MVS MVST3(1) PNODE=SC.DUB.TPYLA2
SVTM036I	S1	PROCESS STARTED MVS MVST3(1) SNODE=SC.DUB.TPYLA2

If your site uses automated operations monitoring and you use this parameter to identify IBM Connect:Direct/Plex members, you may need to revise your monitoring program because of the change in message format. In this case, you may want to leave this parameter at NONE.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.PLEXCLASSES = (*,plexclass,...,plexclass)

This parameter specifies which PLEXCLASSES are supported by the IBM Connect:Direct/Server. You can restrict a IBM Connect:Direct/Server to run only jobs in the specified PLEXCLASSES. An asterisk (*)

indicates that the IBM Connect:Direct/Server supports any Process that does not specify a PLEXCLASS or specifies a PLEXCLASS of ‘*’.

If you specify CDPLEX.PLEXCLASSES, you must explicitly specify ‘*’ as its value to run any Processes with a Plexclass of ‘*’ on that IBM Connect:Direct/Server.

The PLEXCLASS name is 1–8 characters long and up to 8 classes may be specified.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.REDIRECT

This parameter is used by the IBM Connect:Direct Plex Manager in the IBM Connect:Direct Plex environment to determine the redirection address that is presented to the remote node. It allows you to specify redirection addresses based on the security node type (internal or external) and session type (TCP or UDT33) of the adjacent node in the network map for the IBM Connect:Direct Plex Servers within the local initialization parameters.

The syntax for this parameter is as follows:

```
CDPLEX.REDIRECT = ((INT_IPv4,EXT_IPv4),(INT_IPv6,EXT_IPv6),  
(INT_UDT_IPv4,EXT_UDT_IPv4),(INT_UDT_IPv6,EXT_UDT_IPv6))
```

The CDPLEX.REDIRECT parameter is a local initialization parameter for servers only, and defaults to the first IPv4 or IPv6 address indicated for the server.

The following shows an example of the syntax for the CDPLEX.REDIRECT parameter followed by an example with actual TCP/IP and UDT addresses and port numbers.

```
CDPLEX.REDIRECT=((INT TCP IPV4 , PORT),(EXT TCP IPV4 , PORT ) -  
(INT TCP IPV6 , PORT),(EXT TCP IPV6 , PORT ) -  
(INT UDT IPV4 , PORT),(EXT UDT IPV4 , PORT ) -  
(INT UDT IPV6 , PORT),(EXT UDT IPV6 , PORT ) )
```

```
CDPLEX.REDIRECT=((10.20.201.2,4199),(199.1.22.333,4199) -  
(FD00::22CE:0:0:0:82,4299),(FD00::22CE:1:2:3:99,4299) -  
(10.20.201.2,4399),(199.1.22.333,4399) -  
(FD00::22CE:0:0:0:82,4499),(FD00::22CE:1:2:3:99,4499) )
```

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.REDIRECT.EXCEPTION = ((Mgr-IP, Ext_Svr-IP, Ext_Svr-port, Exception-IP, Exception-port),...)

This parameter specifies exception redirection IP addresses and ports in addition to those specified in the CDPLEX.REDIRECT parameter and will be used based on how the remote server contacts the IBM Connect:Direct/Plex Manager. If the remote IBM Connect:Direct server contacts the IBM Connect:Direct/Plex Manager on the Mgr-IP listen address and the IBM Connect:Direct/Plex Workload manager selects a IBM Connect:Direct/Plex server whose CDPLEX.REDIRECT IP and Port matches the Ext_Svr-IP and Ext_Svr-port, the Exception-IP and Exception-port will be used instead of the IBM Connect:Direct/Plex Server's CDPLEX.REDIRECT IP address and port.

This parameter is only valid for the IBM Connect:Direct/Plex Manager.

You can use TCP/IP IPv4, TCP/IP IPv6, UDT IPv4, and UDT IPv6 addresses in the CDPLEX.REDIRECT.EXCEPTION parameter but all of the IP addresses in any given CDPLEX.REDIRECT.EXCEPTION string must be of the same type (IPv4 or IPv6). In addition, the IBM Connect:Direct/Plex Manager must listen on specific IP or UDT addresses and not listen on ANYADDR (0.0.0.0) or ANYADDR6, for example:

```
TCP.LISTEN=((10.20.201.3,3802),(10.20.129.165,3802), -
           (FD00:0:0:22CE::3802,3802))
```

Value	Description
Mgr-IP	IBM Connect:Direct/Plex Manager's Listen IP address
Ext_Svr-IP, Ext_Svr-port	IBM Connect:Direct/Plex Server's CDPLEX.REDIRECT EXTERNAL IP address and port for the IBM Connect:Direct/Plex server selected by Workload Manager for this Process
Exception-IP, Exception-port	Exception IP address and port.

The following shows an example of the syntax for the CDPLEX.REDIRECT.EXCEPTION parameter where:

- If the remote node contacts the IBM Connect:Direct/Plex Manager on 10.20.129.165 and the selected IBM Connect:Direct/Plex Server redirects to 10.20.201.3 port 3811, then 10.20.129.162 port 3811 will be used instead.
- If the remote node contacts the IBM Connect:Direct/Plex Manager on FD00:0:0:22CE::3802 and the IBM Connect:Direct/Plex Server redirects to FD00:0:0:22CE::3803 port 4812, then FD00:0:0:22CE::3805 port 3811 will be used instead.
- Otherwise, the address and port specified in the CDPLEX.REDIRECT parameter will be used.

```
CDPLEX.REDIRECT.EXCEPTION=( -
  ( 10.20.129.165, 10.20.201.3, 3811, 10.20.129.162, 3811) -
  ( FD00:0:0:22CE::3802, FD00:0:0:22CE::3803, 4812, -
    FD00:0:0:22CE::3805, 3811) )
```

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.SERVER = IBM Connect:Direct/Server name

This parameter specifies the name of the IBM Connect:Direct/Server. Use only for IBM Connect:Direct/Plex members whose CDPLEX.MANAGER parameter value is NO. Each IBM Connect:Direct/Server in the IBM Connect:Direct/Plex must have a unique 1–8 character name. The IBM Connect:Direct/Server name cannot be the same as the XCF.NAME name. (The XCF.NAME name is used as the IBM Connect:Direct/Manager name).

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.SERVER.JOBDSN = data set name

This parameter specifies the data set name that contains the various jobs or started tasks that are submitted at initialization. This parameter is only valid for the IBM Connect:Direct/Manager.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.SERVER.JOBMEM = ((member name,server name),...)

This parameter specifies the CDPLEX.SERVER.JOBDSN member containing the JCL or start command that starts the specified IBM Connect:Direct/Server. The member name can be up to 8 characters long.

This parameter is only valid for the IBM Connect:Direct/Manager.

You can specify a maximum of 32 member/server name combinations.

If you want to start the IBM Connect:Direct/Server as a started task in a JES2 environment, use one of the following to issue a start command:

- To issue a START command on the same z/OS image, include START= as the first statement in the member. The member is not submitted as JCL, but a START command is issued with the rest of the first statement.

For example, if START=HOST4100,X is the first statement, then the command START HOST4100,X is issued to z/OS.

- To issue a START command on a different z/OS image, include /*\$VS,'command' as the first statement in the member. The member is submitted to JES, but JES identifies the /*\$VS and issues the appropriate command to z/OS.

For example, if /*\$VS,'RO CSGB,S CDICOMB', the member is submitted to JES and the

RO CSGB,S CDICOMB command is issued by JES to z/OS rather than placed in the job queue.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.SERVER.NODE = node name

This parameter specifies a unique 1–16 character name for a IBM Connect:Direct/Server. This parameter is not used for a IBM Connect:Direct/Manager.

In a IBM Connect:Direct/Plex, all members appear to external IBM Connect:Direct systems as a single local node. However, if you use network map checking and an external IBM Connect:Direct system communicates with more than one IBM Connect:Direct/Server, you can specify a name to identify each IBM Connect:Direct/Server.

The adjacent node's USE.SERVER.NODE network map parameter (see “Keyword Parameters for Adjacent Node Entries” on page 253) must specify that the IBM Connect:Direct/Server use this name in the initial communications.

Modifiable through MODIFY INITPARMS command: NO

CDPLEX.TIMER = 5 | number of minutes

This parameter specifies the time-out value for XCF communications in minutes. The valid range is 0, 5–99. Zero (0) indicates that no time-out is set for XCF communications.

Modified through MODIFY INITPARMS command: NO

CDPLEX.VTAM = (VTAM-APPL,P/S-Node-APPL)

This parameter specifies the VTAM APPLIDs for a IBM Connect:Direct/Server. This initialization parameter is required for a IBM Connect:Direct/Server if SNA=YES is specified in the global initialization parameters.

Modifiable through MODIFY INITPARMS command: NO

Using the UDT Protocol

This section provides information for using UDT (UDP-based Data Transport) to achieve higher throughput rates for file transfers on high-speed networks with latency.

Installation Considerations

Due to requirements for UDT, the installation procedure defines the SDGALINK data set as a PDSE with the following attributes and space allocation requirements:

- RECFM = U
- LRECL = 0
- BLKSIZE = 32760
- DSNTYPE = LIBRARY

Due to higher CPU overhead, the UDT Protocol defaults to NO. For the same reason, the Language Environment XPLINK option is defaulted to OFF. To run with UDT=YES, a JCL override is required to set XPLINK(ON) for Language Environment support. For UDT=YES, add the following DD to both Manager and Server JCL, where \$cdz matches the High Level Qualifier for the local SMP/E installed environment:

- //CEEOPPTS DD DISP=SHR,DSN=\$cdz.SDGAPARM(DGAIXPLK)

Without this override, UDT=YES causes startup to fail with the following message and ABEND:

- SABA001I - DMTCPSTRU S000 U4039 ID=D STATE=\$

Initialization Parameters for UDT Connections

To implement UDT connections, specify the global initialization parameter, UDT=YES, to load the UDT connection modules during initialization. If UDT= NO is specified, no Connect:Direct for z/OS Processes can run using the UDT protocol. If the PNODE is defined to use UDT with a remote node, the remote node must also be set up to accept UDT connections.

To specify the UDT listen ports and related parameters, include the following parameters in the initialization parameter file suitable for your environment:

- UDP.SRC.PORTS
- UDP.SRC.PORTS.LIST.ITERATIONS
- UDT=YES
- UDT33.LISTEN
- UDT.MAXPROCESS



CAUTION: You cannot use the same port for both TCP and UDT listens.

Note: The UDT, UDT33.LISTEN and UDT.MAXPROCESS parameters can be specified as a global or local initialization parameter where in a Connect:Direct for z/OS Plex server environment, the local parameter will override the global setting.

If you have a Connect:Direct for z/OS Plex environment, you may also want to include the CDPLEX.REDIRECT parameter to specify a redirection address to be used by the IBM Connect:Direct Plex Manager based on a UDT33 session type.

Network Map Parameters

To specify UDT as the protocol to use for a particular adjacent node, specify UDT33 as the session type positional parameter. To specify UDT as the protocol used for an alternate address, specify UDT33 as the ALternate.TYPE in the ALternate.COMMinfo keyword parameter definition.

For more information on using UDT as the session type, see [Positional Parameters for Adjacent Node Entries](#).

PROCESS Statement Parameter

To specify a UDT connection not defined in the network map, you can add the value, UDT33NAM, to the SNODE parameter of the PROCESS statement.

In the example below, The CSDPERF.TESTFILE data set is being copied to the SNODE via a UDT connection, which specifies to use the IPv4 address of 10.20.129.20 along with a port value of 1363.

UDTNAM	PROC	SNODE=UDT33NAM=10.20.129.20;1363	
STEP01	COPY	FROM(PNODE	-
		DSN=CSDPERF.TESTFILE	-
		DISP=SHR)	-
		TO(SNODE	-
		DSN=CSDPERF.TESTFILE.OUTPUT	-
		DISP=(NEW,CATLG,DELETE))	

For a complete description of the SNODE parameter and how to use it in the PROCESS statement, see the [Connect:Direct Process Language](#) help.

Chapter 7. User Guide

Introduction to IBM Connect:Direct commands

Use IBM Connect:Direct commands to submit and manipulate Processes stored in the Transmission Control Queue (TCQ). For example, you can select, delete, and suspend Processes by using IBM Connect:Direct commands.

You can use either of the following two methods to issue commands:

Method	Description
Batch Interface	Enables you to issue IBM Connect:Direct commands with the DGADBATC program in a batch environment. See Introduction to the Batch Interface for more information about how to use the Batch Interface.
Interactive User	Provides easy-to-use command entry through fill-in-the-blank screens. The Interactive User Interface (IUI) is an ISPF Dialog online interface. See Introduction to the Interactive User Interface (IUI) , for more information on how to use the IUI.

Some command options which you can select through the IUI do not have a batch equivalent. The documentation notes these exceptions in option descriptions.

Supported commands

The following table lists IBM Connect:Direct commands, which perform various administrative and user functions.

Command	IUI	Function
CHANGE PROCESS	CP	Modify a Process in the TCQ
DELETE PROCESS	DP	Remove a Process from the TCQ
DELETE TYPE	DT	Delete a record from the Type file (both a user and administrative task)
DELETE USER	DU	Delete a user from the Authorization file (administrative task)
FLUSH PROCESS	FP	Terminate an Executing Process (both a user and administrative task)
FLUSH TASK	TF	Remove a task from the Execution queue (administrative task)
INQUIRE	INQ	View information about IBM Connect:Direct DTF status
INQUIRE SNMP	INQ SNMP	Display SNMP Trap Table
INSERT TYPE	IT	Insert a record in the Type file (both a user and administrative task)
INSERT USER	IU	Add a user to the Authorization file (administrative task)
MODIFY	MD	Run diagnostics and modify initialization parameters (administrative task)
SELECT MESSAGE	SEL MSG	Display message detail using message ID
SELECT NETMAP	NM	Display node from the network map file

Command	IUI	Function
SELECT PROCESS	SP	Examine a Process in the TCQ
SELECT STATISTICS	SS, S2	Examine records in the Statistics Log
SELECT TASK	TS	Display IBM Connect:Direct system tasks status (administrative task)
SELECT TCPXLAT	NM	Translate TCP/IP host names - addresses
SELECT TYPE	ST	Examine the records in the Type file (both a user and administrative task)
SELECT USER	SU	Examine user authorization (administrative task)
SIGNON		Connect to IBM Connect:Direct DTF
SIGNOFF	X	Terminate a connection to the IBM Connect:Direct DTF
STATISTICS	STAT	Perform statistics functions (administrative task)
STOP CD	SN	Stop IBM Connect:Direct operation (administrative task)
SUBMIT	SB	Submit a Process for execution
SUSPEND PROCESS	SUS	Suspend Process execution
SWAP NODE	SW	Swap to another IBM Connect:Direct node
UPDATE NETMAP	UNM	Update the network map file (administrative task)
UPDATE TYPE	IT	Update a record in the Type file (both a user and administrative task)
UPDATE USER	IU	Change user privileges (administrative task)

Writing IBM Connect:Direct commands

A IBM Connect:Direct command is a string of characters that conveys your requests to the IBM Connect:Direct Data Transmission Facility (DTF) for execution. You write these commands by using a unique syntax called the IBM Connect:Direct native format.

If you are using the batch interface, operator interface, or your own user-written application, you write the commands in this native format. If you are using the IUI, IBM Connect:Direct builds the commands in the format for you.

Native format

The following table shows the structure of a IBM Connect:Direct command in its native format.

Label	Command	Parameters and Subparameters
-------	---------	------------------------------

The following table explains each command component:

Component	Description
Label (optional)	<p>You can optionally identify your IBM Connect:Direct command through use of a label. A label must begin in position 1 and consists of a 1-8 character alphanumeric string, with the first character alphabetic.</p> <p>Note: The Connect:Direct commands are reserved words and should not be used as labels. For example, SUBMIT, SELECT etc.</p> <p>Note: Any value in column position 1 is considered a label and is processed as such.</p>

Component	Description
Command	<p>The command itself specifies the IBM Connect:Direct function requested and must begin after position 1.</p> <p>Use one or more blanks to separate the command from the parameters which follow it.</p>
Parameters or Subparameters	<p>Parameters or subparameters specify further instructions for the command. Separate your parameters by one or more blanks. Parameters can be either keyword or positional.</p> <p>Keyword parameters are usually followed by an equal sign and can have a set of subparameters. An example of a keyword parameter is CASE in the following command.</p> <p>SIGNON NETMAP=network.map.name CASE=Yes No</p> <p>You must type positional parameters in a specific order, with commas replacing any parameter not typed. These parameters are always on the right of the equal sign. Enclose positional subparameters in parentheses, with the parentheses preceded and followed by blanks or commas.</p> <p>In the following command USERID is an example of a keyword parameter, and ID,pswd,newpswd are examples of positional subparameters:</p> <p>SIGNON USERID=(ID,pswd,newpswd)</p> <p>A positional parameter or the variable information in a keyword parameter is sometimes a list of subparameters. Such a list can include both positional and keyword parameters.</p>

Command example

The following CHANGE PROCESS command changes a preexisting Process named PAYROLL so that the new destination node is DALLAS1 and the Process executes every Friday.

Note: In the following example the command CH should start in column position 2 or greater and would not be appropriate as a label starting in column position 1.

```
CH PROC   WHERE (PNAME=PAYROLL) -
          DEST=DALLAS1          -
          RETAIN=Y              -
          STARTT=(FR)
```

Command syntax

This section describes the syntax used to construct IBM Connect:Direct commands and Processes.

For a complete description of Process syntax and examples, see the *IBM Connect:Direct Process Language Reference Guide*.

Asterisks

Use asterisks to indicate generic specifications of parameters in the SELECT commands. With generics, you request information by specifying just a single asterisk (*) or a character string plus an asterisk.

To examine records for users whose user IDs begin with ST, specify the following parameter:

```
USERID=ST*
```

Commas

Commas function to separate items within a list (except in the case of symbolic substitution. Refer to [Symbolic Substitution](#)), and to control the order of values specified as positional parameters. You must use a comma to indicate omission of a positional parameter. In the following example, the omission of the pswd subparameter is indicated by the extra comma.

```
SIGNON USERID=(ID, ,newpswd)
```

Continuation Marks

A command can continue across multiple lines. Use the hyphen (-), preceded and followed by a space, to indicate that the command continues on the following line. The hyphen can appear anywhere in positions 3-79. The following command is continued on a second line.

```
CHANGE PROCESS WHERE (PNAME=PAYROLL) -  
DEST=DALLAS1
```

Parentheses

In the following example, parentheses enclose lists and associate a group of values.

```
SIGNON USERID=(MYUID1,MYPWD)
```

Symbolic Substitution

Use symbolic substitution to substitute information in a IBM Connect:Direct Process. The substitution is represented by an ampersand (&) plus 1-8 alphanumeric characters. In the following example, the value for &DATA is resolved when you submit the Process.

```
DSN=&DATA
```

If you have multiple symbolics, separate them with one or more spaces.

Single and Double Quotation Marks

The rules for using single and double quotation marks are:

- Single-quote strings allow the parsing of parameters as typed.
- Double-quote strings allow the resolution of symbolic substitution in a quoted string.

The following example shows the use of single quotation marks to enable you to embed special characters or blanks within a parameter or subparameter value.

```
SIGNON PACCT='JOB FOR SYMAINT,DEPT.27'
```

The following example shows the use of double quotation marks to allow for the resolution of symbolic substitution.

```
SIGNON PACCT="JOB FOR SYMAINT,DEPT. &DEPND"
```

Single quotation marks are not valid for symbolic substitution in a IBM Connect:Direct command but can be used with a keyword parameter in a Process statement (refer to the *IBM Connect:DirectProcess Language Reference Guide*)

For example, the following SUBMIT Process command, which uses single quotation marks:

```
SUB PROC=DCB HOLD=YES -  
&DCB='DCB=(LRECL=80,BLKSIZE=3120,RECFM=FM)'
```

results in the following invalid Process.

```
SUB PROC=DCB HOLD=YES      &DCB=' DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB) '
```

The following example which uses the same Process information as the example above but this time with double quotation marks:

```
SUB PROC=DCB HOLD=YES      -  
  &DCB="DCB=(LRECL=80,BLKSIZE=3120,RECFM=FM)"
```

resolves to:

```
SUB PROC=DCB HOLD=YES      &DCB=DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
```

Comments

Comments allow you to include descriptive information within a command. Comments are available for your convenience and do not affect IBM Connect:Direct. Use the following formats for comments:

- Preceded by a slash-asterisk (/*) and followed by an asterisk-slash (*).
- An asterisk (*) in position 1, followed by the comment.

Note: The /* must be paired with an */ even if it goes across multiple lines.

The following lines are valid comments, indicating that the SIGNON command labeled DSIGN signs on to the DALLAS node.

```
/* SIGN ON TO DALLAS */  
*SIGN ON TO DALLAS  
DSIGN SIGNON NODE=DALLAS /*DALLAS*/
```

```
/* Start of my comment  
SIGNON is then followed by  
SIGNOFF  
End of my comment */
```

Concatenation

Use the double bar (||) to concatenate, or link together, character strings. You must precede the double bar and follow it with a blank. Use the double bar to join a long value that continues over multiple records. For example, you can type the following command.

```
PACCT=JOBACCTDATA ||      -  
WITHNOBLANKS
```

The vertical bar is x'4F'.

IBM Connect:Direct resolves your command to the following format:

```
PACCT=JOBACCTDATAWITHNOBLANKS
```

Special Characters

Certain characters cause IBM Connect:Direct to take special actions. These special characters are the hyphen (-), double bar (||), ampersand (&), and the IBM Connect:Direct delimiters. The following table lists the delimiters recognized by IBM Connect:Direct.

Delimiters	Description
	blank
<	less than sign

Delimiters	Description
>	greater than sign
*	asterisk
(open parenthesis
)	close parenthesis
¬	not sign
/	slash
\	backslash
,	comma
.	period
'	single quotation mark
"	double quotation mark
=	equal sign
{	opening brace
}	closing brace
[opening bracket
]	closing bracket

Special Purpose Bracketing

You must often maintain special characters as part of a string. To maintain special characters, enclose the string in bracketing characters. Bracketing characters are backslashes (\), single quotation marks ('), and double quotation marks (").

Bracketing backslashes are indicators of special processing of a character string. IBM Connect:Direct does not maintain them as part of the string at its final resolution. Use bracketing backslashes to:

- Continue a string containing special characters across multiple lines
- Ensure that quotation marks within the string are maintained

The following is an example of using bracketing backslashes in a command:

```
PACCT=\ 'DEPT\MIS\ || -
        \602'\
```

IBM Connect:Direct resolves the command as follows:

```
PACCT='DEPT\MIS602'
```

Indicating case sensitivity

The CASE parameter specifies whether parameters associated with accounting data, user ID, password, and data set name in the commands and Processes are case-sensitive. You can make this designation at the signon level for all commands that are issued for the session that is established by the signon, and at the command level.

The following table describes the levels at which you can set the CASE parameter.

Level	Description
Session	The CASE parameter of the SIGNON command indicates if commands entered during this session are case-sensitive. The default value is NO. You can override this setting at the command level.
Individual Command	<p>You can change the CASE parameter for an individual command and override the session case sensitivity as follows:</p> <ul style="list-style-type: none"> In the Connect:Direct for z/OS IUI, the CASE parameter is usually indicated on the IUI by the following words: DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> Reply Y or N to this question. The value for the CASE parameter stays in effect as long as you reuse the same panel. When you change panels, the value reverts to the session default. In native command mode, such as used in the Batch Interface, you must designate your override on each command record if you want to override the CASE parameter of the SIGNON command.
Process	For commands that refer to Processes already in the TCQ, only the CASE parameter applies to the fields in the command. For commands that refer to Processes that are not yet in the TCQ, the case sensitivity applies to the Process statements.

Indicating selection criteria

Use the WHERE parameter to specify which records to select, change, or delete based on subparameters, such as Process name or Userid.

Selection subparameters are optional; however, you must specify at least one. These subparameters are special in that they identify which records are selected for the command activity defined by the other parameters.

Use the WHERE parameter to indicate selection criteria for commands which affect Process, Task, Type, User, and Statistic records. For Statistics records, you may also specify WHERE2 which provides support for all *NODE selection criteria. The command function applies to all records which match the selection criteria.

For example, in the CHANGE PROCESS command, you can use the WHERE parameter with its PNUMBER subparameter to select a list of Processes by number and release all of them through the RELEASE parameter.

In the IUI, the WHERE parameter is represented by the list of selection criteria which precedes the command parameters themselves. For example, you see the following on the CHANGE PROCESS screen:

```
PROCESS NUMBERS:
==> _____ ==> _____ ==> _____ ==> _____
```

Additionally, you see lists for Process name and submitter to indicate the selection criteria. The IBM Connect:Direct IUI builds the WHERE parameter for the command as it selects the fields from left to right on the screen.

Indicating output destination

The FILE, PRINT, TABLE, and DISPLAY parameters specify the form in which information is presented when the command produces output.

The following table describes each parameter.

Parameter	Description
FILE	<p>Causes IBM Connect:Direct to return the output as unformatted records to the temporary file. You indicate the name of your temporary file through the TMPDSN parameter of your SIGNON command. See “Using SIGNON through the Batch Interface” on page 546, for a description of the TMPDSN parameter. The statistics record macros in the IBM Connect:Direct sample library provide the output record format.</p> <p>When IBM Connect:Direct puts the unformatted records in a temporary file, other programs can process the records and customize reports for specific needs. You can use this functionality to process statistics records. You can archive a statistics file each day with a Generation Data Group (GDG). The IBM Connect:Direct product provides sample job streams that extract statistics on a daily basis.</p>
PRINT	<p>Routes output of the command to the destination specified in the PRINT keyword of the SIGNON command. See “Using SIGNON through the Batch Interface” on page 546, for a description of the PRINT parameter.</p>
TABLE or DISPLAY	<p>Stores the output of the command in the IBM Connect:Direct temporary file or file specified in the TMPDSN parameter and displays it upon successful completion of the command. (For more information on the TMPDSN parameter, see “Using SIGNON through the Batch Interface” on page 546.) The output is usually in tabular format. You can browse the output by using such commands as UP, RIGHT, or FIND. Press the PF1 key for Help on how to browse the file.</p>

Some screens offer additional output format, such as summary tables. These options are listed along with the display, file, and print options.

The batch interface

You request IBM Connect:Direct services in the batch environment through the batch interface program, DGADBATC.

When you submit a job stream that contains IBM Connect:Direct commands such as SYSIN input, DGADBATC reads the input data stream and processes the requested functions. DGADBATC supports all IBM Connect:Direct commands.

Batch interface job requirements

Batch interface jobs must meet certain requirements.

- You cannot use Process statements in the job stream. Use the SUBMIT command with a preexisting Process that contains Process statements. The batch job is notified of a successful SUBMIT, but not whether the PROCESS itself is successful. Unless MAXDELAY is used, DGADBATC processing of the SUBMIT is asynchronous.
- For DGADBATC, specify a region size of 0 megabytes (REGION=0M), especially if you specify MAXDELAY. If you specify a different region size, the DGADBATC job may ABEND with an 878 code.
- The Processes that you submit using the PROC parameter must be in the IBM Connect:Direct Public Process Library, allocated to the DMPUBLIB DD statement in the DGADBATC JCL. If the Process is not in the IBM Connect:Direct Public Process Library, use the DSN parameter of the SUBMIT command to indicate the location of the Process. See [Building, Modifying, and Submitting Processes](#), for the DSN parameter description.

The IBM Connect:Direct commands that you use in the batch job stream must follow the syntax that is outlined in [Writing IBM Connect:Direct Commands](#).

- If you are not using the Extended Submit Facility (ESF), the DTFs that you sign on to must be active when you submit the DGADBATC job.

- You can specify ESF as a SIGNON command parameter. You can only issue SIGNON, SIGNOFF, and SUBMIT commands by using ESF. ESF is only available for Processes you submit on the local node. You cannot use ESF with MAXDELAY.

Processing rules

The following rules apply to IBM Connect:Direct commands and options in the DGADBATC job stream.

- The first command in the job stream must be a SIGNON command.
- The maximum command string length is 4,096 bytes. Each new IBM Connect:Direct command in the job stream must start on a separate line. You can split a command across more than one line by using the continuation mark. However, the first word or string of the next command must start on a new line as shown in the following example.

```
SIGNON USERID=(LYNN) -
      NODE=CD.BOSTON
SEL STAT WHERE (PNUM=24)
SIGNOFF
```

- All IBM Connect:Direct commands must start in column 2. Any data in column 1 is considered a label.

Sample job stream to run the batch interface

The following example shows a sample job stream to run the DGADBATC program. The job stream is located in \$CD.SDG AJCL(DGADBATC).

```
//JOBNAME JOB (ACCTNG), PROGRAMMER, TIME=2, CLASS=A,
//
//DGADBATC EXEC PGM=DGADBATC, REGION=0M, PARM= 'YYSLYNNNNNN'
//STEPLIB DD DISP=SHR, DSN=$CD.SDGALINK
//DMNETMAP DD DISP=SHR, DSN=$CD.NETMAP
//DMPUBLIB DD DISP=SHR, DSN=$CD.SDGAPROC
//DMMSGFIL DD DISP=SHR, DSN=$CD.MSG
//DMPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SIGNON USERID=(USERID, -
      PASSWORD)
SUBMIT PROC=COPY SNODE=CD.NODE.B -
      HOLD=NO -
      &NODE=PNODE -
      &DSN1=DATASET1 -
      &DSN2=DATASET2 -
      &DISP1=RPL -
      &DISP2=CATLG -
      &COMPRESS=COMPRESS
SEL PROC WHERE (QUEUE=A) TABLE
SIGNOFF
/*
```

The following example shows the COPY Process submitted in the previous example as it is displayed in the IBM Connect:Direct Public Process Library, \$CD.SDGAPROC:

```
COPY      PROCESS
STEP1     COPY      FROM(&NODE DSN=&DSN1 DISP=SHR) -
                  TO (DSN=&DSN2 DISP=(&DISP1,&DISP2)) -
                  &COMPRESS
```

DDNAMES for DGADBATC

The DGADBATC program has both required and optional ddnames.

The following table describes the required and optional ddnames for the DGADBATC program shown in the sample job stream example.

DDNAME	Description
STEPLIB	Indicates the location of the library that contains the IBM Connect:Direct load modules. This DD statement is required, unless the data set is in linklist.
DMNETMAP	Indicates the name of the network map for the node you are signing on to. You can also specify this name through the SIGNON command NETMAP parameter. If you use both methods, the network map file name specified on the SIGNON command takes precedence. This DD statement is optional.
DMPUBLIB	Indicates the library that contains the IBM Connect:Direct Processes. This DD statement is required.
DMMSGFIL	Indicates the IBM Connect:Direct message file that contains IBM Connect:Direct messages. This DD statement is required.
DMPRINT	Indicates the destination of the job output from DGADBATC that collects output messages such as error messages. This DD statement is required. Always check DMPRINT output for verification of the processing that occurred.
SYSPRINT	Indicates the destination of the job execution messages. This DD statement is optional.
NDMCMDS	Use to test new job streams. If specified, it prints an image of each command sent to IBM Connect:Direct after processing all symbolic substitution and parameter overrides. This DD statement is optional.
APITRACE	Used to provide an automatic trace of the API code. This DD statement is optional. Note: Depending on the number of commands in SYSIN a large amount of trace data can be generated.
APISECUR	Used to provide an automatic Security trace for the API code. This DD statement is optional. Note: Depending on the number of commands in SYSIN a large amount of trace data can be generated.
SYSIN	Indicates the location of the IBM Connect:Direct commands. You can represent it as a sequential file, PDS member, or instream data. If using a file, its DCB should be RECFM=FB,LRECL=80. This DD statement is required.

DGADBATC EXEC parameters

The DGADBATC EXEC parameters identify optional output formatting routines.

The characters in the PARM keyword in the DGADBATC EXEC statement are required parameters. The output is displayed in the data set defined by DMPRINT. In the following table which lists the positional DGADBATC EXEC parameters, Y stands for yes, N for no, S for short text, and L for long text.

Position	Value	Description
1	Y	Display the command string that executed.
	N	Do not display the command string that executed.
2	Y	Display the API return code and message ID.
	N	Do not display the API return code and message ID.
3	S	Display short message text when the API sends a return code of zero.
	L	Display long message text when the API sends a return code of zero.
	N	Do not display message text when the API sends a return code of zero.

Position	Value	Description
4	S	Display short message text when the API sends a non-zero return code.
	L	Display long message text when the API sends a non-zero return code.
	N	Do not display message text when the API sends a return code greater than zero.
5	Y	Display the data that generated in the temporary file.
	N	Do not display the data that generated in the temporary file.
6	N	Reserved
7	N	Reserved
8	Y	If parameter 9 is specified as N, this parameters prints output from a SELECT STATS command to DMPRINT when a MAXDELAY Process fails with a return code other than 52.
9	Y	Enables monitoring of the MAXDELAY Restart Process. Produces informational messages about monitoring and about the submitted and monitored Process number. For more information about MAXDELAY restart, see <i>Using the MAXDELAY keyword parameter to synchronize submitted Processes</i>
	N	Allows the MAXDELAY Process to execute without restart. The DGADBATC step is suspended until the Process completes execution.
10 Applies to MAXDELAY Restart only.	Y	If parameter 9 is specified as Y, the MAXDELAY progress is processed on a status report to the DMPRINT DD using the Select Process command. Note: Using this parameter may result in a large amount of data for DMPRINT DD.
	N	Suppresses the progress report.
11 Applies to MAXDELAY and MAXDELAY Restart.	Y	Forces the DGADBATC step to terminate on the first return code of 8 or greater.
	N	Allows the DGADBATC step to continue to process the commands in SYSIN even after a non-zero return code.

DGADBATC return codes with DGADCHLA

DGADBATC communicates with the IBM Connect:Direct API through a high-level application interface program, DGADCHLA. The DGADCHLA program detects error situations while processing commands for DGADBATC, and issues special return codes based on these situations.

The return code from DGADBATC reflects the highest return code of all commands processed.

If you receive a return code of 4 or 8, check the DMPRINT DD output for the exact error that is encountered during command processing to DGADBATC.

A return code of 8 normally indicates an invalid parameter or keyword. A return code of 4 indicates that DGADCHLA found nothing for the command. For example, you could receive a return code of 4 if you issued a SELECT STATISTICS command for a Process with no statistics.

If you receive a DGADBATC return code greater than 8, check return codes for error information.

Return Code – Hexadecimal	Return Code – Decimal	Description
0000000C	12	A session is lost in a multiple session environment.
00000010	16	The master session is lost.
00000014	20	The master session is signed off successfully.
0000001C	28	A non-master signon failed.
00000020	32	High-level interface program, DGADCHLA, received an invalid number of input parameters.
00000024	36	The output specification included an invalid parameter.
00000028	40	Invalid pointer to the UICB is passed to DGADCHLA.
00000034	46	MAXDELAY=0 Process never executed and was removed from the Process queue.

Using the MAXDELAY keyword parameter to synchronize submitted Processes

You can synchronize submitted Processes by coding the MAXDELAY keyword parameter in either the SUBMIT command or the PROCESS statement.

To suspend the execution of the DGADBATC job step until the submitted Process either completes or a specified interval of time elapses, use the MAXDELAY keyword.

For a detailed description of the MAXDELAY parameter, see the Submit command in [“SUBMIT Command”](#) on page 560.

Important: Because of the following restrictions, you might want to code MAXDELAY in the SUBMIT command instead of the PROCESS statement:

- If you code MAXDELAY in a PROCESS and submit it through the IUI, the submit will fail with the message SCBI220I MAXDELAY not supported feature for IUI.
- If you use the MAXDELAY PROCESS statement keyword, the message SCBI221I MAXDELAY not supported for ESF submits is displayed when the DTF is down and the Process does not submit.

The following example shows the MAXDELAY keyword that is coded in the SUBMIT command.

```
//SYSIN      DD *
SIGNON  USERID=(USER01,PASSWRD)ESF=NO
SUBMIT  PROC=PROCAA MAXDELAY=UNLIMITED
SIGNOFF
/*
```

You can also place the MAXDELAY keyword in the PROCESS statement for PROCAA, as in the following example.

```
PROCAA  PROC SNODE=REMOTE.NODE.B MAXDELAY=UNLIMITED
STEP01  COPY TO  (DSN=USER01.TESTDATA.OUTPUT -
                  DISP=RPL -
                  SNODE) -
          FROM  (DSN=USER01.TESTDATA.INPUT -
                  DISP=SHR -
                  PNODE)
```

SIGNON Parm Options

When the DGADBATIC parameter nine enables the MAXDELAY Restart, it also enables new SIGNON command parameters, MAXWAIT and MAXRETRY. These parameters control how often and how many times the API attempts to restart a failed API session while monitoring a MAXDELAY process when the ninth DGADBATIC EXEC is set to Y.

MAXRETRY defines the number of attempts to restart the API session if a session failure occurs after a successful SUBMIT using MAXDELAY. If the failure occurs before SUBMIT returns the Process name and Process number, no restart will be attempted. If the MAXRETRY count is exhausted without successfully restarting the session, DGADBATIC terminates with MSGID=SCIB024I. The value for nn has a range of 1 to 20, with a default value of 3.

MAXWAIT defines the interval of delay between restart attempts after a failed API session. The value for nn specifies seconds to wait and has a range of 0 (no wait) to 60 (1 minute), with a default value of 10 seconds.

Example:

```
SIGNON USERID=(joe,joepwd) -  
      MAXRETRY=nn -  
      MAXWAIT=nn
```

MAXDELAY Restart

MAXDELAY Restart allows the API to respond to API session failures while waiting for a MAXDELAY Process to complete and is only valid for processes submitted with the MAXDELAY parameter and activated using new DGADBATIC PARM values. Without the new PARM values, there is no change in behavior for MAXDELAY or the API.

When activated with the new DGADBATIC PARM values, the C:D Server will return Process name and Process number immediately for processes submitted with the MAXDELAY parameter. The API monitors process execution using Select Process commands until the process is no longer active in the queues. When the process is no longer in the queues, the API issues a Select Statistics command to determine process disposition.

Depending on process disposition, the MAXDELAY parameter requested, and DGADBATIC PARM options, the API will either proceed to the next command or issue commands to delete the process and terminate the API session.

If a API session failure occurs while monitoring the MAXDELAY process, the API will attempt to restart the session then continue monitoring the process. API session restarts are controlled by new parameters specified or defaulted from the SIGNON command.

DGADBATIC Parm Options

Three DGADBATIC PARM options activate and control the MAXDELAY RESTART. These values are represented as Y or N values in positions nine, ten and eleven of the PARM string.

Parameter nine enables or disables the MAXDELAY RESTART. To enable MAXDELAY RESTART and shift the MAXDELAY processing to the API, set this parameter to Y. The default for this parameter is N, causing the MAXDELAY, as described above, to be used.

Parameter ten allows the API to display to DMPRINT, a process progress report from the Select Process commands used to monitor the process. To enable this progress report, set this parameter to Y. The default for this parameter is N.

Important: Setting the tenth parameter to Y may produce additional output to DMPRINT for long running MAXDELAY processes. You may want to consider utilizing other facilities, such as the IUI or IBM Control Center to monitor the progress of the process.

Parameter eleven provides control over the remaining commands within SYSIN in the event that a SUBMIT with MAXDELAY terminates with a non-zero return code. When specified as Y, this parameter forces an EOF or EOJ of the API without attempting to execute the remaining commands in SYSIN. This value defaults to N.

For example, if the SYSIN has 100 SUBMIT commands using MAXDELAY and the second submit fails with a non-zero return, with parameter eleven specified as Y the remaining 98 submit commands will not execute.

```
//STEP1 EXEC PGM=DGADBATC,PARM='yylllyyyYNN'
```

"yylllyyy" are the old 8 parameters and YNN represent the new parameters. For more information about parameter options, see [“DGADBATC EXEC parameters”](#) on page 514

MAXDELAY Limitations and Restrictions

There are many considerations to take into account when using PROCESS submission and MAXDELAY. These considerations include:

- ESF and MAXDELAY are mutually exclusive. During an ESF session, the only command possible is SUBMIT. ESF and SUBMIT with MAXDELAY are not possible and results in SCBI221I MAXDELAY, a failure of the SUBMIT. An ESF submit occurs when the API cannot establish the connection to the Server for any reason, such as when the Server is down or when there is not enough SNA APPLID defined in the NETMAP; in these cases, it is the responsibility of the API to construct and assign the process number, and write the process to the TCQ.
- When using a MAXDELAY parameter of UNLIMITED, processes submitted that are not immediately eligible for execution are subject to the intelligent retry logic when a PROCESS to the same SNODE cannot establish a session. These processes will be placed into the HOLD queue while the failing process retries. If this failing process exceeds its maximum retry, all processes for that same SNODE will be placed into the HOLD and will require manual intervention to run. This will cause the API to hang while not producing your desired results.
- If the connection between the API and the C:D Server is lost due to a time out or other network issues, the API is terminated without any knowledge of the results of the submitted process. However, with the DGADBATC EXEC ninth parameter as Y, the MAXRETRY and MAXWAIT will attempt to restart the connection.

How MAXDELAY Affects PROCESS Submission

When the Server constructs and writes the PROCESS to the TCQ, the Server does not communicate the results or process number back to the API; instead, it places the API in a WAIT status. The API waits until the PROCESS completes its execution; it does not know the process number assigned. The API remains in WAIT status and is dependent on the positional parameter defined with the MAXDELAY, unlimited, queued, time value or 0 (zero).

If set as unlimited or MAXDELAY UNLIMITED, the API connection will be in WAIT status until the PROCESS completes execution and is removed from the TCQ. In the event of an error that causes that PROCESS to be placed into the HOLD queue, the API connection will remain in WAIT until manual action is taken for that held PROCESS.

When queued, time value, or MAXDELAY QUEUED, the API connection will be in WAIT until the PROCESS completes execution or is removed from the TCQ or the time interval expires. If the PROCESS executes longer than that specified time, or is placed into the HOLD queue the API will not get the complete or expected results. If 0 (zero) or MAXDELAY ZERO, the API connection will be in WAIT until the PROCESS completes execution and is removed from the TCQ. However, if an error occurs that would normally place the PROCESS into the HOLD queue, the MAXDELAY ZERO PROCESS is removed from the queue and the API is posted complete with an error. An API connection cannot be in WAIT forever with MAXDELAY UNLIMITED.

MAXDELAY Parm Options

The API applies the MAXDELAY restrictions depending on the MAXDELAY option requested, for example:

MAXDELAY = hh:mm:ss or QUEUED

If the process has not completed execution in the interval defined, the DGADBATC is terminated with an appropriate message and a non-zero return code:

MAXDELAY = UNLIMITED or 0 (zero)

When the process completes, the results are reported to DMPRINT and the API proceeds to the next command in the SYSIN input. If the execution causes the process to be placed in the HOLD queue, notification will be reported to DMPRINT and the process will be deleted from the queue and the DGADBATC terminated with an appropriate message and non-zero return code.

Please note this is a change in behavior for MAXDELAY=UNLIMITED and that the process will be deleted similar to MAXDELAY=0.

The Interactive User Interface (IUI)

Issue IBM Connect:Direct commands using the IUI through two menus.

Menu	Description
The Primary Options Menu	Enables you to invoke screens where you can create and submit Processes, view statistics about your Processes, control your Processes in the Transmission Control Queue (TCQ), view your IBM Connect:Direct environment, access the message file, sign on, swap nodes, and sign off IBM Connect:Direct. This option is available to all users, and is described in this book.
The Administrative Options Menu	Enables the IBM Connect:Direct product administrator to perform maintenance to the Type file, the Authorization file, and the network map and to perform functions such as selecting and flushing tasks, initializing traces, typing native commands, displaying storage utilization, requesting traces, and stopping IBM Connect:Direct.

Note: You can also use the Sterling Connect:Direct Browser User Interface to perform some of the procedures in this chapter.

Primary Options Menu

The Primary Options Menu contains the IBM Connect:Direct functions which are available to most users.

```

View  Modify  Control  Create  Execute  Help
-----
CD.ART          IBM Connect:Direct for z/OS          Primary Options Menu
Option ==>

Select one of the following:

CF  - Copy a file                                     *****
SB  - Submit a predefined process                     *           *
DF  - Define a process using ISPF Edit                 * Today: 08.30.2018 *
SS  - View statistics for a completed process          * Time: 14:00      *
S2  - Advanced view statistics                        * UID:  EPETE1     *
MB  - Submit a batch to Connect:Enterprise for z/OS   *           *
CP  - Change characteristics of a process             * OPT Enabled     *
DP  - Delete a non-executing process                  * OPT Part-Enabled *
FP  - Flush an executing process                     * OPT Disabled    *
SP  - View data about an executing process            *           *
PS  - Suspend an executing process                   *****
MSG - View Connect:Direct message text
SW  - Swap among concurrent Connect:Direct sessions
SD  - View/Change your Connect:Direct signon information defaults
NM  - View information in the Connect:Direct network map
WHO - View characteristics of your Connect:Direct IUI environment
SPF - Enter ISPF/PDF
AUTH - View your Connect:Direct function authorization
MS  - Sign on to multiple Connect:Direct nodes concurrently
ADMIN - Perform Connect:Direct administrative functions

```

To request a function, type its option identifier on the command line (CMD==>) and press **Enter**.

Selection Panel Flower box legend

The Primary Option Menu, like other IUI selection menus, has a flower box with the date, time, UID, and a legend of color scheme indicating user's authority to functions on the menu. If an option navigates to another selection menu, the color indicates the user's authority to the functions on that menu.

- If an option has the same color as the flower box string **OPT Enabled** (default=WHITE), then the user is fully authorized to use the function, or all the functions in the menu it navigates to.
- If the option is the same color as **OPT Part-Enabled** (default=GREEN), then the user is authorized for some of the functions in the menu it navigates to.
- If the option is the same color as **OPT Disabled** (default=BLUE) then the user is not authorized for the function or any function in the menu it navigates to.

For example, if the Primary Option Menu **ADMIN** option is BLUE, the user is not authorized for any of the functions on the ADMIN selection menu, and cannot even navigate to it via the Primary Option Menu. If the ADMIN option is GREEN, the user is authorized for at least one, but not all of the functions on the ADMIN selection menu, and can navigate to it. If the ADMIN option is WHITE, the user is authorized for all the functions on the ADMIN selection menu, and can navigate to it.

The color scheme enables users to see what they are authorized to use on the selection menus, without having to navigate to the **User Authorization** screen and interpret its contents. The colors can be changed by your Connect:Direct System Administrator.

Create and Submit Processes

The following options enable you to create and submit Processes to the TCQ:

Option	Description
CF	Displays the main Copy File Menu where you can set up a COPY Process.
SB	Displays the Submit Process screen where you can submit a predefined Process.
DF	Displays the Process Definition Screen where you can define or update a Process using the ISPF edit function and then submit the Process.

See [Building, Modifying, and Submitting Processes](#) for more information.

The following option enables you to view statistics related to completed Processes:

Option	Description
SS	Displays the Select Statistics screen where you can examine the system statistics log records with search criteria applied and select the output to go to a file, table, or printer. See Process Results and Statistics for more information on the Select Statistics screen.
S2	Displays the Advanced Select Statistics screen where you can enter all selection criteria in one scrollable panel. It also allows you to use the new *NODE selection criteria.

The following options enable you to delete, flush, view, suspend, and change Processes in the TCQ:

Option	Description
CP	Accesses the Change Process screen where you can change the priority and class of a Process and the status of the Process in the TCQ. See Modifying a Process in the TCQ with CHANGE PROCESS for more information on the Change Process screen.
DP	Displays the Delete Process screen where you remove a nonexecuting Process from the TCQ.
FP	Displays the Flush Process screen where you can remove an executing Process from the TCQ.
SP	Displays the Select Process screen where you view the status of Processes submitted (placed in the TCQ) for execution. See Controlling Processes with Commands for information.
PS	Displays the Suspend Process screen where you put an executing Process in the Hold TCQ. See Controlling Processes with Commands for information.

The following options enable you to view your IBM Connect:Direct environment:

Option	Description
MSG	Displays the Message Maintenance screen where you can display IBM Connect:Direct messages or print a summary or full report of the IBM Connect:Direct message file. See Viewing and Printing Messages Using Different Criteria for more information on the Message Maintenance screen.
SD	Displays the Signon Defaults screen where you can examine and change your signon default values. See Signing On to Connect:Direct for z/OS for more information on the Signon Defaults screen.
NM	Displays the Select network map screen, where you choose for display or print the defined nodes from the network map file and translate TCP/IP host names and network addresses.
WHO	Displays the User Information screen, which indicates your user ID, user node, version, release, maintenance level of IBM Connect:Direct and file information about your current session. See Signing On to Connect:Direct for z/OS for more information on viewing current sign on parameters.
AUTH	Displays the User Authorization screen. This screen displays commands you are authorized to use. See Displaying Your User Authorization for more information.

The following options enable you to sign on and swap among multiple sessions:

Option	Description
MS	Displays the Multiple Session Signon screen where you sign on to another DTF session. See page 46 for information.

Option	Description
SW	Displays the Swap/Display User Sessions screen where you view all the multiple sessions that you are signed on to and swap sessions. See Signing On to Multiple Sessions for more information on the Multiple Session Signon screen.

The following options display other menus:

Option	Description
SPF	Displays an ISPF/PDF session, one level beneath IBM Connect:Direct panels. See Navigating the Interactive User Interface for information on branching to an ISPF/PDF session.
ADMIN	Displays the Connect:Direct Administrative Options Menu with command options used for administering the system.

Navigating the Interactive User Interface

You can move from screen to screen with branch commands and function keys.

Using Branch Commands

Branch commands enable you to navigate quickly to a specific screen without returning to the Menu. You can type the branch command on the command line at the top of the screen (CMD==>) or on any input line preceded by ==>. To execute the branch, type = followed by the ID of the screen you want to branch to and press **Enter**.

For example, to go to the Select Process (SP) screen, type **=SP** on the command line and press **Enter**.

Note: The equal sign (=) is not required before an option on the command line of the Signon screen or the Primary Options Menu.

The following table describes two additional branch commands that you can use.

Command	Description
=ADMIN.xx where xx is the screen ID	Enables you to move to an administrative screen from a primary screen. For example, to branch from the WHO screen (which is an option of the Primary Menu) to the INQ screen (which is an option of the Administrative Menu), type =ADMIN.INQ at the command line of the WHO screen.
SPF.x where x is the SPF option that you want to display	The SPF option on the Primary Options Menu branches to the ISPF/PDF session, one level beneath the IBM Connect:Direct panels. If you want to bypass the ISPF/PDF menu, type the SPF command with a period and the option. For example, to use the BROWSE function within ISPF/PDF, type SPF.1 on the command line. The SPF command is valid on each IBM Connect:Direct screen command line.

Using the PF Keys

The PF keys listed in the following table are some of the programmed default definitions that you can use with IBM Connect:Direct screens. The ISPF user-defined PF keys are supported.

Key	Result
PF1	Displays online field Help or long error message
PF3	Returns to the previous screen
PF4	Returns to the Primary Options Menu
PF7	Scrolls up

Key	Result
PF8	Scrolls down
PF10	Scrolls right
PF11	Scrolls left

Browsing the IUI Display

The TABLE or DISPLAY parameters which are options at the bottom of most Selection screens specify that the output of the selection command is stored in the IBM Connect:Direct temporary file and then displayed upon successful completion of the command. The output is usually in tabular format. Use the commands, UP, RIGHT, or FIND, to browse the output. Press the **PF1** key to see extensive Help on how to browse the file.

Displaying Your User Authorization

Use the User Authorization screen to determine commands you are authorized to execute. To access the screen, select the **AUTH** option from the Primary Options Menu.

CD.ART CMD ==>		USER AUTHORIZATION		13:43	
AUTH COMMAND		AUTH COMMAND			
-----		-----			
1)	YES - CHANGE PROCESS	15)	YES - SELECT TASK		
2)	YES - DELETE PROCESS	16)	YES - SELECT TYPE		
3)	YES - DELETE TYPE	17)	YES - SELECT USER		
4)	YES - DELETE USER	18)	YES - SUBMIT PROCESS		
5)	YES - FLUSH PROCESS	19)	YES - SUBMIT WITHIN PROC		
6)	YES - FLUSH TASK	20)	YES - SUSPEND PROCESS		
7)	Y/Y - INSERT/UPDATE TYPE	21)	YES - STAT COMMAND		
8)	Y/Y - INSERT/UPDATE USER	22)	YES - EVENT COMMAND		
9)	YES - MODIFY (TRACE)	23)	YES - VIEW PROCESS		
10)	YES - STOP CONNECT:DIRECT	24)	YES - PERFORM CRC OVERRIDES		
11)	YES - UPDATE NETWORK MAP	25)	NO - CONFIRM DELETE		
12)	YES - SELECT NETWORK MAP	26)	YES - CONFIRM DEL OFF		
13)	YES - SELECT PROCESS	27)	Y/Y - SECURE+ ADMIN/CMDS		
14)	YES - SELECT STATISTICS	28)	YES - UPDATE INITPARM		

YES or Y means you are authorized, NO or N means you are not authorized, and SUB means you are authorized only if you submitted the Process.

To branch to an authorized function, type the corresponding number on the command line or cursor-select an enabled **AUTH** field, a field under the AUTH column.

Getting Help

IBM Connect:Direct provides a Help facility which describes the parameters that you can type through the IUI.

Procedure

- Access the Help information by typing **HELP** on the command line and press **Enter** or pressing the **PF1** key.
- Press **Enter** to continue viewing the Help screens.
- When you have viewed the screens, you can either press **PF1** to continue to the tutorial or press **PF3** to return to the screen you branched from.

Operator Tables

The Operator Table is available only under the IUI. This display enables you to monitor and control Processes. It is accessed through the SELECT PROCESS command and provides a summary of PNODE Processes in the TCQ. See [Viewing and Controlling a Process through the Operator Table](#) for a description and samples of the Operator Table.

node.name				Row 1 to 6 of 6		
-----OPERATOR TABLE-----						
==> Q					SCROLL ==>	PAGE
OPTION	PNAME	PNUMBER	SUBMITTER.NODE--	OTHER.NODE-----	QUEUE	
	SERVER	USERID				

	SEND100M	18	CD.NODE1	CD.STD45	EX	EX
	USER01					
	SEND50K	20	CD.NODE2	10.20.129.151	-P	EX
	USER01					
	COPYPAY	25	CD.NODE3	CD.STD44	-P	EX
	USER01					
	BKUPDSK1	2	CD.NODE4	10.20.129.141	-P	EX
	USER01					
	BENCH50	13,218	CD.NODE5	CD.STD45	-P	EX
	USER01					
	COPYCF	21	CD.NODE6	CD.STD45	HO	HI
***** Bottom of data *****						
	USER01					

You can press **Enter** to refresh the display and monitor the progress of the Processes displayed on the screen. The following table describes the commands that you can issue from the Operator Table.

Command	Description
H	Hold
D	Delete
R	Release
P	Suspend
F	Flush
V	View Processes
S	Show Detail

Type the option you want in the OPTION column next to the name of the Process as shown in the previous screen.

Note: The Executing Queue version of the Operator Table displays the executing Processes only; therefore, the options available to it are limited to the Suspend (P) and Show detail (S) only.

Messages

You may encounter two types of messages: ISPF/PDF and IBM Connect:Direct messages.

Message Type	Description
ISPF/PDF messages	Display at the top of the screen when the value you typed for a field does not pass the editor. You get this kind of message if, for example, you type an alphabetic character in a numeric field. Press PF1 to see a longer explanation of the error or type HELP on the command line and press Enter .

Message Type	Description
IBM Connect:Direct messages and sense codes	<p>Contain a short and long text which explain an error you get when you submit your command. This type of error is indicated by a message ID with the format xxxnnnx, for example, SVSL003I. You get this error, if, for example, you type an invalid Process name on a SELECT PROCESS command.</p> <p>IBM Connect:Direct generates a sense code to indicate a connection error involving a netmap. The format for a sense code is SENSnnnn where nnnn is the four-digit number representing the sense code.</p>

Message Example

In the following example, the user typed ABC for Process Number, a numeric field, resulting in the message PROCESS NUMBER in the upper right corner. Pressing **PF1** produces the longer message "ABC" IS INVALID...

node.name	SELECT PROCESS	PROCESS NUMBER
CMD ==> 0		hh:mm
"ABC" IS INVALID; MUST BE IN THE RANGE 0 - 199,999.		
CMD: 0 ... OPERATOR TABLE S ... OPERATOR TABLE/EXEC QUEUE STATUS		
P ... PRINT REPORT D ... DISPLAY REPORT V ... VIEW PROCESS		
QUEUE: ==> _ (A-ALL,W-WAIT,E-EXECUTE,H-HOLD,T-TIMER)		
PROCESS NUMBERS:	==> -----	==> -----
PROCESS NAMES:	==> -----	==> -----
SERVER NAMES:	==> -----	==> -----
STATUS:	(H0,HR,HI,HE,HC,HP,HS,RH,RA,WC,WX,WT,H,R,W)	
	==> --	==> --
DESTINATION NODES:	==> -----	==> -----
	==> -----	==> -----
USER ID:	NODE ID:	
==> -----	==> -----	
==> -----	==> -----	
	==> -----	
DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> NO		

The following is an of the short text message that is displayed at the top of the screen.

No process(es) found matching the search criteria.

The long text provides a more detailed explanation of the message and can also include the system action and a suggested response. Press **PF1** to see the long explanation of the error or type **HELP** or **M** on the command line and press **Enter**. The following figure shows a sample message.

node.name	Connect:Direct MESSAGE DISPLAY	hh:mm
CMD ==>		
MSGID ==>	SVSL003I	
MODULE ==>	DMVSOPEN	
Copy requested DISP=(,CATLG) to already cataloged dataset.		
The PROCESS COPY step requested a DISP=(,CATLG) on the TO		
clause of the COPY statement. The requested dataset already		
exists as a cataloged dataset.		
System Action. The PROCESS COPY step is terminated with a		
completion code of 8.		
Response: Either correct the COPY dataset or uncatalog the		
existing dataset and re-submit the PROCESS.		

The Statistics Summary screen, an option of the SELECT STATISTICS command, also gives you the option of branching to the message screen associated with a failed Process. Type an **M** next to the Process name which failed (indicated by an *), and IBM Connect:Direct displays the extended message.

Messages in a IBM Connect:Direct/Plex Environment

In a IBM Connect:Direct/Plex environment, messages can originate from any IBM Connect:Direct Server or from the IBM Connect:Direct Manager. The system administrator can define a 2-character message ID that identifies which IBM Connect:Direct/Plex member originated the message. This message ID is displayed after the message number, as in the following example. The message ID is highlighted in bold in this example.

```
SVTM055I S1 SESSION (001) ESTABLISHED WITH      SNODE=SC.DUB.OS390
SVTM055I S1 SESSION (001) ESTABLISHED WITH      PNODE=SC.DUB.OS390
SVTM036I S1 PROCESS STARTED      MVS2MVST(      1) PNODE=SC.DUB.OS390
SVTM036I S1 PROCESS STARTED      MVSM2VST(      1) SNODE=SC.DUB.OS390
```

The system administrator defines the message ID using the CDPLEX.MSGID initialization parameter. For more information, see *Customizing IBM Connect:Direct* in the *IBM Connect:Direct for z/OS Administration Guide* and search for .

Viewing and Printing Messages Using Different Criteria

About this task

You can use the Message Maintenance screen to view or print IBM Connect:Direct Message Text. You can display messages by message ID if you need to look up an error message in your Process statistics, or if you have a batch job that failed.

Procedure

1. Type **DIR** at the COMMAND prompt at the bottom of the screen to display the directory of the module messages.
2. From the Primary Options Menu, select option **MSG**.

The MSG command is available only through the UII. The following sample screen is a combination of two panels to show the prefixes used in IBM Connect:Direct messages:

```
Connect:Direct MESSAGE LOOKUP      2 Members processed      DATE => yyyy/mm/mm
ENTER OPTION ==> _      TIME => hh:mm
=====
                                OPTIONS
                                =====
1 = DISPLAY MESSAGE TEXT BY MESSAGE ID
2 = DISPLAY LIST OF ENTRIES (OPTIONALLY BY MODULE)
  MODULE==> _____
3 = PRINT SUMMARY REPORT
4 = PRINT FULL REPORT
The following is the general basis for C:D messages:
  MAINFRAME: z/OS, VM, VSE, Security (ACF2, RACF, TSS), etc.
MESSAGE
PREFIX      Applies to:
N xxx       CICS API failures
RACF xxxxx  Security system MSGs
SENS nnnn   NETMAP sense code
CSPx xxxxx  Secure Plus
NON-Mainframe
MESSAGE
PREFIX      Applies to:
A xxxxxxxx  i5/OS or OS/400
EMLD xxxxx  HP NonStop or Tandem
IOXT xxxxx  HP NonStop or Tandem
L xxxxxxxx  Windows, SI, or Select
X xxxxxxxx  Linux, UNIX, or Stratus
MESSAGE
PREFIX      Applies to:
Q xxx       CICS API failures
S xxxxxxxx  C:D Mainframe msgs
U nnnn      z/OS User ABEND code
MBCS xxxxx  Multi-Byte Character Set
MESSAGE
PREFIX      Applies to:
CCUP xxxxx  Control Center
FSLD xxxxx  HP NonStop or Tandem
J xxxxxxxx  C:D for the Web
V xxxxxxxx  OpenVMS
```

3. Type **1** on the command line and press **Enter**.
The Message Display Screen appears.

4. Type the message ID and press **Enter**. (To check a sense code, type SENSnnnn where nnnn is the four-digit number representing the sense code and press Enter.) The full text of the message (or sense code) is displayed. In the following example, the information associated with Message SCBI190I is shown.

```
Connect:Direct MESSAGE DISPLAY

MESSAGE ID==> SCBI190I
MODULE      ==> DMCBSUBM
=====
SHORT TEXT==> Process specified not in process library.
LONG TEXT:
  LINE 1 ==> This message may be generated for the following reasons:
  LINE 2 ==>
  LINE 3 ==> 1) The member specified is not in the public process library
  LINE 4 ==> or in the PDS specified in the 'DSN=' keyword.
  LINE 5 ==> 2) The process library has not been properly concatenated.
  LINE 6 ==>
  LINE 7 ==>
  LINE 8 ==>
  LINE 9 ==>
  LINE 10==> SYSTEM ACTION: Return to invoker with RC=8.
  LINE 11==>
  LINE 12==> RESPONSE: Verify the above is correct and resubmit process.
COMMAND    ==> ----- ENTER 'DIR' TO DISPLAY THE DIRECTORY
```

Viewing a Message List

You can display a list of all IBM Connect:Direct messages or a list of messages generated by a module.

Procedure

1. From the Primary Options Menu, select option **MSG**.

Note: The MSG command is available only through the IUI.

2. Type **2** on the command line, specify a Module (optional), and press **Enter**. The Message Summary screen appears. (If you do not specify a module name, all messages are displayed.)

In the sample below, the module, DMINIT2, was specified.

```
-----Connect:Direct MESSAGE SUMMARY----- Row 1 of 31
CMD ==> SCROLL ==> PAGE
MSG ID  MODULE  MESSAGE
CSPN010E DMINIT2 SIGNATURE.DSN.BASE Initialization Parameter Invalid
CSPN011E DMINIT2 SIGNATURE.FILE.PAIRS Initialization Parameter Invalid
CSPN012E DMINIT2 SIGNATURE.DSN.BASE and/or SIGNATURE.FILE.PAIRS missing
CSPN013E DMINIT2 SECURE+ Interface Module not found
CSPN116E DMINIT2 PARM FILE VALIDATION FAILURE
CSPN117E DMINIT2 SECURE.DSN IS NOT A VSAM KSDS
CSPN118E DMINIT2 SECURE+ INTERFACE MODULE NOT FOUND
SCFS006I DMINIT2 AUTOLOG OF CMS SERVER FAILED
SITA137E DMINIT2 POSIX Init Parm conflict.
SITA169I DMINIT2 SECURE.DSN Data set not found.
SITA170I DMINIT2 TRACE.BUFFER outside 0-999 range, TRACE.BUFFER=2 assumed.
SITA185I DMINIT2 SITA185I (VSE) TCP.NAME VALUE OVERRIDEN BY SYS Parm.
SITA190I DMINIT2 Secure+ Initialization failed, Secure No Override No
SITA278I DMINIT2 All nodes are quiesced, QUIESCE=YES specified
SITA338I DMINIT2 Asset Protection Failure.
SITA339I DMINIT2 Number of licensed copies has been exceeded.
SITA604I DMINIT2 Invalid option specified for SNA paramater.
SITA607I DMINIT2 SNA=YES but no VTAM APPLID is available.
SITA609I DMINIT2 VTAM services disabled.
SITA610I DMINIT2 Both SNA and TCP are NO, can not continue.
SITA641W DMINIT2 Level of PDSE.SHARING is not supported.
```

Note: You must specify the “old” modules names used in versions prior to the 5.1 release of Connect:Direct for z/OS. For information, go to *IBM Connect:Direct for z/OS Release Notes* and search for *Upgrading to Connect:Direct for z/OS Version 5.1*.

3. Type an **S** next to the message ID for the full **Message Display** screen.

Printing Messages

You can print the IBM Connect:Direct messages file in summary form or in detail, including the long form of the message text.

Procedure

1. From the **Primary Options Menu**, select option **MSG**.

Note: The MSG command is available only through the IUI.

2. Do one of the following:

- Type **3** on the command line for a summary report, and press **Enter**.
- Type **4** on the command line for a detail report, and press **Enter**.

Specify the sysout class for the message file print output and the ID for a remote printer. If you want your output to go to a preallocated data set instead, specify the name of the data set. You must enclose the data set name in single quotes and give it the following DCB attributes.

```
DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=0
```

The Application Program Interface

The high-level application-program, DGADCHLA, handles communication between the Connect:Direct for z/OS API and any application program that provides IBM Connect:Direct with command strings for batch processing. You can write an application program following the rules described in the following sections.

You can also process IBM Connect:Direct command strings by using the DGADBATC batch interface program. For more information on using DGADBATC, refer to [Sample Job Stream to Run the Batch Interface](#).

Both DGADCHLA and DGADBATC are in \$CD.SDGALINK.

You can design user-written applications that have an interface to IBM Connect:Direct. You can write the applications in any computer language, including PL/I, Assembler, and COBOL. The following samples are in the \$CD.SDGASAMP library:

Application Name	Description
DGAXSAMP	Assembler language user-written application
DGAXPLIS	PL/I language user-written program
DGAXCOBS	COBOL language user-written program

Important: All API programs run in 31-bit mode and they must call DGADCHLA in 31-bit mode. You can either link your program as AMODE 31, or establish 31-bit addressability before calling DGADCHLA. See the sample program DGAXSAMP for an example of establishing 31-bit addressability and returning to 24-bit addressability after the call to DGADCHLA. Also execute DGADCHLA with a BASSM rather than a BALR. Macro GENCALL is provided in the sample library to perform a BASSM. See the sample program DGAXSAMP for an example.



Attention: When calling DGADCHLA from an application that uses multiple subtasks to invoke the API simultaneously, it is recommended that an APIINIT call be made from the application main task or a permanent subtask prior to attaching the API subtasks. The APIINIT call will anchor and initialize DCB storage for standard print/trace DDs used by all API subtasks. This will prevent possible ABEND conditions caused by CLOSE or I/O requests from subtasks that did not initially OPEN those DCBs. APIINIT should only be issued once by the application and will cause SCIB025I messages to be written to the DDs identified in SDGAMAC member DGA\$SYAT, if they are present. APIINIT is specified as the ninth parameter and all other parameters will be ignored for that call.

DGADCHLA Program

The DGADCHLA program communicates with the API through a control block interface called the User Interface Control Block (UICB).

DGADCHLA works in the following sequence:

1. Optional APIINIT call to anchor print/trace DCB storage when API can be called from multiple subtasks in calling application. This is a one time call from the application main task or a permanent subtask prior to attaching other subtasks that perform API calls. APIINIT is passed in the ninth parameter position.
2. DGADCHLA accepts IBM Connect:Direct command strings from an application program and passes the strings to the API.
3. The user-written application requests DGADCHLA to perform output formatting routines after returning from the API. These formatting routines display information about the IBM Connect:Direct command that just completed processing.
4. After execution of each command, DGADCHLA issues a return code reflecting the status of API communications.

Required Parameters for DGADCHLA

DGADCHLA requires IBM Connect:Direct to pass three parameters on every invocation from an application program.

The parameters, in order, are the following:

Parameter	Description						
CMDLEN (first parameter)	<p>Points to a variable-length character string (up to 4096 bytes) that contains the string length in the first halfword and the command text in the remainder of the string. Just as in DGADBATC, the first command string must be a SIGNON and the last command must be a SIGNOFF. The following figure shows the command string format.</p> <div><table><tr><th>Length of Command String</th><th>Command String</th></tr><tr><td>H'32'</td><td>SELECT PROC WHERE(PNUM=02)</td></tr><tr><td colspan="2">132.... 32</td></tr></table></div>	Length of Command String	Command String	H'32'	SELECT PROC WHERE(PNUM=02)	132.... 32	
Length of Command String	Command String						
H'32'	SELECT PROC WHERE(PNUM=02)						
132.... 32							
UICB@ (second parameter)	<p>Points to a fullword pointer of zeroes, which is filled in with the UICB address by DGADCHLA when DGADCHLA receives the SIGNON command from the application program. Information in the UICB can now be interrogated by the application program, if necessary. The UICB@ field is cleared and filled again with zeroes when DGADCHLA receives the SIGNOFF command.</p>						
OUTSPECS (third parameter)	<p>Points to a 7-character string containing the output format specifications. The output is written to the file name defined by DMPRINT. Each specification is one character long.</p> <p>All output generated as a result of these specifications is routed to a DDNAME of DMPRINT. No output is created if the DDNAME is not present.</p>						

The following table describes the OUTSPECS output format specifications:

Field	Values	Meaning
1	Y	Displays the command string that executed.
	N	Does not display the command string that executed.
2	Y	Displays the API return code and message ID.
	N	Does not display the API return code and message ID.
3	S	Displays the short message text when a return code of zero comes from the API.
	L	Displays long message text when a return code of zero is received from the API.
	N	Does not display message text when a return code of zero is received from the API.
4	S	Displays the short message text when a nonzero return code comes from the API.
	L	Displays long message text when a nonzero return code comes from the API.
	N	Does not display message text when a return code greater than zero comes from the API.
5	Y	Displays the data generated in the temporary file.
	N	Does not display the data generated in the temporary file.
6	Y	Displays the string that identifies UICB fields which the Extract feature returns information about.
	N	Does not display the string that identifies UICB fields which the Extract feature returns information about.
7	Y	Displays a dump of the area that received output from the Extract feature.
	N	Does not display a dump of the area that received output from the Extract feature.

Field	Values	Meaning
8	Y	If parameter 9 is specified as N, this parameters prints output from a SELECT STATS command to DMPRINT when a MAXDELAY Process fails with a return code other than 52.
9	Y	Enables monitoring of the MAXDELAY Restart Process. Produces informational messages about monitoring and about the submitted and monitored Process number. For more information about MAXDELAY restart, see <i>Using the MAXDELAY keyword parameter to synchronize submitted Processes</i>
	N	Allows the MAXDELAY Process to execute without restart. The DGADBATC step is suspended until the Process completes execution.
	APIINIT	Optional parameter used on first invocation to anchor DCB storage for print/trace output. All other parameters are ignored.
10 Applies to MAXDELAY Restart only.	Y	If parameter 9 is specified as Y, the MAXDELAY progress is processed on a status report to the DMPRINT DD using the Select Process command. Note: Using this parameter may result in a large amount of data for DMPRINT DD.
	N	Suppresses the progress report.
11 Applies to MAXDELAY and MAXDELAY Restart.	Y	Forces the DGADBATC step to terminate on the first return code of 8 or greater.
	N	Allows the DGADBATC step to continue to process the commands in SYSIN even after a non-zero return code.

The following example shows the most common specifications for this parameter.

```
OUTSPECS DC C'YYSLYNNNNNN'
```

Positions 6 and 7 are not used by DGADCHLA unless the extract feature of DGADCHLA is used, as explained in the next section.

Extracting Return Codes

If you want the Return Code from the processed IBM Connect:Direct command, you must use the extract feature. This feature enables you to extract certain UICB fields after IBM Connect:Direct command execution.

To activate the optional extract feature, the program must pass the following additional parameters to DGADCHLA:

Note:

DGADCHLA specifies either the first three required parameters or all eight parameters which include these parameters related to return codes. If an incorrect number of parameters is passed, DGADCHLA issues an error message and assigns a return code of 20. Processing cannot occur during this time.

Parameter	Description
EXTSTRLN (fourth parameter)	<p>The UICB extract string. The user-supplied application must set up an extract string identifying the UICB fields to extract. The string consists of a halfword length field containing the length of the extract string (excluding the itself halfword length field), followed by the first UICB field to extract, a space, the second UICB field to extract, a space, and so on.</p> <p>A sample extract string follows:</p> <div><div>H'32'</div><div>UITMPDDN UIRTNCD UIPROC# UIMSGID</div></div> <p>You can only specify the UICB fields listed in the extract string. If you encounter an invalid keyword in the string, the extract routine terminates execution, and no more information is extracted. The application is informed of the error by means of the EXTRC, EXTMSG, and INVALIDKEY parameters. If field 6 of the OUTSPECS parameter is set to Y, the extract string is written to the file defined by DMPRINT.</p>
EXTAREA (fifth parameter)	<p>Where the UICB extracted information is placed. The application is responsible for ensuring that the area is large enough to accommodate the information requested by the extract string. If field 7 of OUTSPECS is set to Y, the extract string is written to the file defined by DMPRINT. The following shows how to calculate required storage for the extract area necessary for the extract string in the figure shown for the EXTSTRLN parameter.</p>
EXTRC (sixth parameter)	<p>A 4-byte binary field containing the extract feature return code.</p>
EXTMSG (seventh parameter)	<p>An 8-character field containing an 8-character message ID from the extract feature.</p>
INVALIDKEY (eighth parameter)	<p>An 8-character field containing an invalid extract string keyword. DGADCHLA relays the location of the extract string error to the application in this parameter. This field contains the invalid item in the extract string if EXTRC is nonzero.</p>

The following table shows how to calculate required storage for the extract area necessary for the extract string in the figure shown for the EXTSTRLN parameter:

UICB Field	Data Definitions	Storage
UITMPDDN	Character length of 8	8 bytes
UIRTNCD	Fullword	4 bytes
UIPROC#	Fullword	4 bytes
UIMSGID	Character length of 8	8 bytes

UICB Fields

The following table shows the valid UICB fields. Specifying these fields in the EXTSTRLN parameter results in feedback in the extract area, defined by the EXTAREA parameter. Exact definitions are in the DGA\$UICB macro found in \$CD.SDGAMAC. In the table, the Control Block Builder Syntax Error Work Areas have a value only when appropriate. For the Boolean Flags, output is Y (bit is on) or N (bit is off).

Name	Type	LN	Description
UIRCBLNG	halfword	2	The length of the control block
UIDESCR	character	16	UICB identifier
UITCA	address constant	4	Task Control Area
UIBRCB	address constant	4	Batch Region Control Block
UIDYNCB	address constant	4	Dynamic Allocation Control Block
UITPCB	address constant	4	Text Parser Control Block
UILEVEL	fullword	4	Modal level counter
UITMPDCB	address constant	4	Temporary file DCB
UIMSGCB	address constant	4	Message Control Block
UIUNODE	character	16	User node ID
UIUID	character	64	User ID
UIPSWD	character	64	Signon password
UINPSWD	character	64	New Signon password
UITMPVOL	character	6	Volume serial number of temporary file if not specified by user
UNTMPDDN	character	8	DDNAME used for temporary DSN
UIUSRTYP	character	1	User Type (operator, administrator, user)
UIAPPLID	character	8	VTAM logon ID
UIRTNCD	fullword	4	API return code
UIMSGID	character	8	API message ID
UILNODE	character	16	Name of node that is "local" to DTF
UILPP	halfword	2	Lines per page for printed output
UITMPLNG	halfword	2	Length of TSO temporary file name
UITMPDSN	character	44	TSO temporary file name
UIPUBLNG	halfword	2	Length of Process Library name
UIPUBDSN	character	44	Process Library name
UIMSGLNG	halfword	2	Length of message library name
UIMSGDSN	character	44	Message library name
UINETMAP	character	64	Network map file name
UIPROC#	fullword	4	Process number from submit
UIDSPLY	address constant	4	Address of SCDSPLY
UISTRING	address constant	4	Address of SCSTRING

Name	Type	LN	Description
***** *	*****		Start of control block builder syntax error work area
UILABL#	halfword	2	Length of IBM Connect:Direct label
UILABL	character	8	IBM Connect:Direct label
UICMD1#	halfword	2	Length of first word in IBM Connect:Direct command
UICMD1	character	8	First word in IBM Connect:Direct command
UICMD2#	halfword	2	Length of second word in IBM Connect:Direct command
UICMD2	character	8	Second word in IBM Connect:Direct command
UIKLST#	halfword	2	Length of keyword that starts a list
UIKLST	character	8	Keyword that starts a list
UIKEYW#	fullword	4	Length of keyword in list before error
UIKEYW	character	8	Last keyword in list before an error
UIPARM#	halfword	2	Length of parameter associated with UIKEYW
UIPARM	character	8	Parameter in error associated with UIKEYW
UIERRM1#	halfword	2	Length of UIERRM1 string
UIERRM1	character	64	All of the above work areas resolved into a string
UIERRM2	character	64	Msg ID and text for parsing error
***** *	*****		Start of boolean flags
UIERRON	character	1	Indicates message in UIERRM1
UIERRLAB	character	1	Indicates something in UILABL
UIERRCM1	character	1	Indicates something in UICMD1
UIERRCM2	character	1	Indicates something in UICMD2
UIERRSCP	character	1	Indicates parsing error
UIERRC1O	character	1	Indicates open delimiter after command keyword, for example, "IF ("
UIERRC2O	character	1	Indicates open delimiter after second command keyword
UIERRG	character	1	Indicates VTAM error msg in UIERRM1
UIERRLST	character	1	Indicates something in UIKLST
UIERRLOP	character	1	Indicates open delimiter after UIKLST
UIERRLCL	character	1	Indicates close delimiter after UIKLST
UIERRLEQ	character	1	Indicates equal sign after UIKLST
UIERRLCM	character	1	Indicates comma after UIKLST
UIERRLSP	character	1	Indicates space after UIKLST
UIERRKEY	character	1	Indicates something in UIKEYW
UIERRKOP	character	1	Indicates open delimiter after UIKEYW

Name	Type	LN	Description
UIERRKCL	character	1	Indicates close delimiter after UIKEYW
UIERRKEQ	character	1	Indicates equal sign after UIKEYW
UIERRKCM	character	1	Indicates comma after UIKEYW
UIERRKSP	character	1	Indicates space after UIKEYW
UIERRBPC	character	1	Indicates close delimiter before a parameter in a list
UIERRPRM	character	1	Indicates something in UIPARM
UIERRPOP	character	1	Indicates open delimiter after UIPARM
UIERRPCL	character	1	Indicates close delimiter after UIPARM
UIERRPEQ	character	1	Indicates equal sign after UIPARM
UIERRPCM	character	1	Indicates comma after UIPARM
UIERRPSP	character	1	Indicates space after UIPARM
UIF1SUBM	character	1	Indicates a submitted Process
UIGOTDSN	character	1	Indicates found temporary file name as Signon command parameter
UITFILE	character	1	Indicates data is generated into temporary file
UIEOF	character	1	Indicates reached EOF of Process file
UIMODAL	character	1	Indicates modal statement processed
UITFILEX	character	1	Reserved
UIMASTER	character	1	Indicates this UICB is master
UIRECON	character	1	Indicates a reconnect attempted
UIINACT	character	1	Indicates VTAM session for this UICB failed
UILOCAL	character	1	Indicates local node session
UIESF	character	1	Indicates ESF=YES on Signon command
UIZOPSWD	character	1	Indicates blank password on Signon
UIZNPSWD	character	1	Indicates blank new password on Signon
***** *	*****		END OF BOOLEAN FLAGS
UIRAT@	address constant	4	PTR to resource address table
UIDRLSE#	halfword	2	DTF release level
UIDPUF#	halfword	2	DTF PUF level
UIDPUT#	halfword	2	PUT tape number
UILINE#	halfword	2	Line number within Process in error
UIRLSE#	halfword	2	Current release, version and mod level
UIPUF#	halfword	2	Current PUF level
UI@MASTR	address constant	4	Master (user) UICB
UI@ACTIV	address constant	4	Currently active UICB

Name	Type	LN	Description
UI@FPTR	address constant	4	Next UICB
UI@BPTR	address constant	4	Previous UICB
UIALOTYP	character	8	Allocation type for temporary file
UIALOPRI	character	8	Allocation of prime space for temporary file
UIALOSEC	character	8	Allocation of secondary space for temporary file
UIALOUNI	character	8	Allocation unit for temporary file
UIALOVOL	character	8	Allocation of volume serial number for temporary file
UIFOLD	character	3	Fold to upper case if YES UIPRTALC
UIPRTALC	character	80	User-defined IBM Connect:Direct print destination
UIPACCT#	halfword	2	Length of PNODE accounting data
UIPACCT	character	255	PNODE accounting data text
UISACCT#	halfword	2	Length of SNODE accounting data
UISACCT	character	255	SNODE accounting data text
UIGSCAPI (VM)	character	8	GCS API virtual machine identifier

DGADCHLA Return Codes

Return codes reflect the status of the DGADCHLA communications with the API. These codes only reflect whether DGADCHLA could process the request and pass the command to the DTF. They do not reflect the completion status of the command.

The following table defines each return code.

RC	Description
00000000	The command executed normally.
00000004	Signon to the master session failed, but the ESF environment is established.
00000008	A non-ESF command attempted in an ESF environment.
0000000C	A session is lost in a multiple session environment.
00000010	The master session is lost.
00000014	The master session signed off successfully.
00000018	The master session signon failed, and no ESF ability exists.
0000001C	A non-master signon failed.
00000020	DGADCHLA received an invalid number of input parameters.
00000024	The output specifications included an invalid parameter.
00000028	An invalid pointer to the UICB is passed to DGADCHLA.
00000034	MAXDELAY=0 Process never executed and was removed from the Process queue.

For more information on master sessions and multiple session environments, refer to [Signing On to Connect:Direct for z/OS](#).

Extract Fields for DGADCHLA Return Codes

The following extract fields apply to DGADCHLA return codes:

Field	Description
UIRTNCD	Lists return codes. UIRTNCD is set on completion of every command processed by the DTF.
UIMSGID	Lists message IDs. UIMSGID is set on completion of every command processed by the DTF.
UIPROC#	Identifies Process numbers. UIPROC# is set for every successful Submit command.

See [UICB Fields](#) for the complete list of UICB fields.

Important: When you are using DGACHLA within an application, ensure that for each SIGNON command a SIGNOFF executes successfully (RC=x'14') to avoid a storage creep within the application program.

DGADCHLA Examples

The following examples show how required and optional parameters appear in sample programs. There is also a sample job stream for executing the program.

Required Parameters and Calling Sequences

The following example shows how the required parameters and calling sequences appear in a sample Assembler program:

```
*****
*   THE DATA DEFINITIONS ARE AS FOLLOWS:   *
*****
CMDLEN      DS      H           LENGTH OF COMMAND STRING
CMD         DS      CL4096      COMMAND STRING
UICB@       DS      A           POINTER TO UICB
OUTSPECS    DC      C'YYSLYNN'  OUTPUT SPECIFICATIONS
*****
*                                           *
*   THE DGADCHLA INVOCATION WOULD BE:      *
*                                           *
*****
CALL  DGADCHLA, (CMDLEN, UICB@, OUTSPECS), VL
```

The following example shows how the required parameters and calling sequences appear in a sample PL/I program:

```
*****
*   THE DATA DECLARATIONS ARE AS FOLLOWS:   *
*****
DECLARE
CMD_PARM      FIXED BINARY (31)  BASED (R),
UICB_PARM     FIXED BINARY (31)  INITI (0'B'),
OUTSPEC_PARM  CHAR (7)  INIT ('YYSLYNN'),
CMD           CHAR (4096)  VARYING,
R             POINTER;
*****
*                                           *
*   THE DGADCHLA INVOCATION WOULD BE:      *
*                                           *
*****
R=ADDR(CMD) ;
CALL  DGADCHLA(CMD_PARM, UICB_PARM, OUTSPEC_PARM);
```

Optional Parameters

The following example shows how the optional parameters look in a sample Assembler program:

```

*****
**      THE PARAMETERS WOULD BE DEFINED AS FOLLOWS:      **
*****
CMDLEN   DS   H
CMD      DS   CL4096
UICB@    DS   A
OUTSPECS DC   C'YYSLYYY'
EXTSTRLN DC   H'32'
EXTSTR   DC   C'UITMPDDN UIESF UIPROC# UIMSGID'
EXTAREA  DS   OCL24      /*OUTPUT FROM THE UICB EXTRACT      */
EXTMPDDN DS   CL8        /*VALUE OF UITMPDDN      */
EXESF    DS   F          /*VALUE OF UIRTNCD      */
EXPROC#   DS   XL4        /*VALUE OF UIPROC#      */
EXMSGID   DS   CL8        /*VALUE OF UIMSGID      */
EXTRC     DS   F
EXTMSG    DS   CL8
INVALKEY  DS   CL8
*****
**      NOTE THAT THE EXTRACT STRING ITSELF IS 30 BYTES LONG,      *
**      BUT THE DUMP OF ITS EXTRACT WILL USE 31 BYTES.              *
**                                                                *
**      INSIDE THE MAIN BODY OF THE PROGRAM, THE CALL TO          *
**      DGADCHLA WOULD LOOK LIKE:                                  *
**                                                                *
*****
CALL  DGADCHLA,(CMDLEN,UICB@,OUTSPECS,EXTSTRLN,EXTAREA,
              EXTRC,EXTMSG,INVALKEY),VL

```

Sample Job Stream for Executing the Program

The following example shows a sample job stream that executes a IBM Connect:Direct program that invokes DGADCHLA.

```

//jobname JOB (ACCT),'PGMR NAME',
//      NOTIFY=TSOUER,TIME=(1),
//      MSGCLASS=X,CLASS=A,PRTY=9,REGION=1024K
//JOBLIB   DD   DISP=SHR,DSN=$CD.SDGALINK
//CD       EXEC  PGM=USERPROG
//DMPRINT   DD   SYSOUT=*
//DMNETMAP  DD   DISP=SHR,DSN=$CD.VSAM.NETMAP
//DMPUBLIB  DD   DISP=SHR,DSN=$CD.SDGAPROC
//DMMSGFIL  DD   DISP=SHR,DSN=$CD.VSAM.MESSAGE
//SYSUDUMP  DD   SYSOUT=*
//NDMCMD5   DD   SYSOUT=*

```

You must allocate several files using DD statements before executing a IBM Connect:Direct application program that calls DGADCHLA.

The following table defines the DD statements in the sample job stream:

Statement	Definition
DMPRINT	All output from SELECT PROCESS, SELECT STATISTICS, SELECT TYPE, and SELECT USER commands is written to this DD. Also, output as a result of the OUTSPECS specification is written to DMPRINT. DMPRINT is optional, but often required.
DMNETMAP	This definition is the name of the network map file that contains the names of all the nodes that IBM Connect:Direct communicates with. DMNETMAP is required if no NETMAP keyword is specified on the SIGNON command.
DMPUBLIB	This definition is a library containing Processes. It is required.
DMMSGFIL	This definition is the name of the Message file. It is required.
NDMCMD5	If allocated, all command strings are written to the current file. It is optional.

Managing Sessions

Signing On to Connect:Direct for z/OS

The SIGNON command establishes your session with the DTF so that you can issue commands.

You can control access to the IBM Connect:Direct system through the IBM Connect:Direct Authorization Facility and security exits. For information on security control, go to the *IBM Connect:Direct for z/OS Administration Guide* and search for Implementing Security.

If you are using the Extended Submit Facility (ESF), you can submit the SIGNON command even though the DTF is inactive.

Timeout Problems

If you are logged off involuntarily while using IBM Connect:Direct via TCP/IP, your problem may be related to the TCP/IP connection not being able to communicate with the DTF. If you see the message TCP/IP SEND error; Connection Lost, and press F1 to see the help for this message, the following text is displayed for message SVTC006I:

```
MSGID ==> SVTC006I
MODULE ==> DMTSTSND
TCP/IP SEND error; Connection Lost
An error has been detected while sending a command via TCP/IP.
The TCP/IP connection has been lost. A possible cause is
Connect:Direct has been shutdown and all TCP/IP connections
have been closed.
System Action: Normal processing can not continue.
Response: Logoff and attempt to signon again.
```

Although the TCP/IP protocol does not have a mechanism to send you a console message indicating that the client has been dropped and your session has timed out, you can determine if this has happened to you by looking in the statistics file for a signoff record with an SAFA019I error message that matches your user ID (be sure to enter Y in the CHANGE EXTENDED OPTS field on the SELECT STATISTICS panel or use the ADVANCED SELECT STATISTICS panel, and then enter SO as a record type on the next screen). For more information, go to the *IBM Connect:Direct for z/OS Administration Guide* and search for Signon and IUI/API Errors.

IBM Connect:Direct provides an inactivity timer that is based on the number of minutes a user can remain inactive without communicating with the DTF. An administrator can specify this value using the TCP.API.TIMER global initialization parameter. IBM Connect:Direct will only recognize changes made to the TIMEOUT parameter by an administrator. If you attempt to change this parameter and you are not an administrator, you will not be able to sign on. You can determine if this has happened to you by looking in the statistics file for a signon record with an SAFA020I error message that matches your user ID. For more information, go to the *IBM Connect:Direct for z/OS Administration Guide* and search for Signon and IUI/API Errors.

Note: The rationale behind the timeout feature is to do housekeeping to keep the number of hung sessions to a minimum and avoid the limit specifying the maximum number of users that can be signed on to IBM Connect:Direct. Without the timeout feature, an administrator would have to recycle the DTF to allow TCP/API logons to reconnect once the MAXUSER limit was reached.

Using SIGNON through the IUI

To execute the SIGNON command through the IUI, perform the following steps:

Procedure

1. Select IBM Connect:Direct from your ISPF/PDF Primary Option Menu. The Copyright and the Signon screen are displayed, unless SIGNON defaults are previously set.

```

CMD ==>
          IBM
Connect:Direct for z/OS
          VERSION      : 06
          RELEASE      : 000
          MODIFICATION  : 000
          LEVEL        : 0000
TODAY IS   : 2018/02/24 (2018.055 )      THE TIME IS: 09:01
                                           More:      +
USER ID => USER01
PASSWORD =>
NEW PASS =>
NODE NAME =>
TRANSPORT =>
COMMUNICATION ADDRESS =>(           ,           )
DESCRIPTION =>
TIMEOUT =>
Licensed Materials - Property of IBM
5655-X11 Copyright IBM Corp. 2013,2018
All Rights Reserved
IBM is a Trademark of International Business Machines

```

The Signon screen indicates the version, release, modification, and Program Update Tape (PUT) levels of IBM Connect:Direct.

2. Type your user ID and press **Enter**.

Depending upon your site configuration, you may also need to type in a password, a new password, node name, transport, communications address, and case sensitivity option before pressing **Enter**. These parameters are described in [“SIGNON Command”](#) on page 542, or press **PF1** for Help.

If you are using the Extended Submit Facility (ESF), the DTF does not need to be active to execute the SIGNON, SIGNOFF, and SUBMIT commands.

Note: If the Error requesting session, IBM Connect:Direct may be inactive message is displayed, the IBM Connect:Direct DTF is not started.

Results

If your signon is successful, the Primary Options Menu is displayed, unless you have updated your SIGNON defaults to execute an initial command.

Viewing Your Current Signon Parameters

You can view your signon default parameter settings.

Procedure

- To see how your signon parameters defaults are currently set, select the **WHO** option from the Primary Options Menu.

The following figure shows an example of the User Information display produced by the WHO option.

```

Connect:Direct IUI USER INFORMATION                               hh:mm
CMD ==>
  USER ID==> USER01
    USER NODE ==> node.name
    TRANSPORT PROTOCOL ==> NETMAP
    COMMUNICATION ADDRESS ==> APPLID
    TEMPORARY DSN DDNAME ==> TMPDD
    TEMPORARY DSN VOLSER ==> TMPVSER
    DATA SETS:
      TEMPORARY DSN ==> CD.TEMP.DSN
      MESSAGE DSN ==> CD.MSG.VER01
      NETWORK MAP DSN ==> CD.NETMAP.VER01
      PUBLIC PROCESS DSN ==> CD.PUBLIB
      IUI VERSION ==> VV
      IUI RELEASE ==> RRR
      IUI MODIFICATION ==> MMM
      IUI PUT LEVEL ==> PPPP
      DTF VERSION ==> V.RR.MM

```

Setting Up Signon Defaults

Use the Signon Defaults screen to set up signon default information.

Procedure

- Select **SD** from the Primary Options Menu to access the Signon Defaults screen.

```

node.name                SIGNON DEFAULTS                        hh:mm
CMD==>
  CURRENT DEFAULTS:
  USER ID ==>
  PASSWORD ==>
  TRANSPORT==>
  COMMUNICATION ADDRESS ==>(          ,          )
  DESCRIPTION ==>
  TIMEOUT ==> YES
  TEMPORARY DATA SET NAME ==>
  ALLOCATION TYPE ==>
  PRIMARY SPACE ==>
  SECONDARY SPACE ==>
  EXTENDED SUBMIT FEATURE ==>
  UPPER CASE FOR PRINT ==>
  PRINT FILE DESTINATION ==>
  PNODE ACCOUNTING DATA ==>
  SNODE ACCOUNTING DATA ==>
  DO YOU WANT ALL COMMANDS FOR THIS SESSION TO BE CASE SENSITIVE? ==> NO
  INITIAL COMMAND ==>
  UNIT TYPE ==>
  VOL=SER ==>

```

Note: Instead of users individually altering their signon defaults, the IBM Connect:Direct administrator can define global signon defaults. With this method, users will not be responsible for increasing their allocation and can also avoid SB37 ABENDs. For more information, refer to *Global Signon Defaults* in the *IBM Connect:Direct for z/OS Administration Guide*.

The INITIAL COMMAND field enables you to bypass the Primary Options Menu and have a designated command screen display automatically at signon. For example, to have the User Information screen (WHO) display automatically at signon, type **WHO** in the Initial Command field.

Command stacking is allowed. For example, type **ADMIN.ST** in the INITIAL COMMAND field to branch to the View Type Record (ST) option of the Administration Menu (ADMIN).


By typing your user ID and other information in the SIGNON Defaults panel, you can bypass the SIGNON panel.

SIGNON Command

The SIGNON command has the following format and associated parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SIGNON	NETMAP = network map data set name
		CASE= Yes <u>No</u>
		COMADDR= (Port number, IP address alias name)
		ESF= <u>Yes</u> No
		FOLD= Yes <u>No</u>
		NODE= node name
		PACCT= 'pnode accounting data'
		PRINT= destination of printed output
		RECONNECT
		SACCT= 'snode accounting data'
		SPACE= (<u>CYL</u> TRK blk,([prim],[sec]))
		TMPDD= preallocated data set ddname
		TMPDSN= preallocated data set name
		TRANSPORT= SNA TCP <u>NETmap</u>
		UNIT= temporary dsn unit type
		USERID= (ID, pswd, newpswd)
		VOLSER= volume serial
		DESCRIPTION= '30-character string'
		TIMEOUT= <u>Yes</u> No

Parameter Value	Description
NETMAP = network map data set name	<p>Specifies the network map data set name, which is a 1-44 alphanumeric character string with the first character alphabetic. This parameter is required. Ask your administrator for information about completing this value.</p> <p>If you provide the DMNETMAP DD name in the job stream used with DMBATCH, you are not required to include this parameter with the SIGNON command. If you specify the data set name in both places, the network map data set name you specify on the SIGNON command takes precedence.</p> <p>For the IUI, the network map data set name is specified in the ISPF/PDF Primary Option Menu as a parameter of the DMISTART program. This data set name is established by your IBM Connect:Direct administrator at installation and is not indicated through the IUI SIGNON command screens.</p>

Parameter Value	Description
CASE= Yes <u>No</u>	<p>The CASE parameter determines case sensitivity. It specifies whether parameters associated with accounting data, user ID, password, and data set name in the commands which follow the signon must be case sensitive. The default is No. This parameter is optional.</p> <p>You can override this designation on a command-by-command basis by adjusting the case sensitivity parameter on the individual command.</p>
COMADDR= (Port number, IP address alias name)	<p>Specifies the communications address used in TCP/IP connections to the DTF. This parameter is specified only if the TRANSPORT parameter is set to TCP.</p> <p>The port number specifies the port number used for TCP/IP communications. This value is the value from the TCP.API.LISTEN initialization parameter of the DTF that you want to sign on to.</p> <p>The IP address/alias name specifies the IP address or alias name used for TCP/IP communications.</p> <p>If you use the COMADDR parameter, you must specify both the port number and IP address/alias name.</p> <p>If you use alias name or long domain name resolution, you do not need to supply the IP address in the COMADDR parameter or in the COMMUNICATIONS ADDRESS field during Connect:Direct SIGNON. The name resolution feature resolves a valid alias name or long domain name to the matching IP address.</p> <p>The maximum length of the IP address alias name variable is 16 characters. If you want to use an LDNS (long domain name system) or an alias name greater than 16 characters, the name must be defined in the network map, and the TRANSPORT must be coded as NET.</p> <p>Note: Allocate the SYSTCPD DD to the TCPDATA file in both the Connect:Direct startup JCL and API session startup JCL when you request domain name resolution.</p>
ESF= <u>Yes</u> No	<p>Specifies whether the Extended Submit Facility (ESF) is available for the current signon. This parameter is optional. The default value is Yes.</p> <p>The ESF enables you to submit Processes even if the DTF is not active. When signing on to a Connect:Direct DTF that is active, but with no VTAM APPLID available, the ESF enables signon to the DTF. When you submit the Process it is enabled and placed in the TCQ as soon as the ESF.WAIT time expires.</p> <p>ESF is available only for Processes submitted on the local node. You can only issue the Connect:Direct SIGNON, SUBMIT, and SIGNOFF commands using the ESF; all other Connect:Direct commands are rejected.</p> <p> CAUTION: If you use the MAXDELAY PROCESS statement keyword, the message SCBI221I MAXDELAY not supported for ESF submits, is displayed when the DTF is down and the Process will not submit.</p>
FOLD= Yes <u>No</u>	<p>Determines printing upper case control. It specifies whether printed output, an option of the various SELECT commands, are all uppercase letters. The default value is No, specifying printed output of uppercase and lowercase letters. This parameter is optional.</p>
NODE= node name	<p>Specifies the name of the node (DTF) you sign on to. It is the name assigned to a node defined in the network map definitions. This parameter is optional.</p>

Parameter Value	Description
PACCT= 'pnode accounting data'	Specifies the accounting data for the primary node (PNODE). The maximum length of the accounting data is 256 characters. If special characters, such as a space, are part of the accounting data, you must enclose the string in single quotation marks. Connect:Direct uses this data as a default for each Process unless you override it on the SUBMIT command, PROCESS statement, or SUBMIT statement. This parameter is optional.
PRINT= destination of printed output	<p>Specifies an allocation string that determines the destination of the printed output that you request on the various SELECT commands. This parameter is optional. SYSOUT=A is the default value. An example follows.</p> <p>'SYSOUT=C DEST=RMT1 FREE=CLOSE RETURN=(DD) '</p> <p>You must include both FREE=CLOSE and RETURN=(DD) in the string for this specification. When using the IUI, if you do not specify a destination, Connect:Direct routes the output to the SYSOUT class specified in the initialization parameters (external to Connect:Direct) under the TSO user ID. If you modify the destination, you need to sign off Connect:Direct after updating the field in order for the new destination to be in effect.</p>
RECONNECT	Specifies that you can reestablish your VTAM session without having to sign off and sign on again. You can lose a VTAM session due to a link or line failure. This parameter is optional. The USERID parameter is the only other parameter that you specify with the RECONNECT option.
SACCT= 'snode accounting data'	Specifies the accounting data for the secondary node (SNODE). This parameter is optional. The maximum length of the accounting data is 256 characters. If special characters, such as a space, are part of the accounting data, you must enclose the string in single quotation marks. Connect:Direct uses this data as a default for each Process unless you override it on the SUBMIT command, PROCESS statement, or SUBMIT statement.
SPACE= (CYL TRK blk,([prim],[sec]))	<p>Specifies the temporary data set (TMPDSN) allocation type, primary space allocation, and secondary space allocation. This parameter is optional.</p> <p>The first subparameter is the allocation type. An allocation type of CYL specifies space allocation by cylinder. CYL is the default value. An allocation type of TRK specifies space allocation by track. An allocation type of blk specifies space allocation by the average block length of the data. Connect:Direct computes the number of tracks to allocate.</p> <p>The second subparameter, prim, specifies the primary allocation of storage. The default value is 1.</p> <p>The third subparameter, sec, specifies the secondary allocation of storage. The default value is 1.</p>
TMPDD= preallocated data set ddname	Specifies the ddname of a user-supplied preallocated data set. The maximum length is an 8-character alphanumeric string. This parameter is optional. Do not use both the TMPDD and TMPDSN parameters at the same time in the SIGNON command.

Parameter Value	Description
TMPDSN= preallocated data set name	<p>Specifies the name of a preallocated data set used to receive command results that are printed or displayed. The data set name is a 1-44 alphanumeric character string that must conform to z/OS standards for data set names. This parameter is optional, and when not specified, Connect:Direct builds a system temporary data set. It normally derives the name for the data set based on your TSO ID. In some cases, this derived data set name violates your facility naming convention. Use the Signon Defaults screen to override the data set name as needed.</p> <p>If you name a data set at signon time, preallocate it with the following attributes prior to signing on to Connect:Direct (4104 is the minimum BLKSIZE):</p> <pre>DSORG=PS LRECL=4100 RECFM=VBA BLKSIZE=4104</pre> <p>Do not use both the TMPDD and TMPDSN parameters at the same time in the SIGNON command.</p>
TRANSPORT= SNA TCP <u>NETmap</u>	<p>Defines the communications protocol to use for the session. This parameter is optional.</p> <p>The default value (NETmap) indicates that the protocol defined in the network map is used for the session.</p> <p>A value of SNA tells Connect:Direct to establish the session using SNA. You must define the node you are signing on to as an SNA node, or the SIGNON Process fails.</p> <p>A value of TCP tells Connect:Direct to establish a TCP/IP session. Connect:Direct uses the communications address specified in the COMADDR parameter (see page 62 for a description of the COMADDR parameter), if that parameter is specified as part of the SIGNON command. If the COMADDR parameter is not specified, Connect:Direct tries to establish the session using the communications address from the network map. If an invalid COMADDR parameter is supplied, Connect:Direct establishes an ESF API session.</p>
UNIT= temporary dsn unit type	<p>Specifies the temporary data set name (TMPDSN) unit type, which must be a 1-8 alphanumeric character string defining the unit type for data set allocation. This parameter is optional. If omitted, the operating system defined default for DASD is used by dynamic allocation.</p>
USERID= (ID, pswd, newpswd)	<p>Specifies your signon security information. This parameter is optional.</p> <p>ID specifies the security ID that Connect:Direct passes to a security exit. It can contain 1-64 characters of any kind, and is case-sensitive.</p> <p>pswd specifies the current security password. The security exit uses this parameter to validate the current security password. It contains 1-64 alphanumeric characters, and is case-sensitive.</p> <p>newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. It contains 1-64 alphanumeric characters, and is case-sensitive.</p> <p>When you attempt a sign on to a remote node, the user ID and password information must correspond to security procedures at that remote node.</p>
VOLSER= volume serial	<p>Specifies the temporary data set name (TMPDSN) volume serial number, which must be a 1-8 character alphanumeric string defining the volume serial number for data set allocation. This parameter is optional.</p>

Parameter Value	Description
DESCRIPTION= '30-character string'	Describes the API type that is logging on. It is displayed with the Select Task command. If Connect:Direct cannot determine the API type, the communication address is used. Defaults: DMCHLAPI, DMBATCH, ISPFUI, CONSOLEUI, ICOUI, CICSUI, and Java Application Interface
TIMEOUT= <u>Yes</u> No	Specifies whether the value specified for the TCP.API.TIMER global initialization parameter applies to this specific user. This parameter is optional. The system will only accept changes to this field made by an administrator and only for their own IUI session; all others will be governed by the initialization parameter. The default value is Yes. Do not use the TIMEOUT keyword in DMBATCH.

Using SIGNON through the Batch Interface

About this task

To use the SIGNON command from the batch interface, perform the following steps:

Procedure

1. Place the command in the DGADBATC job stream as described in [Sample Job Stream to Run the Batch Interface](#).
2. Submit the job while is running. If you are using the Extended Submit Facility (ESF), the DTF does not need to be active for the SIGNON command to process.

The following example shows a SIGNON command in which the user specifies a pre-allocated temporary data set named MYUID1.TEMP.DSN which resides on SYSDA. The accounting data is specified for the PNODE and SNODE.

```
SIGNON      NETMAP=DALLAS.NETMAP  -
            USERID=(MYUID1,MYPWD) -
            PACCT='JOB FOR SYMAINT, DEPT.27, MARKETING' -
            SACCT='JOB FOR INVENTORY, DEPT.55, ACCOUNTING' -
            TMPDSN='MYUID1.TEMP.DSN' -
            UNIT=SYSDA
```

See [Signing On to Connect:Direct for z/OS](#) for additional examples of the SIGNON command in the batch interface.

Signing On to the API

The communication address used to establish a connection to IBM Connect:Direct is determined by the TRANSPORT parameter defined in the SIGNON command. The default for the TRANSPORT parameter is NET (NETMAP) which means that the protocol defined in the NETMAP Adjacent node entry is used. For more information, see [Adjacent Node Definition Examples](#).

Parameter Value	Description
TRANSPORT = NET	Default. When TRANSPORT is defined as NET, the sign on process retrieves the Adjacent node entry to determine if the TCPAPI parameter has been defined. If TCPAPI exists, then a TCP connection is attempted using the communication address defined. The communication port number is obtained from the TCPAPI parameter, and if the IP address exists in the TCPAPI parameter, it is also used. If the IP address does not exist in the TCPAPI parameter, it must be obtained from either the Adjacent node or the LDNS parameter. If the TCPAPI does not exist, the APPLID parameter is retrieved and SNA is used as the protocol.

Parameter Value	Description
TRANSPORT = SNA	When TRANSPORT is defined as SNA, the sign on process retrieves the Adjacent node entry to determine the APPLID parameter, and SNA is used as the protocol. If the APPLID parameter does not exist, then an ESF SIGNON is performed.
TRANSPORT = TCP	When the TRANSPORT is defined as TCP, the communication address must be specified on the SIGNON command. For more information, go to the <i>IBM Connect:Direct for z/OS Administration Guide</i> and search for Initializing IBM Connect:Direct without SNA Support.

Adjacent Node Definition Examples

To only allow SNA API signons:

```
/* PNODE=SNODE WITH SNA API SIGNON ONLY */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
```

To only allow TCP API signons using an IPv4 address:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING IP ADDRESS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
)
```

To only allow TCP API signons using an IPv6 address:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING IP ADDRESS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,1111:2222:3333:4444:5555:6666:7777:8888) -
)
```

To only allow TCP API signons using the LDNS parameter:

```
/* PNODE=SNODE WITH TCP API SIGNON ONLY USING LDNS */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
)
```

To allow both SNA and TCP API signons:

```
/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,111.222.333.444) -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
or
/* PNODE=SNODE WITH BOTH SNA AND TCP API SIGNON */
ADJACENT.NODE=(( CD.OS390.NODE,M1DEV93C) PARSESS=(53 2) -
TCPAPI=(4198,) -
LDNS=long.domain.name -
APPLIDS=(M1CDI701 M1CDI702 M1CDI703) -
)
```

Signing On to Multiple Sessions

Use the SIGNON command to sign on simultaneously to multiple sessions of the DTF. The SIGNON command parameters which are valid for the multiple session signon are a subset of the full SIGNON command parameters. For more information, see [Using SIGNON through the Batch Interface](#).

Master Session

The first successful SIGNON command establishes the master session. The master session is the active session until another session is established. Each subsequent signon becomes the active session unless sessions are swapped using the SWAP NODE command.

Signing on to the master session and signing on to another node under the master session requires two different signons. The parameters specified for the master session signon dictate the environment for all sessions running under the master session.

Non-Master Session

For a session other than master, you are limited to the following SIGNON command parameters.

Label	Command	Parameters
(optional)	SIGNON	NODE= node name
		RECONNECT
		USERID= (ID, pswd, newpswd)

See [Using SIGNON through the Batch Interface](#) for descriptions of these parameters.

SIGNOFF Sequence

Carefully control your SIGNOFF command sequence because if you sign off the master session, IBM Connect:Direct signs off all other sessions. If you sign off a non-master session, IBM Connect:Direct only signs off that session, and the master session immediately becomes the active session again, regardless of the order in which the sessions are established. See [Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands](#) for examples of the importance of the sequence of your SIGNOFF commands.

Switching Nodes

When you issue multiple signons, one node can perform one function and another node a different function. The SWAP NODE command enables you to swap from node to node. See [Using SWAP NODE through the Batch Interface](#) for the SWAP NODE command description and examples.

Using the Batch Interface for Multiple Session SIGNON

To use the multiple session SIGNON command through the batch interface, place your commands in the DGADBATC job stream and submit the job while IBM Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, the DTF does not need to be running to submit the job.

See the examples in [Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands](#) for job streams that issue SIGNON, SWAP NODE, and SIGNOFF commands and an explanation of the significance of the command sequence in the job stream.

Using the IUI Multiple Session SIGNON Command

To access the Multiple Session Signon screen, select **MS** from the Primary Options Menu.

```

node.name                               hh:mm
CMD ==>                                yyyy.mm.dd
                                         yyyy.ddd

                                MULTIPLE SESSION SIGNON

USER ID  ==> USER01
PASSWORD ==>
NEW PASS ==>
NODE NAME ==>
TRANSPORT ==> NET
COMMUNICATION ADDRESS ==>(           ,           )
DESCRIPTION ==>
RECONNECT ==> N      (Y OR N. TO THE *ACTIVE* NODE
                     AFTER A SESSION FAILURE)
PNODE ACCOUNTING DATA ==>
SNODE ACCOUNTING DATA ==>
DO YOU WANT ALL COMMANDS FOR THIS SESSION TO BE CASE SENSITIVE? ==> NO

```

The Multiple Session Signon screen enables you to sign on to more than one node during a single ISPF/PDF session. The APPLIDS value in the network map Adjacent Node definition determines the maximum number of sessions that IBM Connect:Direct is allowed on one DTF. Each signon uses an additional VTAM APPLID.

See “SIGNON Command” on page 542 for field descriptions, or press the **PF1** key for Help.

Swap Node Command

The SWAP NODE command enables you to swap to another node when you are signed on to more than one node at a time. Use the SWAP NODE command to specify which node a command is issued to.

The SWAP NODE command has the following format and associated parameter:

Label	Command	Parameters
(optional)	SWAP NODE	node name

The only parameter for the SWAP NODE command is node name, which specifies which node to swap to in a multiple session environment. The node name is the name assigned to the LOCAL.NODE field defined in the network map definitions.

Using SWAP NODE through the UII

You can use the Swap/Display User Sessions screen to swap to another node or delete (signoff) another node.

Procedure

1. Select option **SW** from the Primary Options Menu to access the SWAP/DISPLAY User Sessions screen. The screen displays all user sessions for this ISPF/PDF session.

```

-----SWAP/DISPLAY USER SESSIONS----- Row 1 to 2 of 2
==>                                     SCROLL ==> PAGE
      NODE          COMMUNICATION ID      CURRENT  STATUS
      USERID
-----
node.name1          userid2      applid0    *ACTIVE*  *MASTER* **LOCAL*
node.name3          userid1      applid1
***** Bottom of data *****

```

2. Do one of the following:

- To swap sessions, type **S** in the input field to the left of the node name and press **Enter**. ACTIVE is displayed next to the node name you just selected. To determine which node is currently signed on when this field is not displayed, look to the upper left corner of most panels for the display of the node name.
- To delete or sign off a session, type **D** in the input field to the left of the node name and press **Enter**.

Note: If you sign off the master session, the first node you signed on, all sessions are lost.

Using SWAP NODE through the Batch Interface

To use the SWAP NODE command from the batch interface:

Procedure

1. Place your commands in the DGADBATC job stream as described in [Sample Job Stream to Run the Batch Interface](#).
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

For example, the following command swaps you from your current active remote node with a user ID of JONES2 to the local node under the user ID of JONES1:

```
//SYSIN DD *
SIGNON USERID = (JONES1) -
      NODE = CD.LOCAL
SIGNON USERID = (JONES2) -
      NODE = CD.REMOTE
SWAP NODE CD.LOCAL
SELECT NETMAP WHERE(NODE=CD.LOCAL)
SIGNOFF
SIGNOFF
/*
```

Sign Off Command

The SIGNOFF command terminates a connection to the DTF. If you are using the Extended Submit Facility (ESF), you can submit the SIGNOFF command even though the DTF is inactive.

The SIGNOFF command has the following format. No associated parameters are available.

Label	Command	Parameters
(optional)	SIGNOFF	

Using SIGNOFF Through the IUI

The following table identifies how to SIGNOFF from various locations of the IUI. You must press **Enter** after each SIGNOFF command.

SIGNOFF Command	Location
SIGNOFF	SIGNON screen or the Primary Options Menu
End	SIGNON screen or the Primary Options Menu
X	SIGNON screen or the Primary Options Menu
=SIGNOFF or =X	Command Line

When you sign off IBM Connect:Direct, the system returns you to the ISPF menu.

Follow these rules when you sign off IBM Connect:Direct:

- You must type a SIGNOFF command twice from the command line.

Note: The first SIGNOFF command returns you to the Primary Options menu and the second SIGNOFF command returns you to the ISPF menu.

- You only need to sign off once from the SIGNON screen or the Primary Options Menu.

Using SIGNOFF through the Batch Interface

To use the SIGNOFF command from the batch interface, perform the following steps:

Procedure

1. Place your commands in the DGADBATC job stream as described in [“Sample job stream to run the batch interface”](#) on page 513.
2. Submit the job while IBM Connect:Direct is running. If you have the Extended Submit Facility (ESF) available, the DTF does not need to be running in order for the SIGNOFF command to function.
3. Verify your results.

The following example shows a SIGNON command followed by a SIGNOFF command.

```
SIGNON    USERID=(MYUID1,MYPSWD) NETMAP=NETMAP.NAME  
SIGNOFF
```

See the examples in [Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands](#) for more examples of the SIGNOFF command used with the Multiple Session Signon feature and the SWAP command.

Sequencing the SIGNON, SWAP NODE, and SIGNOFF Commands

When you issue SIGNON, SWAP NODE, and SIGNOFF commands, the sequence in which you issue these commands is important, as shown in the following examples.

Example 1: Terminating Two Sessions with One SIGNOFF Command

The following example shows a DGADBATC SIGNON to two different nodes and the use of the SWAP NODE command. The SIGNOFF command in this example causes IBM Connect:Direct to terminate both sessions, because the active session at signoff is the master session.

```
/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER) */  
SIGNON    USERID=(SAM) NODE=CD.BOSTON  
/* SIGN ON TO JERSEY */  
SIGNON    USERID=(SAM) NODE=CD.JERSEY  
/* SUBMIT COPY ON JERSEY */  
SUB       PROC=COPY1  
/* SWAP TO BOSTON */  
SWAP NODE CD.BOSTON  
/* SUBMIT COPY ON BOSTON */  
SUB       PROC=COPY1  
/* SIGN OFF BOTH NODES (terminating master session)*/  
SIGNOFF
```

Example 2: Terminating a Master Session with the SIGNOFF Command

In the following example, the first SIGNOFF command terminates the CD.NEWYORK node because it is the last SIGNON command issued. The second SIGNOFF command terminates both the CD.JERSEY and CD.BOSTON nodes because the first SIGNOFF command made the master session the active session and signing off the master session signs off all other sessions.

```

/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER */
SIGNON  USERID=(SAM) NODE=CD.BOSTON
/* SIGN ON TO JERSEY */
SIGNON  USERID=(SAM) NODE=CD.JERSEY
/* SIGN ON TO NEWYORK */
SIGNON  USERID=(SAM) NODE=CD.NEWYORK
/* BOSTON IS ACTIVE AFTER SIGNOFF TO CD.NEWYORK */
SIGNOFF
/* SIGN OFF ALL SESSIONS (terminating master session) */
SIGNOFF

```

Example 3: Terminating a Non-Master Session with the SIGNOFF Command

The following example illustrates that whenever a SIGNOFF command is issued to a non-master session, the master session automatically becomes active again regardless of the order in which the nodes are signed on.

If you try to swap to a node that is already active, the SWAP NODE command is ignored. The USERID (or any other user ID specified) must be a user record in the IBM Connect:Direct authorization file or a valid ID for the security subsystem in use, to sign on to the node.

```

/* NETMAP = CD.NETMAP, BOSTON IS ACTIVE (MASTER) */
SIGNON  USERID=(SAM) NODE=CD.BOSTON
/* SIGN ON TO JERSEY */
SIGNON  USERID=(SAM) NODE=CD.JERSEY
/* SIGN ON TO NEWYORK */
SIGNON  USERID=(SAM) NODE=CD.NEWYORK
/* NEWYORK ACTIVE */
SIGNOFF
/* SIGN ON TO JERSEY */
SWAP NODE CD.JERSEY
/* JERSEY ACTIVE */
SIGNOFF
/* SIGN OFF ALL SESSIONS */
SIGNOFF

```

Building, Modifying, and Submitting Processes

You can construct and submit Processes for execution in several ways, depending upon the user interfaces available on your operating environment. These IBM Connect:Direct user interfaces include:

- Interactive User Interface (IUI)
- Connect:Direct Browser User Interface
- Sterling Connect:Direct File Agent
- Batch interface

User-written programs issued through the IBM Connect:Direct Application Program Interface (API)

For more information about monitoring and manipulating your submitted Processes, see [Controlling Processes with Commands](#).

Note: The maximum size allowed for a Process is 1 MB.

You can also use the Connect:Direct Browser User Interface to perform some Connect:Direct for z/OS procedures.

The following steps describe how a Process executes:

1. You submit a Process.

You create and submit a new Process or submit a predefined Process from a IBM Connect:Direct Process library.

2. The parser checks the syntax of the Process.
3. The Process is queued for execution and the submit message is issued

If the Process passes syntax checking, it is placed in a work queue based on Process parameters, such as priority, class, and start time.

IBM Connect:Direct work queues are referred to as the Transmission Control Queue (TCQ) or the Process queue. A Process can have one of the following queue states in the TCQ:

State	Description
EXECUTION	Indicates that the Process is executing.
WAIT	Indicates that the Process is waiting until a connection is established or available.
HOLD	Indicates that the Process is held on the queue until released by an operator or the SNODE connects with a request for held work.
TIMER	Indicates that the Process is submitted to execute at the user-specified time or date.
PROCESS RETENTION	Indicates the process has completed and would have been deleted, but PROCESS.RETENTION=YES was specified in the INITPARMS. The process can only be viewed or deleted.

You can manipulate any Process in the TCQ with various IBM Connect:Direct commands that are described in the next chapter.

A message indicating that the Process submitted successfully is returned when the Process is placed in the TCQ.

4. The Process is queued for execution and the submit message is issued.

You can manipulate any Process in the TCQ with various IBM Connect:Direct commands that are described in the next chapter.

A message indicating that the Process submitted successfully is returned when the Process is placed in the TCQ.

5. IBM Connect:Direct finds an available connection and Process execution begins. The Process is selected for execution based on Process parameters and the availability of the remote node.

Process Routing

The RETAIN, HOLD, and STARTT parameters route Processes as described in the following table:

Parameters	Queue	Comments
None	Wait	Process remains on the Wait queue until IBM Connect:Direct can start a session with the SNODE at which time it moves to the Execution queue.
RETAIN=INITIAL	Hold	Process automatically executes each time IBM Connect:Direct is initialized with TCQ=WARM.
RETAIN=YES	Hold	A copy of the Process is kept in the Hold queue after it has executed. The Process does not execute again until it is released by a CHANGE PROCESS command.
HOLD=YES	Hold	Process remains in the Hold queue until someone releases the Process.
HOLD=CALL	Hold	Process is automatically moved from the Hold queue to the Wait queue when the SNODE contacts the node on which the Process resides.
STARTT	Timer	When the scheduled time and date arrive, the Process is put on the wait queue and is available for execution.

RETAIN=INITIAL is useful for Processes that contact other IBM Connect:Direct nodes each time IBM Connect:Direct completes initialization. This action causes any work queued on the remote node for the local node to begin. The DGAPHTBT sample Process allows you to test if IBM Connect:Direct is running on a SNODE without actually running a Process on the node. For more information, go to the IBM Connect:Direct for z/OS Administration Guide, search for Maintaining the Network, and read the note on the DGAPHTBT Process.

You can use RETAIN=YES combined with STARTT to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(,12:00) starts the Process each day at noon.

Note: The RETAIN process parameter has no relationship to the PROCESS.RETENTION initialization parameter.

The Timer queue is also used for session retry and file allocation retry based on IBM Connect:Direct initialization parameters specified by a particular installation. When retry limits are exhausted, the Process is moved to the Hold queue with a WC (hold queue, waiting connection) status for session retry.

Note: For allocation failures to be requeued, you must set the REQUEUE initialization parameter to YES so that the Process is placed in the HO RA (HO = Held by Operator, RA = Held for Restart Due to Allocation Error) queue.

Process Queuing

When you submit a Process, IBM Connect:Direct puts the Process in a logical queue based on the Process statement routing parameters on [Process Routing](#).

IBM Connect:Direct selects Processes in a first-in first-out manner for execution within Process class and priority as sessions are available. The following section describes in detail how IBM Connect:Direct selects Processes for execution.

Process Selection

IBM Connect:Direct uses the parallel sessions capability so that multiple Processes can execute simultaneously between any two IBM Connect:Direct nodes. Process selection for each parallel session is based on a class that the user specifies on a Process.

Note: Parallel sessions support requires you to specify PARSESS=YES in the VTAM application definition for both nodes. If two nodes have differing values for parallel sessions, transfers are limited by the maximum number of sessions in the ADJACENT.NODE definition on the node where the Process is submitted. For TCP/IP connections, IBM Connect:Direct manages the parallel sessions and provides multiple connections between the two TCP/IP partners.

You define the maximum number of sessions between two nodes in the network map. Because each session has a corresponding class value, the maximum number of sessions and maximum number of classes are equal. Selection of a Process for execution in a given node is based on Process priority (the PRTY parameter of the Process statement) within session class. User-specified class values allow a Process to execute on the session having the matching class value or on sessions with higher class values. The default class is the value specified in the IBM Connect:Direct network map.

When one Process completes, IBM Connect:Direct selects another Process (if any) to run on the available session. IBM Connect:Direct searches until it finds the first Process with a class eligible to execute on the available session.

A typical use for classes is to define critical Processes with low class values so that more sessions are available for their execution. You can specify higher class numbers for time-consuming Processes; this enables sessions with corresponding lower class numbers to become available more frequently.

Intelligent Session Retry

The Intelligent Session Retry feature provides the ability to manage session retries when a connection to a specific node is down so that all Processes are not retried at the same time. This feature makes the

existing connection retry facility more efficient by reducing the extra processing overhead created when each IBM Connect:Direct Process retries the connection independently.

When multiple Processes are submitted for a specific node that is down, the Processes all initially attempt to connect. When this fails, the first Process submitted to this node continues to retry until MAXRETRIES is exceeded. The other Processes to this node are moved to the Hold (HO) queue with a RH (held for retry) status and are not retried. When MAXRETRIES is exceeded for the first Process, it is moved to the Hold (HO) queue with a WC (waiting connection) status for session retry.

After the first Process is moved to HO WC status, it can be restarted by one of the following ways:

- You can manually release the Process causing it to be retried until a connection is made or MAXRETRIES is exceeded, when it will return to HO WC status
- Another Process is submitted to this same node, causing the Process in HO WC status to be retried until either a connection is made or MAXRETRIES is exceeded, returning it to HO WC status
- The SNODE initiates communication with the PNODE, causing the Process in the HO WC status to be released

When the first Process in HO WC status connects and is released, this causes the other Processes being held in HO RH status to be released serially. Therefore, only one Process going to an inactive node will retry until the connection is made resulting in less overhead, since multiple Processes going to the same node are not all attempting to retry at the same time. This also reduces the number of statistic records which would be produced for multiple retries.

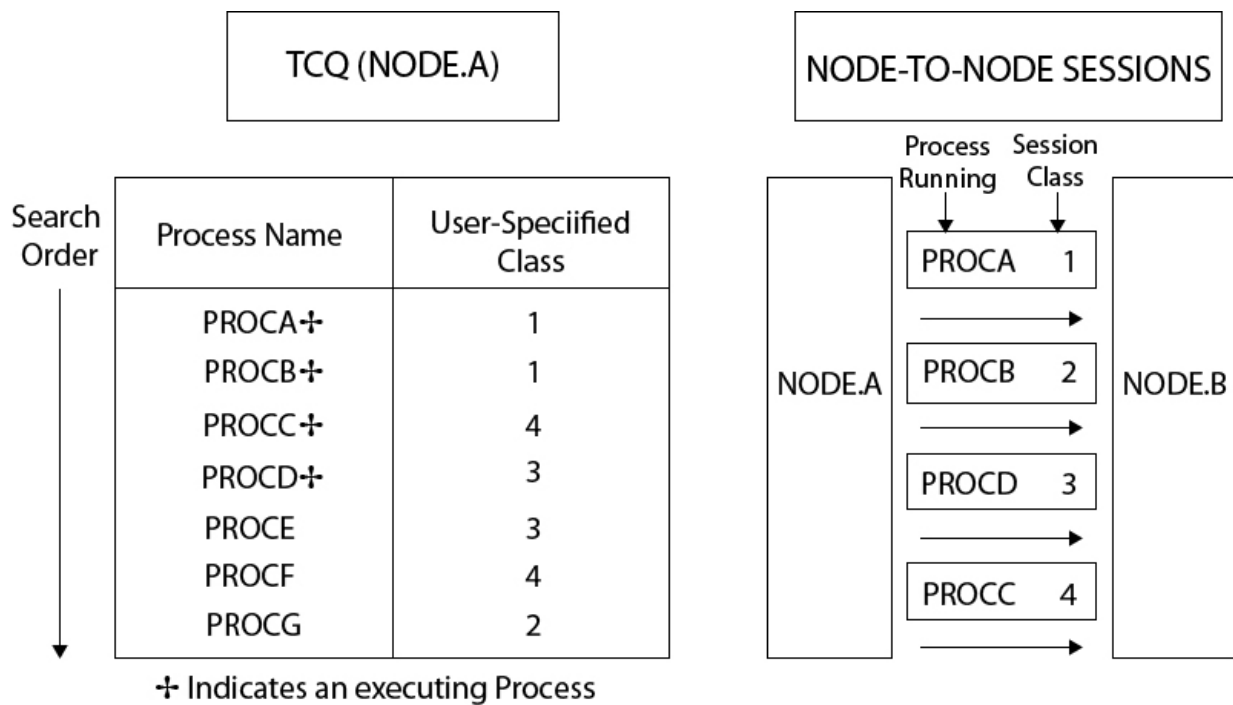
Note: The node name used by the Intelligent Retry feature is the specific name used for the SNODE parameter. This means that if the SNODE is known by more than one node name or TCPNAME value, only those in the queue waiting with that value will be recognized for release.

Process Execution Example

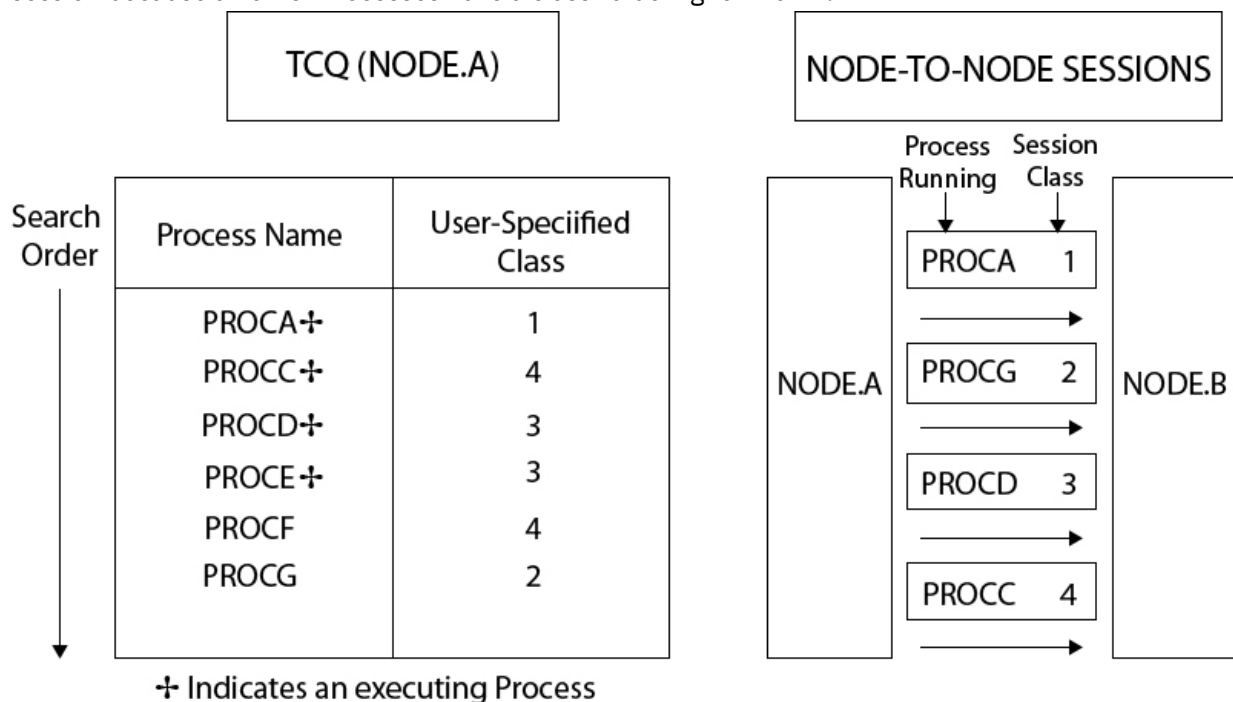
In the following example, IBM Connect:Direct is running. Seven Processes are submitted to the TCQ on NODE.A to execute on NODE.B. All of the submitted Processes are ready to run. In addition, all Processes have a user-specified class value and the same priority. (Class determines which session IBM Connect:Direct selects to run. Class values allow a Process to execute on a session with a matching class value or on sessions with higher class values.)

This site configuration enables up to four sessions to run between NODE.A and NODE.B. Each session between NODE.A and NODE.B has its own unique class number. IBM Connect:Direct executes in the following order:

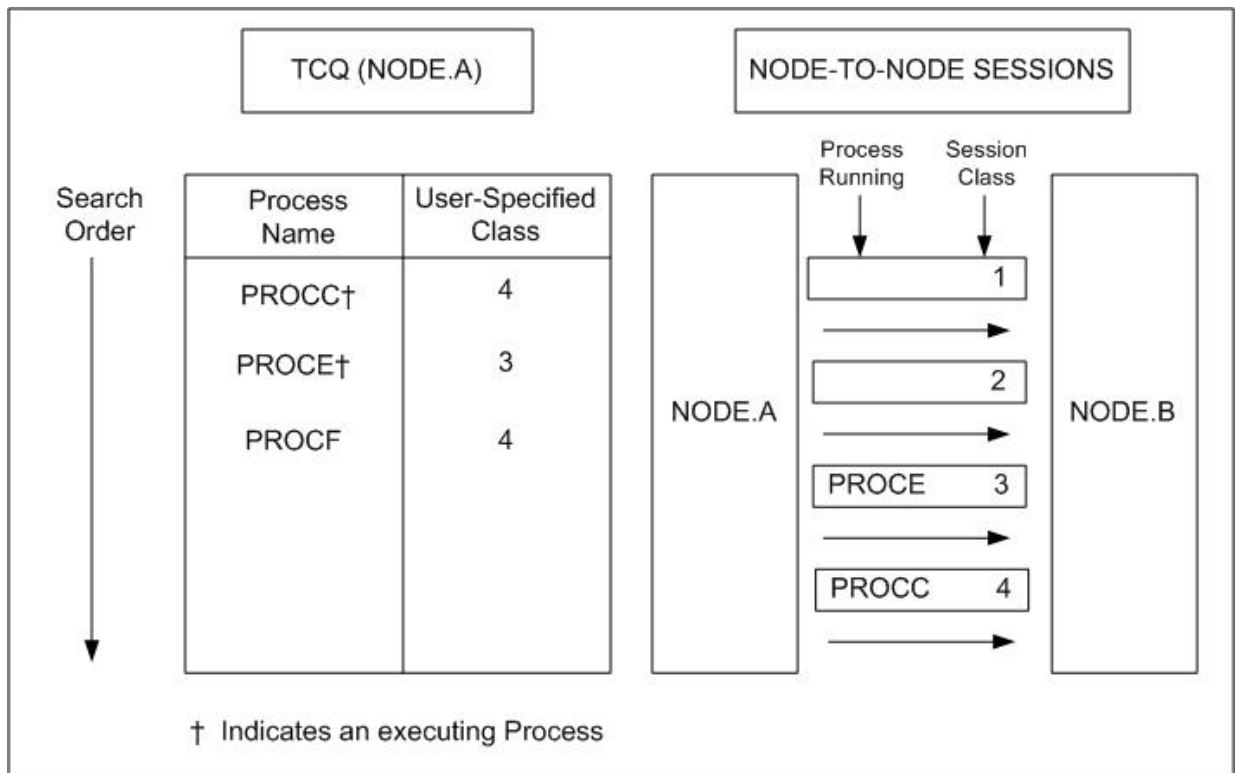
1. NODE.A simultaneously starts four sessions, as in the following figure. Processes are displayed in the same order they appear in the queue. Because the user-specified class of PROCA and PROCB is 1, they can run on the class 2, 3, or 4 sessions, if needed.



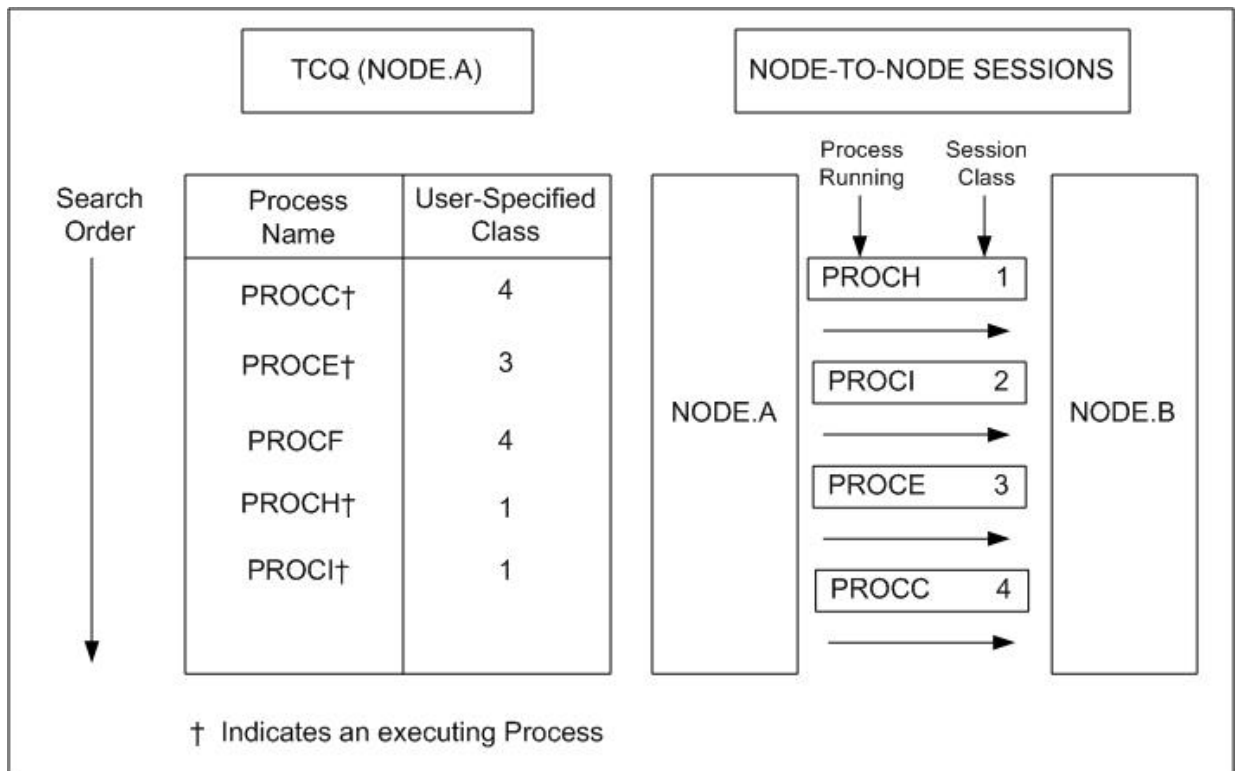
2. PROCB completes execution, making a session available. IBM Connect:Direct looks through the TCQ for the first eligible Process for that session. PROCG is the next Process available to run on the class 2 session because all other Processes have a class value higher than 2.



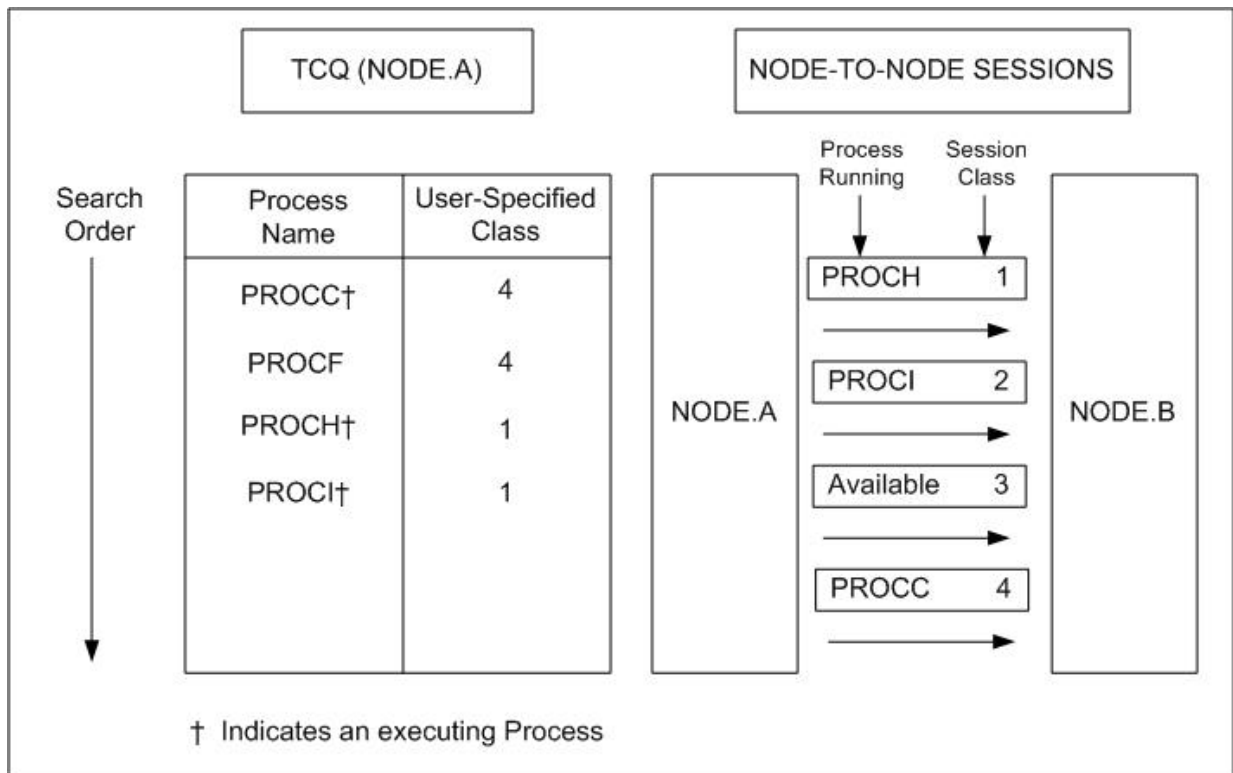
3. PROCA, PROCG, and PROCD complete execution. PROCE begins executing in the class 3 session and PROCC begins executing in the class 4 session. The class 1 and class 2 sessions cannot be used because the only Process remaining in the queue (PROCF) is class 4. PROCF must wait for an available class 4 session.



4. PROCH and PROCI are submitted. PROCF continues to wait, because it can only run in a class 4 session.



5. PROCE completes. PROCF is still waiting for PROCC to complete. If another Process is submitted for class 1, 2, or 3, it can use the available class 3 session.



Process Management in a IBM Connect:Direct/Plex

In a IBM Connect:Direct/Plex, the IBM Connect:Direct Manager uses a set of Process management rules to schedule Processes to IBM Connect:Direct Servers. These rules are applied in the following order:

1. The IBM Connect:Direct Manager first checks if the remote node is running a IBM Connect:Direct release that supports multiple servers. If the remote node does not support multiple servers, the IBM Connect:Direct Manager schedules all Processes to the IBM Connect:Direct Server that is already running a Process with this node, and Process management does not occur. If the node supports multiple servers, the IBM Connect:Direct Manager continues with the Process management rules that follow.
2. The IBM Connect:Direct Manager determines which IBM Connect:Direct Servers on the remote node can run the Process, based on the PLEXCLASS of the IBM Connect:Direct Server and the Process, and the current transport support in the IBM Connect:Direct Server.
3. The IBM Connect:Direct Manager then determines which IBM Connect:Direct Server from step 2 has the lowest workload. The current Process workload of each IBM Connect:Direct Server is determined by dividing the number of active Processes on the server by its MAXPROCESS initialization parameter value.
4. If two or more IBM Connect:Direct Servers have the lowest workload, the Process is sent to the IBM Connect:Direct Server that has the earliest last scheduled time (the time of day that the IBM Connect:Direct Server Process is scheduled).

Note: If a IBM Connect:Direct/Plex communicates with an external IBM Connect:Direct system, the external system must send work to the IBM Connect:Direct Manager to balance Process workloads in the IBM Connect:Direct/Plex. If an external IBM Connect:Direct system communicates directly with a IBM Connect:Direct Server, work from the external system is not distributed using Process management; the IBM Connect:Direct Server simply does the requested work. However, the IBM Connect:Direct Server still informs the IBM Connect:Direct Manager that it is processing work, so the IBM Connect:Direct Manager can continue to correctly balance Process workload in the IBM Connect:Direct/Plex.

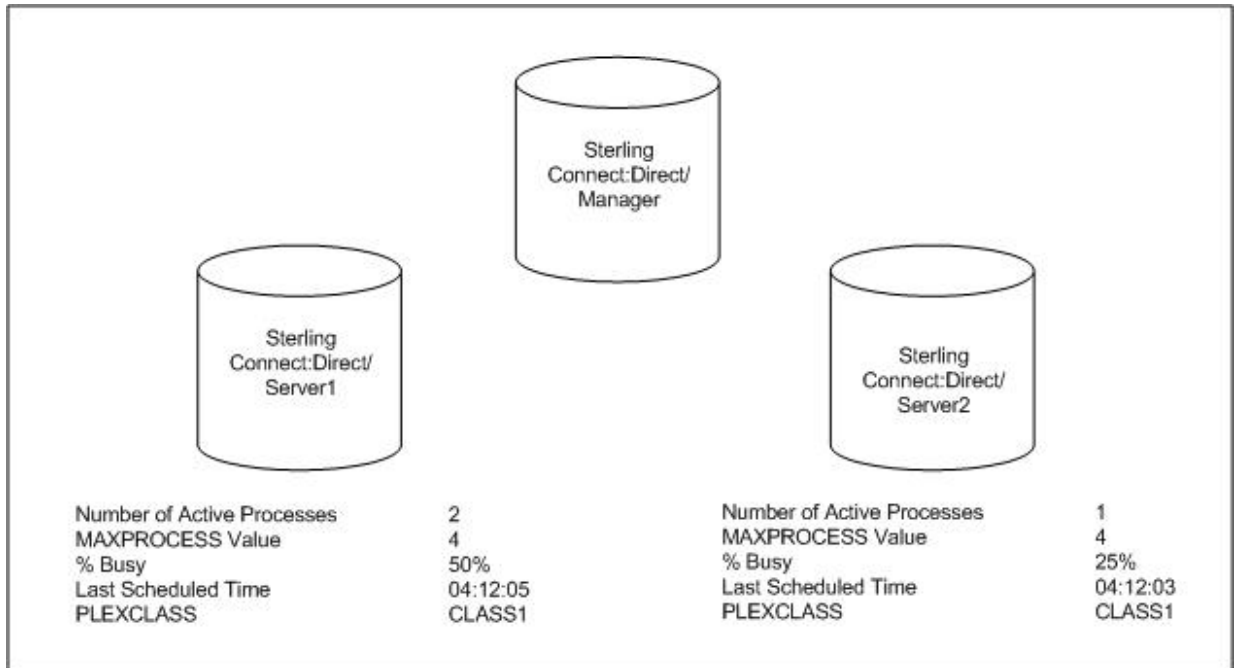
Also, to perform workload balancing with an external IBM Connect:Direct system when the IBM Connect:Direct/Plex is the SNODE, the external system must be a IBM Connect:Direct version that

supports the IBM Connect:Direct/Plex environment. You do not have to configure the external system as a IBM Connect:Direct/Plex. Configure it to communicate directly with the IBM Connect:Direct Manager.

Example: Process Management Steps

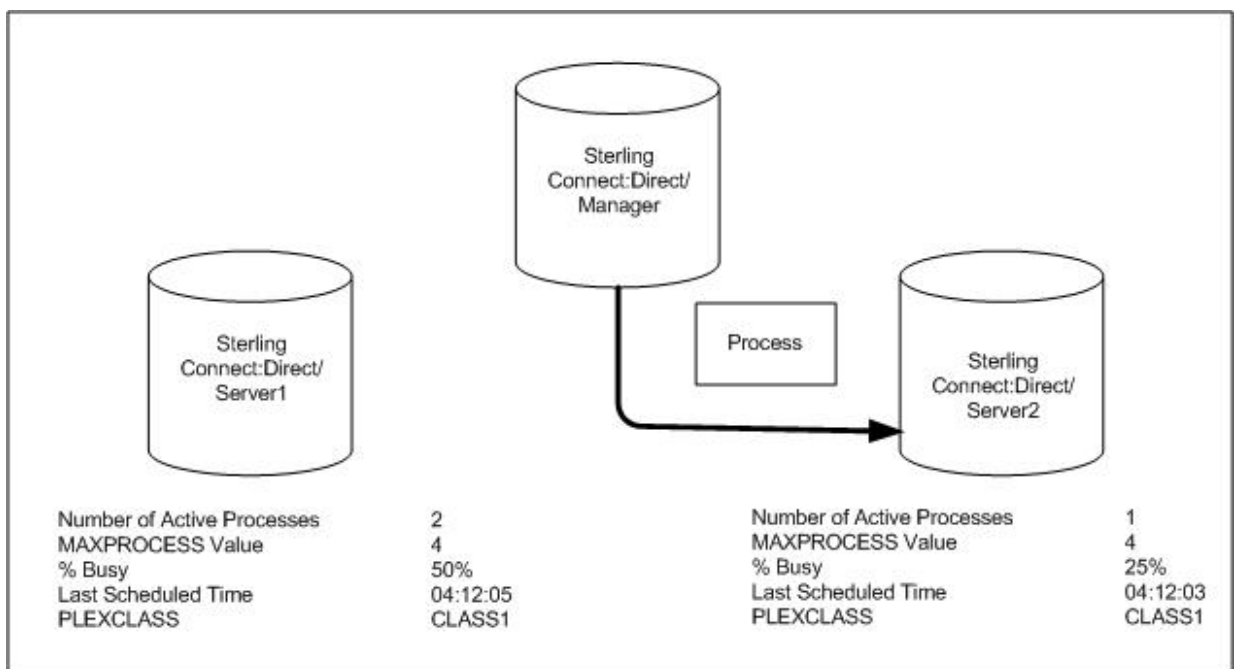
Following is an example of the steps in Process management.

1. A IBM Connect:Direct/Plex consists of a IBM Connect:Direct Manager and two IBM Connect:Direct Servers. Both IBM Connect:Direct Servers can process the same PLEXCLASS (CLASS1).

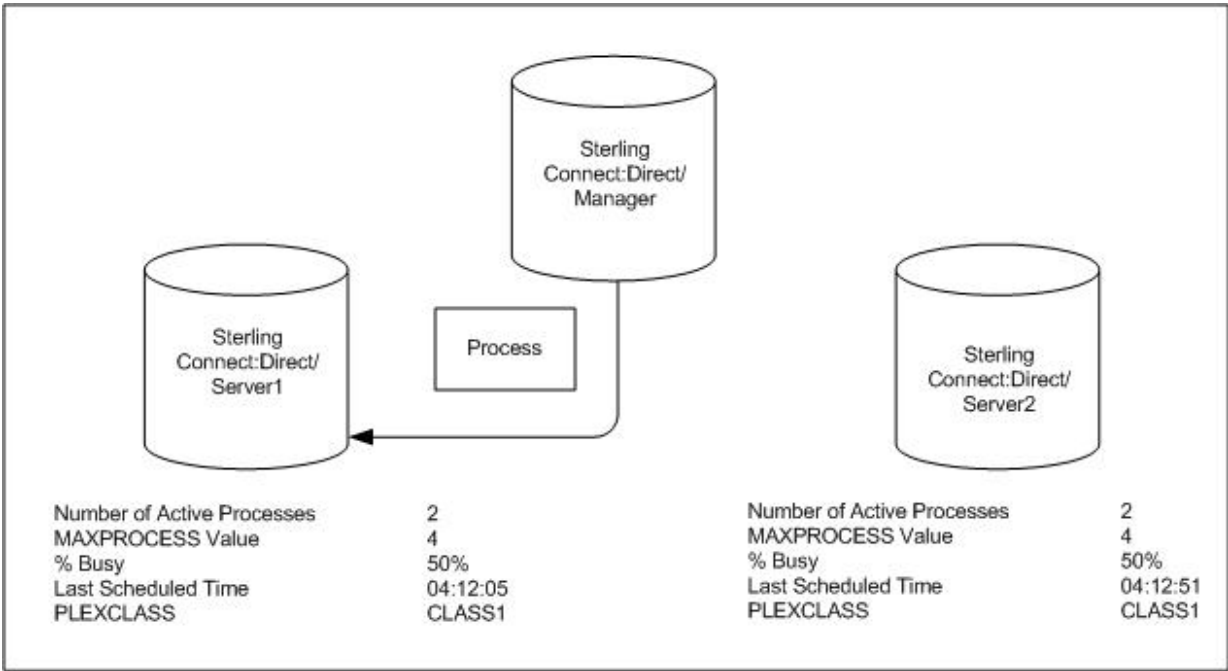


2. The IBM Connect:Direct Manager receives a new Process. After determining that both IBM Connect:Direct Servers support the PLEXCLASS required to run the Process, the IBM Connect:Direct Manager determines which IBM Connect:Direct Server has the lowest workload.

In this case, SERVER2 has the lowest workload; it has one active Process out of a maximum of four. The IBM Connect:Direct Manager routes the Process to SERVER2, and the last scheduled time of SERVER2 is updated to the current time.



3. The IBM Connect:Direct Manager has another Process to schedule. SERVER1 and SERVER2 have the same workload. However, the last Process is scheduled to SERVER2, so the IBM Connect:Direct Manager routes the new Process to SERVER1.



SUBMIT Command

The SUBMIT command enables you to submit a Process to the TCQ for execution. Parameters you specify on the SUBMIT command override any corresponding parameters specified in the Process itself.

Note: If you are using the Extended Submit Facility (ESF), you can use the SUBMIT command to submit Processes to the TCQ, even if the DTF is inactive.

The SUBMIT command has the following format. The Label is optional.

Label	SUBmit	Parameters and Subparameters
-------	--------	------------------------------

The SUBMIT command has the following parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters
(optional)	SUBmit	<p>PROC= member or DSN= dsn dsn(member)</p> <p>CASE= Yes No CLASS= n DEBUG= trace settings for this Process FASP= No SSP FASP.BANDWIDTH=nnn nM nG FASP.FILESIZE.THRESHOLD=nnn nM nG FASP.POLICY=FAIR HIGH LOW FIXED HOLD= Yes No Call MAXDELAY= [unlimited queued 0 hh:mm:ss] NEWNAME= newname NOTIFY= %USER user PACCT= 'pnode accounting data' PLEXCLASS= (pnode class, snode class) PNODE= primary node name PNODEID= (ID, pswd, newpswd) PRTY= n REQUEUE= Yes No RETAIN= Yes No Initial SACCT= 'snode accounting data' SNODE= secondary node name TCPNAME=tcpvalue;port SNODEID= (ID, pswd, newpswd) &symbolic 1= variable string 1 &symbolic 2= variable string 2 ...=... &symbolic N= variable string N</p>

The following tables describes SUBMIT command parameters:

Parameter	Description
PROC = member	<p>Specifies the Process Library member name. This name is the member of the PUBLIB which contains the Process you are submitting. The member name is a 1-8 character alphanumeric string, with the first character alphabetic. This parameter is required if the Process resides in the IBM Connect:Direct PUBLIB.</p> <p>The IBM Connect:Direct PUBLIB PDS is specified by the DMPUBLIB ddname allocated to the TSO session or specified in the batch job stream.</p>
DSN = dsn dsn (member)	<p>Specifies the data set name or the name of the member of a PDS that contains the Process if the Process is not in the IBM Connect:Direct PUBLIB. This parameter is required if the Process does not reside in the IBM Connect:Direct PUBLIB.</p> <p>dsn is the data set name of a sequential file that contains the Process.</p> <p>dsn(member) specifies the PDS name and member name (in parentheses) that contains the Process. Specify either PROC or DSN; do not specify both.</p>

Parameter	Description
CASE = Yes <u>No</u>	Specifies whether parameters associated with accounting data, user ID, password, and data set name in the command and in the Process are to be case sensitive. This parameter is optional.
CLASS = n	Determines the node-to-node session on which a Process can execute. See your IBM Connect:Direct administrator for which class to specify. The range is 1-255. This parameter is optional.
DEBUG= trace	Specifies the 8-position trace setting for this Process. This parameter enables you to specify a part trace for a particular Process. For more information see Debug Settings in the <i>IBM Connect:Direct for z/OS Administration Guide</i> .
HOLD = <u>No</u> Yes Call	<p>Specifies whether or not the Processes are placed in the hold queue. This parameter is optional.</p> <p>No specifies that IBM Connect:Direct does not place the Process in the hold queue, but places it in the WAIT for execution queue (EX). HOLD=NO is the default.</p> <p>Yes specifies that the Process remains in the hold queue until either a CHANGE PROCESS command releases the Process or a DELETE PROCESS command deletes the Process.</p> <p>When HOLD=YES and you specify a value for the STARTT parameter, IBM Connect:Direct places the Process submitted in the hold queue.</p> <p>Note: When RETAIN=Y, IBM Connect:Direct ignores the HOLD parameter.</p> <p>Call specifies that IBM Connect:Direct is to place the Process in the hold queue until a session is established with the specified SNODE. This session is established by either another Process starting on the PNODE destined for the same SNODE or the SNODE contacting the PNODE. For example, a Process submitted with HOLD=NO establishes a session and causes execution of any Processes residing on the SNODE destined for this node that are submitted with HOLD=CALL.</p> <p>Note: To support dial-up connections from Connect:Direct for Microsoft Windows platforms and release Processes serially, define the SNODE using a TCP/IP address of 0.0.0.0 (a null IP address). You may let the port number default since it will be resolved at connection time. Processes submitted to such nodes will default to HOLD=CALL status and can only be released by a NULL or ENABLE process from Connect:Direct for Microsoft Windows. Releasing Processes to such nodes will immediately cause them to return to HOLD=CALL status without executing because the connection cannot be resolved. Processes initiated by such nodes will execute normally, but will not release the HOLD=CALL Processes. Checkpoint restart is supported for such nodes.</p>

Parameter	Description
MAXDELAY = [unlimited queued 0 hh:mm:ss]	<p>Indicates that the submit command waits until the submitted Process completes execution or the specified time interval expires. This parameter is optional. Do not use MAXDELAY for a submit within a Process—use only in SUBMIT commands.</p> <p>Note: If the Process does not complete within the time interval specified by queued or hh:mm:ss, the API returns SSPA006I, RC=4 and DGADBATC terminates with RC=48 (x'30').</p> <p>unlimited specifies that the submit command waits until the Process completes execution. If MAXDELAY is coded and no parameters are specified then in this case MAXDELAY will default to unlimited.</p> <p>queued specifies that the submit command waits until the Process completes or 30 minutes, whichever occurs first.</p> <p>0 specifies that the submit command attempts to start a session for the submitted Process to execute on immediately. If IBM Connect:Direct cannot establish a session, after all retries are exhausted, the Process is flushed and the submit command fails with the error SVTM118I RC=52(x'34').</p> <p>Note: If IBM Connect:Direct cannot establish a session after all retries are exhausted due to all available sessions on the remote node being in use, that is, when session attempts fail with error SVTM080I SESSION (nnn) REJECTED pname (pnum) SNODE=remote.node, the Process is flushed and the submit command fails with the error SVTM118I RC=12(x'0C').</p> <p>MAXDELAY=0 Processes will not use the intelligent retry feature. When a transfer to a remote node times out and retries, subsequent transfers to the same remote node will also time out and retry rather than being added to the wait queue.</p> <p>hh:mm:ss specifies that the submit command waits for an interval no longer than the specified hours, minutes, and seconds or until the Process completes, whichever occurs first.</p>
NEWNAME = newname	<p>Specifies the new name of the Process. The default is the label on the Process statement. This parameter is optional.</p>
NOTIFY = %USER user ID	<p>Specifies the user ID to receive Process completion messages. This parameter is optional.</p> <p>%USER specifies that the user who submitted the Process receives the completion messages, if the IBM Connect:Direct user ID that the user is currently logged on with is the same as the TSO user ID. If the IBM Connect:Direct user ID is different from the TSO user ID, the TSO user is not notified but the ID to which IBM Connect:Direct is logged on will be notified.</p> <p>user ID specifies the TSO user ID that receives Process completion messages.</p> <p>Note: The NOTIFY capability is not supported across z/OS images in the sysplex environment. IBM Connect:Direct cannot send Process completion messages across z/OS images in a sysplex.</p>

Parameter	Description
PACCT = 'pnode accounting data'	<p>Accounting data for the PNODE. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, you must enclose the string in single quotation marks. This parameter is optional.</p> <p>This data overrides any accounting data specified on the SIGNON command and any accounting data specified in the Process statement of the submitted Process.</p>
PLEXCLASS = (pnode class, snode class)	<p>Specifies the class that directs the Process to only certain servers in a IBM Connect:Direct/Plex. This parameter is only used in a IBM Connect:Direct/Plex and is optional.</p> <p>You can designate each server in a IBM Connect:Direct/Plex to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. You can then limit Processes to only those servers by specifying the PLEXCLASS in the Process definition.</p> <p>The pnode class controls which IBM Connect:Direct Server runs the Process. The snode class controls what other node is used with the Process. The pnode class and snode class are each 1-8 characters long. An asterisk (*) indicates that the Process runs on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If you do not specify a PLEXCLASS, the network map is checked for a default PLEXCLASS. If the network map does not specify a default PLEXCLASS, then an asterisk is used as the default.</p> <p>If a Process must run on a specific IBM Connect:Direct Server, specify the IBM Connect:Direct Server name in this field. The Process only runs on that server.</p>
PNODE = primary node name	<p>Specifies the primary node of the Process. This parameter is optional.</p> <p>primary node name is a 1-16 alphanumeric character name defined in the network map. You can express the name in alphanumerics or nationals (@#\$) with embedded periods.</p> <p>The node to which you submit the Process is always the PNODE. You do not need to specify this parameter. It defaults to the name of the node submitting the Process. PNODE is for documentation purposes only.</p>
PNODEID = (ID, pswd, newpswd)	<p>Specifies security user IDs and security passwords at the PNODE. This parameter is optional. ID specifies the security ID that IBM Connect:Direct passes to a security exit for validations on the PNODE side of the Process. The range is 1-64 alphanumeric characters.</p> <p>pswd specifies the current security password. The security exit uses this parameter to validate the current security password on the PNODE side of the Process. It can contain 1-64 alphanumeric characters.</p> <p>newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. The range is 1-64 alphanumeric characters.</p>

Parameter	Description
PRTY = n	Specifies the Process priority in the Transmission Control Queue. High numbers indicate high priorities; low numbers indicate low priorities. IBM Connect:Direct uses this priority only for Process selection; it does not affect the priority during transmission. The default is defined during your installation. The range is from 0-15. This parameter is optional.
REQUEUE = Yes <u>No</u>	<p>Specifies whether IBM Connect:Direct requeues a COPY step if an ABEND occurs during processing. This parameter is optional.</p> <p>Yes places the requeued Process in the HOLD queue with a status of HELD IN ERROR (HE). You can then take corrective action and restart the Process at the step that failed. Note that you must explicitly release the Process from the HOLD queue when the status is HELD IN ERROR (HE).</p> <p>No specifies the failing copy step is not requeued if it fails with an ABEND (such as X37). The remaining steps in the Process are allowed to execute. The default is NO. The value REQUEUE=No is forced in the case of a submit containing the MAXDELAY parameter.</p>
RETAIN = Yes <u>No</u> Initial	<p>Specifies whether or not IBM Connect:Direct keeps a copy of the Process in the HOLD queue after the Process has executed. This parameter is optional.</p> <p>Yes keeps a copy of this Process in the HOLD queue after the Process executes. The copy of the Process does not execute until you release it through a CHANGE Process command. If you specify RETAIN=YES, IBM Connect:Direct automatically holds the Process until you release it, unless you include the STARTT parameter in your Process. Use RETAIN in conjunction with STARTT to cause a Process to run repeatedly at a given time each day.</p> <p>No specifies that IBM Connect:Direct is to delete the Process after execution. The default value for RETAIN is NO. The value RETAIN=No is forced in the case of a submit containing the MAXDELAY parameter.</p> <p>Initial specifies that IBM Connect:Direct is to execute the Process every time IBM Connect:Direct is initialized.</p> <p>Note: Do not code the STARTT parameter with the RETAIN=I parameter.</p>
SACCT = 'snode accounting data'	<p>Specifies the accounting data for the SNODE. The maximum length of the accounting data is 256 characters. If you include special characters with the accounting data, you must enclose the string in single quotation marks. This parameter is optional.</p> <p>This data overrides any accounting data specified on the SIGNON command and any accounting data specified in the Process statement.</p>

Parameter	Description
<p>SNODE = secondary node SNODE=TCPNAME = tcpvalue;port</p>	<p>Specifies the secondary node used in this Process. This parameter is optional.</p> <p>secondary node name is a 1-16 alphanumeric name that is defined in the network map.</p> <p>The following characters are allowed: A-Z, 0-9, !, @, #, \$, %, &, {, }, +, -, ^</p> <p>Connect:Direct for z/OS does not accept the following characters for the SNODE: (,) =, \, ", ' , <, >, , </p> <p>Use SNODE=TCPNAME=tcpvalue to specify TCP/IP connections that are not defined in the network map. tcpvalue can be a DNS name up to 255 characters or a 15-character IPv4 or 39-character IPv6 TCP/IP address.</p> <p>port is the TCP/IP port number. It can be up to 5 characters long.</p> <p>Note: If the TCPNAME keyword is used and the port is not specified, the TCP.IP.DEFAULT entry is used if the NODE is not defined in the Netmap.</p>
<p>SNODEID = (ID, pswd, newpswd)</p>	<p>Specifies security user IDs and security passwords at the SNODE.</p> <p>ID specifies the security ID that IBM Connect:Direct passes to a security exit for validation on the SNODE side of the Process. It can contain 1-64 alphanumeric characters.</p> <p>pswd specifies the current security password. The security exit uses this parameter to validate the current security password on the SNODE side of the Process. It can contain 1-64 alphanumeric characters. In the case where the SNODE can Process a PassTicket password, the PNODE generates a PassTicket when only a SNODE user ID override is specified. The actual generation is contingent upon the information shared in the PNODE Authorization File and the option that generates the stage 2 security exit. The <i>IBM Connect:Direct for z/OS Administration Guide</i> contains details for the Authorization File and stage 2 security exit characteristics.</p> <p>newpswd specifies the new security password. The security exit uses this parameter to change the current security password to the new security password. The range is 1-64 alphanumeric characters.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	<p>Specifies that IBM Connect:Direct not execute the Process until a specified date or time. The date, day, and time are positional parameters. If you do not specify the date or day, place a comma before the time. This parameter is optional.</p> <p>Note: Do not code STARTT with RETAIN=INITIAL.</p> <p>date specifies the date to execute the Process. You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). You can use periods or backslashes (/) to separate the components of a date value. You can omit the separators only for transfers between mainframe nodes. Use separators to guarantee transfers between all platforms.</p> <p>You can use the following date formats, according to which date order is specified in the DATEFORM initialization parameter:</p> <ul style="list-style-type: none"> • DATEFORM=MDY specifies the date format as mm/dd/yy, mm/dd/yyyy, mm.dd.yy, or mm.dd.yyyy • DATEFORM=DMY specifies the date format as dd/mm/yy, dd/mm/yyyy, dd.mm.yy, or dd.mm.yyyy • DATEFORM=YMD specifies the date format as yy/mm/dd, yyyy/mm/dd, yy.mm.dd, or yyyy.mm.dd • DATEFORM=YDM specifies the date format as yy/dd/mm, yyyy/dd/mm, yy.dd.mm, or yyyy.dd.mm <p>Valid Julian dates formats are yyddd, yyyyddd, yy/ddd, yyyy/ddd, yy.ddd, or yyyy.ddd</p> <p>If you specify a date without a time, the time defaults to 00:00.</p> <p>If RETAIN=YES, you cannot specify a date in the STARTT parameter.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM]) (continued)	<p>day specifies the day of the week that IBM Connect:Direct is to release the Process for execution. Valid names are MOnday, TUESday, WEdnesday, THursday, FRiday, SATurday, and SUNday. You can also specify TODAY, which releases the Process for execution today, or TOMORROW, which releases the Process for execution the next day.</p> <p>If you specify the day of the week with RETAIN=YES, the Process executes the same day every week.</p> <p>If you specify a day without a time, the time defaults to 00:00. A time of 00:00 means that if you submit a Process on Monday, with Monday as the only STARTT parameter, the Process does not run until the following Monday.</p> <p>hh:mm:ssXM indicates the time of day in hours (hh), minutes (mm), and seconds (ss) that IBM Connect:Direct is to release the Process. You can set XM to AM or PM, or you can omit it if you are using a 24-hour clock. You need not specify minutes and seconds.</p> <p>You can express the time of day using the 24-hour clock or the 12-hour clock. If you do not use AM and PM, IBM Connect:Direct assumes the 24-hour clock. If you use the 12-hour clock, you can express 01:00:00 hours as 1:00A, and 13:00 hours as 1PM. If you use the 24-hour clock, valid times are 00:00-24:00.</p> <p>You can also specify NOON, which releases the Process for execution at noon, or MIDNIGHT, which releases the Process for execution at midnight.</p> <p>If you code hh:mm:ssXM with RETAIN=YES, IBM Connect:Direct schedules the Process for execution the same time every day.</p> <p>Note: When you specify both HOLD=YES and a STARTT value, the HOLD specification takes precedence. IBM Connect:Direct places a Process submitted with HOLD=YES on the hold queue even if you specify a start time.</p>
&symbolic 1 = variable string 1 &symbolic 2 = variable string 2... &symbolic N = variable string N	<p>Specifies the default value substituted for the symbolic parameter in the Process. If you use a symbolic in the SUBMIT command, it overrides any default values you specify in the Process statement.</p> <p>Specify a null value by immediately following the equal sign with a comma. You must enclose a symbolic parameter containing special characters in single quotation marks.</p> <p>You can set an ampersand (&) symbolic parameter to another single ampersand symbolic parameter that is resolved during the first Process submission. Do not use identical symbolic names. The maximum variable string length is 256 characters.</p>

SUBMIT Command Examples

Use these examples to help you gain a basic understanding of how you can use the SUBMIT command.

Process Example

The following example shows the syntax for a Process named PAYROLL.

```
SUBMIT PROC=PAYROLL -  
  HOLD=Y RETAIN=Y -  
  PACCT='1959, DEPT 27' -  
  SACCT='1962, DEPT 24' -  
  CASE=YES
```

In this example, the SUBMIT command was used to:

- Place the Process (named PAYROLL) in the HOLD queue.
- Retain a copy after it is released for execution.
- Create case sensitivity for fields associated with accounting data, user ID, password, and data set name in the SUBMIT command and in the PAYROLL Process.

Symbolic Substitution Example

In the following example, IBM Connect:Direct resolves the symbolic &DSN in the Process COPYSEQ upon submission:

```
SUBMIT PROC=COPYSEQ -  
  SNODE=CHICAGO, &DSN=MYFILE
```

Submitting Processes through the Batch Interface

About this task

To submit a Process through the batch interface, perform these steps:

Procedure

1. Place your SUBMIT command in the DGADBATC job stream as described in [Introduction to the Batch Interface](#).
2. Submit the job while IBM Connect:Direct is running.

Note: If the Extended Submit Facility (ESF) is available, the DTF does not need to be running to execute your command.

3. Verify your results.

Submitting Processes through the IUI

The Interactive User Interface (IUI) enables you to create, edit, and submit Processes using the SB, DF, and CF options available on the PRIMARY OPTIONS MENU.

IUI Submit Options

The SB, DF, and CF options available from the IUI enable you to perform the following submit options for Processes:

Option	Description
SB	Enables you to submit Processes from the IBM Connect:Direct PUBLIB or a library you specify on the screen. You can also override certain parameters in the existing Process.
DF	Enables you to edit, create, and submit Processes residing in the IBM Connect:Direct PUBLIB or a library you specify on the screen. You edit or create a Process using the ISPF editor.
CF	Enables you to dynamically build a COPY Process through fill-in-the-blank panels and submit it.

Validating Processes from the SB, DF, and CF IUI Options

From any of the IUI SUBMIT screens available through the SB, DF, and CF options, you can validate the Process syntax without submitting the Process, or you can submit the Process, which includes verifying the syntax, as explained in the following table:

Operation	Related Parameter or Command	Result
Validate Process syntax	<ul style="list-style-type: none">• Enable VALIDATE ONLY parameter• Issue SUBV command	<ul style="list-style-type: none">• The Process is displayed in the VIEW PROCESS screen.• If the Process is valid, the message Submit Process for validation command was successful is displayed.• If the Process is invalid, a message identifies the problems in the Process. You can press PF1 for information to troubleshoot the problem.• You can press PF3 to return to the primary SB, DF, or CF screen.
Submit Process and validate syntax	<ul style="list-style-type: none">• Issue SUBMIT or SUB command	<ul style="list-style-type: none">• If the Process is valid, it is executed and a Process number is displayed for reference.• If the Process is invalid, a message describes the problem. You can press PF1 for information to troubleshoot the syntax errors.

Using the SB IUI Option to Submit a Predefined Process

About this task

The Interactive User Interface (IUI) enables you to submit predefined Processes using the SB option, which is available from the PRIMARY OPTIONS MENU (see [Primary Options Menu](#)). You can submit a Process from the IBM Connect:Direct PUBLIB or a library you specify and override certain parameters in the existing Process.

To submit a predefined Process from the IBM Connect:Direct IUI:

Procedure

1. Select option **SB** from the Primary Options Menu to display the SUBMIT PROCESS screen.

```
If you enable the SECURITY OVERRIDE parameter, type the security override
information in the IBM Connect:Direct
SECURITY OVERRIDE screen.
node.name          Connect:Direct SECURITY OVERRIDE          TIME: hh:mm
CMD==>
PRIMARY NODE:
SECURITY ID: -----
PASSWORD  :
SECONDARY NODE:
SECURITY ID: -----
PASSWORD  :
```

2. Type the name of the Process to submit.
3. Take one of the following actions:
 - To submit the Process, press **Enter**.

- To define additional parameters, continue with step 4.
4. Type any additional parameters of your SUBMIT command and press **Enter**.

For SUBMIT parameter descriptions not included in the following table, press **PF1** or see [SUBMIT Command](#).

Field	Description
TRANSPORT	Specifies the method of transport to use for the file transfer. NETMAP—Causes IBM Connect:Direct to search in the network map for the sending and receiving nodes. If you type a Transport of NETMAP, the SNODE keyword remains SNODE=secondary node. TCP—Indicates that the NODE NAME field contains an alias name or an IP address. IBM Connect:Direct does not look in the network map for node information. If you type TCP in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcpvalue. UDT33—Indicates that the NODE NAME field contains an alias name or an IP address. IBM Connect:Direct does not look in the network map for node information. If you type UDT33 in the Transport field, the SNODE keyword becomes SNODE=UDT33NAM=udtvalue. DNS—Enables you to enter a domain name up to 255 bytes.
SECURITY OVERRIDE	Enables you to override security information such as password and user ID for the primary and secondary node. Yes = Enable security override. No = Do not override security information.
VALIDATE ONLY	Enables you to validate the Process syntax without submitting the Process. Yes = Validate without submitting the Process. No = Do not validate without submitting the Process.
CASE SENSITIVITY	Specifies whether parameters associated with accounting data, user ID, password, and data set name in the Submit Process are to be case sensitive. The default is NO , which does not allow mixed case.

5. If you enable the SECURITY OVERRIDE parameter, type the security override information in the IBM Connect:Direct SECURITY OVERRIDE screen.

```

If you enable the SECURITY OVERRIDE parameter, type the security override
information in the IBM Connect:Direct
SECURITY OVERRIDE screen.
node.name          Connect:Direct SECURITY OVERRIDE          TIME: hh:mm
CMD==>
PRIMARY NODE:
  SECURITY ID: -----
  PASSWORD   :
SECONDARY NODE:
  SECURITY ID: -----
  PASSWORD   :

```

6. Press **Enter** to perform one of the following actions:
- Submit the Process. See the table in [Operation](#) for the possible results.
 - View the results of the validate-only operation in the VIEW PROCESS panel, if you enabled the VALIDATE ONLY parameter. See the table in [Operation](#) for the possible results.



CAUTION: Press **PF3** to exit the screen from which you submitted the Process. Each time you press **Enter**, the Process is submitted.

7. If you performed the validate-only operation, press **PF3** to return to the SUBMIT PROCESS screen.

Using the DF IUI Option to Create, Edit, and Submit Processes

The Interactive User Interface (IUI) enables you to modify, create, and submit Processes using the DF option, which is available on the PRIMARY OPTIONS MENU. Using the DF option, you can:

- Modify a Process
- Create a new Process
- Submit a Process
- Copy statement models to create a Process

The IBM Connect:Direct Public Process Library (PUBLIB) contains Process statement models that you can use as templates for creating IBM Connect:Direct Processes. The DF option enables you to copy Process statement models, edit them for your environment using the ISPF editor, and submit Processes residing in the IBM Connect:Direct PUBLIB or in a library you specify.



CAUTION: Do not edit a sample member directly. Keep the source as is, edit a new name, and then copy the member into the new one. Save as a new name. The SDGAPROC Process library contains alias names that if edited and saved will break the aliases.

The PUBLIB contains commented files and uncommented files. File names of commented files are preceded by the at sign (@); file names with the pound sign (#) prefix do not include comments.

See [Validating Processes from the SB, DF, and CF IUI Options](#) for details about how to validate the syntax of a Process with and without submitting it.

Modifying and Submitting a Process Using the DF IUI Option

About this task

To modify a Process using the ISPF editor and submit the Process to the DTF:

Procedure

1. Choose option **DF** from the Primary Options Menu to display the Connect:Direct PROCESS DEFINITION screen.
2. Type the member name of the Process.
3. Verify the library location and press **Enter**.

The PROCESS LIBRARY NAME field is required. If the member is located in the IBM Connect:Direct PUBLIB, type the library name that is displayed in the PUBLIC PROCESS LIB field in the PROCESS LIBRARY NAME field.

4. After IBM Connect:Direct displays the Process in ISPF edit mode, make the desired changes.
5. Press **PF3** to return to the PROCESS DEFINITION screen.
6. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
7. Take one of the following actions:
 - To validate the Process without submitting it, type **SUBV** and refer to the table in [“Validating Processes from the SB, DF, and CF IUI Options”](#) on page 570 for information on the results.
 - To submit the Process, type **SUBMIT** (or **SUB**) on the command line and press **Enter**.

Creating and Submitting a New Process Using the DF IUI Option

About this task

To create and submit a new Process:

Procedure

1. Choose option **DF** from the Primary Options Menu to display the Connect:Direct PROCESS DEFINITION screen.
2. Type the member name of the new Process.
3. Specify the name of the library where you want to save the new Process and press **Enter**.
The member location can be your private Process library (PROCESS LIBRARY NAME) or the IBM Connect:Direct PUBLIC PROCESS LIBRARY.
4. After IBM Connect:Direct displays an ISPF editor screen, type your new Process.
5. Press **PF3** to save the Process in the PROCESS LIBRARY you specified and return to the PROCESS DEFINITION SCREEN.
6. To validate the Process without submitting it, type **SUBV** on the command line and press **Enter**. See the table in [“Validating Processes from the SB, DF, and CF IUI Options” on page 570](#) for information about the results.
7. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
8. To submit the Process, type **SUBMIT** (or **SUB**) in the command line and press **Enter**.

Submitting a Process Using the DF IUI Option

About this task

To submit a Process to the DTF:

Procedure

1. Choose option **DF** from the **Primary Options Menu** to display the Connect:Direct PROCESS DEFINITION screen.
2. Type **SUBMIT** (or **SUB**) on the command line.
3. Type the member name of the Process.
4. Verify that the library name that contains the member you want to submit is displayed in the PROCESS LIBRARY NAME field. If the member is located in the IBM Connect:Direct PUBLIB, type the library name that is displayed in the PUBLIC PROCESS LIB field in the PROCESS LIBRARY NAME field.
5. Specify case sensitivity. Case sensitivity applies to the SUBMIT command and the Process.
6. Press **Enter**. See the table in [“Validating Processes from the SB, DF, and CF IUI Options” on page 570](#) for information about the results.

Using Statement Models to Create a Process

About this task

To copy a Process statement model to a new PDS member:

Procedure

1. Choose option **DF** from the Primary Options Menu to view the IBM Connect:Direct PROCESS DEFINITION screen.
2. Type the name of the new PDS member.
Note: The PUBLIC PROCESS LIB displays the partitioned data set (PDS) allocated by the signon CLIST or the TSO logon procedure.
3. Type the PROCESS LIBRARY NAME and press **Enter**.
4. Type **COPY** on the command line at the top of the blank member and press **Enter**.

5. When the ISPF Edit/View-Copy screen is displayed, type the member name of the PROCESS statement model you want to copy in the DATA SET NAME field and press **Enter**.

For example, to create your Process, first copy the PROCESS statement. Specify @PROCESS to use commented models from the PUBLIB, or #PROCESS to use uncommented models in your Process.

6. To add a statement model, type **a** in the first column of the last line of the new member and press **Enter**.
7. Repeat steps 4 through 6 to copy additional statement models into the member following the PROCESS statement.
8. Edit the statement model or models for your environment according to the following guidelines:
 - Replace underscores with the appropriate parameter values.
 - Provide an appropriate Process name in the PROCESS statement.
 - Delete all lines that are not applicable.
 - Continuation marks are necessary on all but the last line of each statement model.
 - You can delete comment lines. They are optional.
9. Press **PF3** to save the changes and return to the IBM Connect:Direct PROCESS DEFINITION screen.
10. Take one of the following actions:
 - To validate the Process without submitting it, type **SUBV** and press **Enter**. See the table in [Operation](#) for information about the results.
 - To submit the Process, type **SUBMIT** on the command line and press **Enter**. See the table in [Operation](#) for information about the results

Using the CF IUI Option to Generate a COPY Process

The CF option of the Interactive User Interface (IUI) enables you to dynamically generate a COPY Process through fill-in-the-blank panels and submit it.

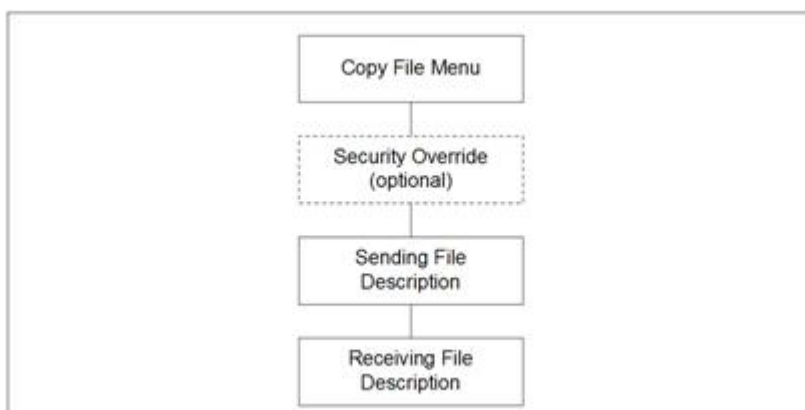


CAUTION: The Process generated through this option is a one-time only Process and there are no facilities to save the Process.

The COPY FILE Menu of the CF option displays a series of screens that collect information to generate a SUBMIT PROCESS command. IBM Connect:Direct submits the predefined Process, COPYCF, with the appropriate command parameters and variable substitutions based on your input. You can use four screens to create the COPYCF Process:

Screen Name	Description
COPY FILE Menu	Collects information for the SUBMIT COMMAND parameters, PNODE/SNODE information, data compression, and checkpoint information.
Security Override Menu	Collects information about user ID, password, and accounting data. This optional panel is displayed only if you request it on the first panel.
SENDING FILE Menu	Collects information for the Process variables for the sending file.
RECEIVING FILE Menu	Collects the information for the Process variables for the receiving file.

The following diagram illustrates the relationship of these four screens.



See [Validating Processes from the SB, DF, and CF IUI Options](#) for details about how to validate the syntax of a Process with and without submitting it.

Generating a COPY Process

About this task

To generate a COPY Process:

Procedure

1. Select the **CF** option from the Primary Options Menu to display the COPY FILE Menu.

```

node.name          Connect:Direct Copy File Menu          TIME: hh:mm
CMD ==>

SENDING  ENVIRONMENT: ZOS_____  TRANSPORT: NETMAP
NODE NAME: CD.NODE1_____
RECEIVING ENVIRONMENT: ZOS_____
NODE NAME: CD.NODE2_____
PROCESS NAME: COPYCF__
CLASS: _____ (NUMERIC)          VALIDATE ONLY:      N
HOLD:      Y____ (Y, N, OR C-CALL)    ZFBA:            -
NOTIFY USER: %USER_____
PRIORITY: _____ (RANGE: 0 TO 15)    FASP:            ____
REQUEUE:    N____ (Y OR N)             FASP Policy:     ____
RETAIN ON TCQ: N____ (Y, N, OR I-INITIAL) FASP Bandwidth:  ____
START DATE: _____ TIME: _____ FASP Filesize:   ____
CHECKPOINT: _____ (BYTE INTERVAL - NK|NM)
DEBUG:      _____ (HEXADECIMAL)
PLEXCLASS:  _____ (PNODE SNODE)
COMPRESS:    N____ (Y, N, X-EXTENDED, X'XX', OR C'C')
EXT COMPRESSION LEVEL: _ WINDOW: ____ MEMORY: _
OVERRIDE SECURITY: N (Y OR N)

DO YOU WANT VALUES FOR THIS COPY TO BE CASE SENSITIVE?:      NO____
  
```

2. If necessary, rename the COPYCF Process by typing a name in the **PROCESS NAME** field.
3. For descriptions of SUBMIT parameters not included in the following table, press **PF1** or see [SUBMIT Command](#).

Parameter Name	Description
SENDING ENVIRONMENT	Specifies the operating system or platform from which the file is transmitted. The values you place in the Sending Environment and Node Name fields determine which Sending File screen is displayed.
NODE NAME	Specifies the name of the IBM Connect:Direct node from which the file is transmitted.

Parameter Name	Description
TRANSPORT	<p>Specifies the method of TRANSPORT IBM Connect:Direct uses for the file transfer.</p> <p>NETMAP—Causes IBM Connect:Direct to look in the network map for the sending and receiving nodes. If you type a Transport of NETMAP, the SNODE keyword remains SNODE=secondary node.</p> <p>TCP—Indicates that the NODE NAME field contains an alias name or an IP address. IBM Connect:Direct does not look in the network map for node information.</p> <p>UDT33—Indicates that the NODE NAME field contains an alias name or an IP address. IBM Connect:Direct does not look in the network map for node information. If you type UDT33 in the Transport field, the SNODE keyword becomes SNODE=UDT33NAM=udtvalue.</p> <p>DNS—Enables you to enter a domain name up to 255 bytes</p> <p>Note: If you type TCP or DNS in the Transport field, the SNODE keyword becomes SNODE=TCPNAME=tcvalue.</p>
RECEIVING ENVIRONMENT	Specifies the operating system or platform to which the file is transmitted. The values you place in the Receiving Environment and Node Name fields determine which Receiving File screen is displayed.
NODE NAME	Specifies the name of the node to which the file is transmitted.
PROCESS NAME	(optional) Specifies the name to be used under which the Process will be submitted. If you do not type a Process name, IBM Connect:Direct provides the default name COPYCF.
VALIDATE ONLY	<p>Enables you to validate the Process without submitting it.</p> <p>YES = Validate the Process, but do not submit it.</p> <p>NO = Do not validate before the Process is submitted.</p>
FASP	Optional: Specifies whether Connect:Direct should request SSP to use the FASP protocol to the SNODE. Valid only when there is a FASP capable SSP and the SNODE is also FASP capable otherwise it is FASP=NO. Valid values are NO or SSP.
FASP POLICY	Optional: Specifies the FASP Policy that SSP is to attempt for the FASP session. Valid values are FIXED or FAIR HIGH or LOW.
FASP BANDWIDTH	Optional: Specifies the FASP Bandwidth that SSP is to attempt for the FASP session. Valid values are specified as nnn or nM or nG
FASP FILESIZE	Optional: Specifies the FASP Filesize Threshold to limit the size of file that can use the FASP protocol. File sizes that are less than the threshold will be processed as FASP=NO.
ZFBA	Specifies the number of zFBA devices to allocate to the Process. The only acceptable value is 2 and this ZFBA feature requires additional configuration settings in the NETMAP and initialization parameters. Consult the <i>Connect:Direct Administration Guide</i> for Using zFBA for File Transfer.

Parameter Name	Description
CHECKPOINT	(optional) Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid transmission point, avoiding the need to restart transmission from the beginning. K denotes thousands; M denotes millions. A checkpoint value of zero stops automatic checkpointing.
COMPRESS	(optional) Specifies whether and how the data being transmitted should be compressed. Y – Uses compression with X'40' (blank) as the PRIMEchar. N – Does not use compression. X – Uses Extended compression. X'xx' – Uses compression with the specified 'xx' hex vale. C'c' – Uses compression with the specified 'cc' character value. Note: Compression is CPU-intensive, and its effectiveness is data dependent. It should only be used if its benefits are known.
EXT COMPRESSION LEVEL	Note: The effects of changing the default values for the extended compression parameters (level, window, and memory) are not always predictable and can significantly increase CPU utilization. Before changing the default values, see Improving Performance in the <i>IBM Connect:Direct for z/OS Administration Guide</i> . (optional) Specifies the value from 1–9 indicating the degree of compression to use. The default is 1 , which usually provides sufficient compression.
WINDOW	(optional) Specifies the value from 8–15 indicating the size of the compression buffer to use. The default of 13 equals 32 KB.
MEMORY	(optional) Specifies the value from 1–9 indicating the amount of memory used to maintain the compression state. The default of 4 equals 8 KB.
OVERRIDE SECURITY	Enables you to change security information for the sending and receiving nodes. YES = Enable NO = Disable
CASE SENSITIVITY	(optional) Specifies the case sensitivity not only to the SUBMIT command, but to the COPY Process itself. IBM Connect:Direct provides for mixed case user input because some platforms allow it. The default is NO , which does not allow mixed case.

4. To override security information, type **Y** in the OVERRIDE SECURITY field.

```

node.name          Connect:Direct SECURITY OVERRIDE          TIME: hh:mm
CMD ==>
SENDING  ENVIRONMENT: ZOS          TRANSPORT: TCP
NODE NAME:  CD.NODE1
SECURITY ID:  -----
PASSWORD   :  -----
ACCOUNTING DATA:  -----
RECEIVING ENVIRONMENT: ZOS
NODE NAME:  CD.NODE2
SECURITY ID:  -----
PASSWORD   :  -----
ACCOUNTING DATA:  -----

```

5. In the SECURITY OVERRIDE screen:

- a) Specify the security information for the node whose security you want to override and press **Enter**. This procedure assumes that you modify security information for both nodes.
- b) Press **PF1** for a description of each field.

Note: For a complete description of the valid parameters of a COPY Statement and examples to help you fill in both the SENDING FILE and RECEIVING FILE screens, see the [Connect:Direct Process Language](#) help.

6. When the SENDING FILE screen for the environment and node name you specified is displayed, type appropriate values and press **Enter**.

Note: If the data set name does not follow z/OS naming conventions, enclose the data set in single quotation marks. An HFS file must begin with a slash (/) and can contain up to 251 bytes.

```

node.name          COPYFILE - SENDING FILE (z/OS or OS/390)          TIME: hh:mm
CMD ==>
NODE NAME: CD.NODE2
SENDING DSNAME: _____
_____
UNIT PARAMETER: ( _____ )
LABEL PARAMETER: ( _____ )
VOLUME SERIAL(S): ( _____ )
RETAI: (Y OR N)
DCB PARAMETER: ( _____ )
TYPE KEY: _____ MSVGP NAME: _____
SYSOPTS: _____
(PDS ONLY:) _____ REPLACE: Y (Y OR N) ALIAS: Y (Y OR N)
SELECTION CRITERIA: ( _____ )
EXCLUSION CRITERIA: ( _____ )

```

7. When the RECEIVING FILE screen for the environment and node name you specified is displayed, type appropriate values and press **Enter**.

Note: If the data set name does not follow z/OS naming conventions, enclose the data set in single quotation marks. An HFS file must begin with a slash (/) and can contain up to 251 bytes.

```

node.name          COPYFILE - RECEIVING FILE (z/OS or OS/390)        TIME: hh:mm
CMD ==>
NODE NAME: Q1A.ZOS.V4600
RECEIVING DSNAME: _____
_____
DISPOSITION: ( NEW , CATLG_ , _____ )
UNIT PARAMETER: ( _____ )
VOLUME SERIAL(S): ( _____ )
RETAI: (Y OR N) COUNT: ____ (1-255)
DCB PARAMETER: ( _____ )
LABEL PARAMETER: ( _____ )
SPACE: ( _____ )
TYPE KEY: _____ MSVGP NAME: _____
SYSOPTS: _____
SMS: DATA CLASS: _____ STORAGE CLASS: _____ MGMT CLASS: _____
AVERAGE RECORDS: _____ DSNTYPE : _____ EATTR: _____
VSAM ORGANIZATION: _____ KEY LENGTH: _____ KEY OFFSET: _____
LIKE DSNAME: _____
SECURITY MODEL: _____
GENERIC MODEL: _____ (YES OR NO)

```

8. Press **Enter**.

- If you enabled the Validate Only parameter, a VIEW PROCESS screen with your Process is displayed. See the table in [Operation](#) for information on the results.
- If you did not select the Validate Only option, the Process is executed. See the table in [“Validating Processes from the SB, DF, and CF IUI Options”](#) on page 570 for information on the results.



CAUTION: Press **PF3** to back out of each screen until the Primary Options Menu is displayed. Each time you press **Enter**, the Process is submitted.

Controlling Processes in the TCQ

Controlling Processes with Commands

IBM Connect:Direct Processes consist of statements with parameters that provide instructions for copying files, running jobs and programs, and altering the sequence of Process step execution. Use the Process Control commands to manipulate these Processes while they are in the TCQ.

Note: You can also use the Connect:Direct Browser User Interface to perform some Connect:Direct for z/OS procedures.

Setting Selection Criteria

The Process control commands have common parameters and IUI screen layouts. All Process control commands allow you to select Processes by Process name, number, and submitter. (Some allow additional criteria.) The WHERE parameter groups the selection arguments, enabling you to make common changes or selections. The command applies to all Processes which match the criteria. For example, if you indicate a Process name PROCA with no further qualification, all Processes named PROCA are affected by the command.

When you specify two or more WHERE subparameters, you further qualify the selection so that all conditions must be satisfied. In this way, you can adequately qualify your selection if you have non-unique Process names or numbers. For example, if you specify in your selection criteria a Process name of PROCA and Process number of 16, the Process number of PROCA must be 16 for the command to apply.

If you specify multiple Processes by using the list option on the subparameter, the position of the arguments in the list is significant. Elements in each list must correspond by position. For example, PNAME=(PROCA, PROCB), PNUM=(16,17) requires that PROCA be Process number 16 and PROCB be Process number 17 for the command to apply.

On the screens, you can type a list of Process names, numbers, and submitters (user ID and node ID) to obtain the same results that the WHERE parameter list provides in batch. IBM Connect:Direct builds the appropriate command from the screen. IBM Connect:Direct accesses the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command.

Modifying a Process in the TCQ with CHANGE PROCESS

Use the CHANGE PROCESS command to modify the parameters of a Process when the Process is in the TCQ in a nonexecuting state. Release a held Process or restart a failed Process with this command. (You can also release a held Process through the Operator Table and Selected Process screens by typing an R next to the Process Name.)

CHANGE PROCESS Command Format

The CHANGE PROCESS command has the following format and associated parameters. Required parameters and keywords are in bold print.

Label	Command	Parameters
(optional)	CHange PROCESS	<p>WHERE (NODE= node name (list) PNAME= name (list) PNUMber= number (list) SUBmitter= (node name, user ID) (list)) CASE= Yes No CLASS= n DEBUG= trace bits DEST= destination node HOLD= Yes No Call NETMAP.REFRESH PLEXCLASS=(pname class, snode class) PRTY= n RELEASE RESTART= [NO FIRST = volume sequence number FIRST = SER = volume serial number LAST = volume sequence number LAST = SER = volume serial number VOLCNT = n] RETAIN= Yes No Initial STARTT= ([date day] [,hh:mm:ssXM])</p>

The following table describes the parameters for the CHANGE PROCESS command:

Parameter	Description
<p>WHERE (NODE = node name (list) PNAME = name (list) PNUMber = number (list) SUBmitter = (node name, user ID) (list))</p>	<p>Specifies which Process(es) to change. Name multiple Processes in the command using the selection criteria method described on “Setting Selection Criteria” on page 579 if you have to make similar changes to many Processes. This parameter is required.</p> <p>The selection subparameters are optional; however, you must specify at least one of NODE, PNAME, PNUMBER, or SUBMITTER. These subparameters are special in that they identify which Processes are selected for the change activity defined by the other parameters.</p>
NODE = node name (list)	<p>Specifies the snode or a list of snodes where the Processes to be changed are running. If you specify a list of snodes, enclose them in parentheses separate each snode with a comma.</p> <p>You can use this subparameter with the HOLD parameter to put all Processes on a node on hold if the node is unavailable, or if problems exist with the node. You can restart all Processes later using the RESTART parameter with the NODE= subparameter.</p>
PNAME = name (list)	Specifies the name of the Process to be changed or a list of Process names enclosed in parentheses and separated by commas. This parameter is optional.
PNUMber = number (list)	Specifies the number of the Process to be changed or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999.

Parameter	Description
SUBmitter = (node name, user ID) (list)	Specifies the node name and user ID of the user that submitted the Process to be changed. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.
CASE = Yes No	Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not to the Process itself. See Indicating Case Sensitivity for a general overview of case sensitivity. This parameter is optional.
CLASS = n	Determines the node-to-node session on which a Process executes. Consult your IBM Connect:Direct administrator for instructions concerning which class to specify. The range is 1-255. This parameter is optional.
DEBUG= trace settings for this Process	<p>Specifies the 8-position trace setting for this Process. This parameter enables you to specify a trace for only this Process. The table on Building, Modifying, and Submitting Processes lists acceptable trace values. This parameter is optional.</p> <p>Note: This option can generate a large amount of output and should be used with caution.</p>
DEST = destination node	Specifies a new destination node. This parameter changes the node that this Process communicates with. This parameter is optional.
HOLD = Yes No Call	<p>Specifies whether or not the Process is placed in the hold queue. This parameter is optional.</p> <p>Yes specifies that the Process remains in the hold queue until one of the following events occurs:</p> <ul style="list-style-type: none"> • CHANGE PROCESS releases the Process • DELETE PROCESS deletes the Process <p>When you specify both HOLD=YES and a STARTT value, the HOLD specification takes precedence. Therefore, IBM Connect:Direct places a Process submitted with HOLD=YES on the hold queue even if you specified a start time.</p> <p>No specifies that IBM Connect:Direct does not place the Process in the hold queue, but places it in the WAIT for execution queue (EX). HOLD=NO is the default.</p> <p>Call specifies that IBM Connect:Direct is to place the Process in the hold queue until a session is established with the specified SNODE. This session could be established by either another Process starting on the PNODE destined for the same SNODE or the SNODE contacting the PNODE. For example, a Process submitted HOLD=NO establishes a session and causes execution of any Processes residing on the SNODE destined for this node that are submitted with HOLD=CALL.</p> <p>Note: IBM Connect:Direct ignores the HOLD parameter if RETAIN=Y.</p>

Parameter	Description
NETMAP.REFRESH	<p>Refreshes the network map for Processes waiting execution in the Wait queue, after an UPDATE NETMAP command is issued for this node. (For a description of the UPDATE NETMAP command, see the <i>IBM Connect:Direct for z/OS Administration Guide</i> and search on Maintaining the Network Map. This parameter is optional.</p> <p>Do not use the NETMAP.REFRESH parameter if the network map protocol is changed (for example, changing a node from LU0 to LU6.2). If you change the protocol for a particular node, you must delete and resubmit all waiting Processes for that node.</p>
PLEXCLASS = (pnode class, snode class)	<p>Specifies the class that directs the Process to only certain servers in a IBM Connect:Direct/Plex. This parameter does not apply to a IBM Connect:Direct Stand-alone Server. This parameter is optional.</p> <p>You can designate each server in a IBM Connect:Direct/Plex to support only certain PLEXCLASSES through the CDPLEX.PLEXCLASSES initialization parameter. You can then limit Processes to only those servers by specifying the PLEXCLASS in the Process definition.</p> <p>The pnode class controls which IBM Connect:Direct Server runs the Process. The snode class controls what other node is used with the Process.</p> <p>The pnode class and snode class are each 1-8 characters long. An asterisk (*) indicates that the Process runs on any server with an asterisk designated in the CDPLEX.PLEXCLASSES initialization parameter. If you do not specify a PLEXCLASS, the network map is checked for a default PLEXCLASS. If the network map does not specify a default PLEXCLASS, then an asterisk is used as the default.</p> <p>If a Process must run on a specific IBM Connect:Direct Server, specify the IBM Connect:Direct Server name in this field. The Process only runs on that server.</p>
PRTY = n	<p>Specifies the Process priority in the Transmission Control Queue. High numbers indicate high priorities; low numbers indicate low priorities. IBM Connect:Direct uses this priority only for Process selection; it does not affect the priority during transmission. The default is set by the PRTYDEF global initialization parameter. The range is from 0-15. This parameter is valid for LU0 only. This parameter is optional.</p>
RELEASE	<p>Releases the Process for execution from the queue where it is currently residing. Either specify RELEASE or omit it. Releases or RELEASE=no is not valid. This parameter is optional.</p>

Parameter	Description
<p>RESTART = [NO FIRST = volume sequence number FIRST=SER=volume serial number </p> <p>LAST = volume sequence number </p> <p>LAST = SER = volume serial number </p> <p>VOLCNT = n]</p>	<p>Specifies the conditions for restarting an interrupted Process.</p> <p>Use the CHANGE PROCESS command to cause IBM Connect:Direct to restart a data transmission at the last checkpoint position taken before the interruption or at a previous checkpoint position. IBM Connect:Direct restarts the copy step at the position you specify. This parameter is optional.</p> <p>NO specifies that the copy step restarts at the beginning of the transmission.</p> <p>FIRST=volume sequence number specifies that the copy step restarts at the beginning of the volume designated by the volume sequence number.</p> <p>FIRST=SER=volume serial number specifies that the copy step restarts at the beginning of the volume serial given.</p> <p>LAST=volume sequence number specifies that the copy step restarts at the end of the volume designated by the volume sequence number. LAST specifies that the copy step restarts at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.</p> <p>LAST=SER=volume serial number specifies that the copy step restarts at the end of the volume serial given. LAST specifies that the copy step restarts at the last block on the volume if the output is disk or the last checkpoint on the volume if the output is tape.</p> <p>VOLCNT=n specifies that the volume count on the interrupted copy step is changed to the value specified. You can use this parameter to increase the number of output volumes if the copy step is interrupted because the volume count of that step is too small.</p>
<p>RETAIN = Yes No Initial</p>	<p>Specifies whether or not IBM Connect:Direct keeps a copy of the Process in the HOLD queue after the Process has executed. This parameter is optional.</p> <p>Yes keeps a copy of this Process in the HOLD queue after the Process executes. The copy of the Process does not execute until you release it through a CHANGE Process command. If RETAIN=YES is specified, IBM Connect:Direct automatically holds the Process until you release it, unless you include the STARTT parameter in your Process. Use RETAIN in conjunction with STARTT to cause a Process to run repeatedly at a given interval.</p> <p>No specifies that IBM Connect:Direct is to delete the Process after execution. The default value for RETAIN is NO.</p> <p>Initial specifies that IBM Connect:Direct is to execute the Process every time IBM Connect:Direct is initialized. Do not code the STARTT parameter with the RETAIN=I parameter.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	Specifies that the Process not execute until a specified date or time. See “SUBMIT Command” on page 560 for a complete explanation of the STARTT parameter. This parameter is optional. When changing the date, day, or hh:mm:ssXM , you must respecify even the values that do not change. You cannot specify TODAY or TOMORROW for the day subparameter.

Note: Processes with a QUEUE or STATUS of PR cannot be changed with the CHANGE PROCESS command.

CHANGE PROCESS Command Examples

The following CHANGE PROCESS command places the Processes named PAYROLL and BILLING in the hold queue:

```
CH PROC WHERE ( -
                PNAME=(PAYROLL, BILLING)) -
                HOLD=Y
```

In the following example, the Process named PAYROLL is changed so that the new destination node is DALLAS1 and the Process executes every Friday:

```
CH PROC WHERE (PNAME=PAYROLL) -
                DEST=DALLAS1 -
                RETAIN=Y -
                STARTT=(FR)
```

In the following example, the Process called PAYROLL with the Process number of 60584 has the RETAIN status changed so that the Process is deleted after execution:

```
CH PROC WHERE (PNUM=60584, PNAME=PAYROLL) -
                RELEASE -
                RETAIN=N
```

Issuing CHANGE PROCESS through the Batch Interface

About this task

To use the CHANGE PROCESS command from the Batch Interface:

Procedure

1. Place commands in the DGADBATC job stream as described in [Sample Job Stream to Run the Batch Interface](#).
2. Submit the job while IBM Connect:Direct is running.

A report similar to the following is displayed:

```

=====
                          CHANGE PROCESS
=====
PROCNAME PROCNUM_ SUBMITTER_NODE__ SNODE.NAME_____ MESSAGE_____
USERID_____
-----
DMNOTEST 271          node.name          node.name          DEST CHANGED
USER01
-----

```

3. Check this report to verify your results.

Issuing CHANGE PROCESS through the IUI

About this task

Use the CHANGE PROCESS screen to change a Process and its parameters. To issue the CHANGE PROCESS command from the IBM Connect:Direct IUI, perform the following steps:

Procedure

1. Select option CP from the Primary Options Menu to display the CHANGE PROCESS screen.

```

node.name          CHANGE PROCESS          hh:mm
CMD ==>
PROCESS NUMBERS: ==> _____ ==> _____ ==> _____ ==> _____
PROCESS NAMES:   ==> _____ ==> _____ ==> _____ ==> _____
NODES            ==> _____ ==> _____ ==> _____
USER IDS:         USER NODES:
==> _____
==> _____
==> _____
==> _____
PLEXCLASS         ==> _____ (PNODE SNODE)
DESTINATION NODE  ==> _____ DEBUG ==> _____
PRIORITY          ==> _____ (0 TO 15)
CLASS            ==> _____
HOLD PROCESS      ==> _____ ('Y'-YES, 'N'-NO, 'C'-CALL)
RELEASE PROCESS   ==> _____ ('Y'-YES, 'N'-NO)
RETAIN PROCESS    ==> _____ ('Y'-YES, 'N'-NO, 'I'-INIT)
REFRESH FROM NETMAP ==> _____ ('Y'-YES, 'N'-NO)
SCHED. START DATE ==> _____ TIME ==> _____
RESTART ALTERATIONS ==> _____
DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

IBM Connect:Direct selects the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command. This list fully qualifies your search criteria.

2. Type at least one of the parameters or a combination of parameters as described in the CHANGE PROCESS command syntax section on [Modifying a Process in the TCQ with CHANGE PROCESS](#) or press the **PF1** key for Help.

IBM Connect:Direct changes the parameters for the Processes that match the search criteria. The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

Suspending, Flushing, and Deleting Processes

The SUSPEND PROCESS, FLUSH PROCESS, and DELETE PROCESS commands have similar formats and parameters. The following table describes each command:

Process	Description
SUSPEND PROCESS	Terminates execution and puts an executing Process in the Hold queue. IBM Connect:Direct places the Process in HOLD status. You can restart the Process through the RELEASE parameter of the CHANGE PROCESS command, or by using the R option on the Operator Table screen or the Selected Process screen.
FLUSH PROCESS	Terminates an executing Process. Note that if you specify RETAIN=YES, the flushed Process remains in the TCQ for execution at the next scheduled time. If you specify RETAIN=NO, IBM Connect:Direct removes the Process from the TCQ, and you must resubmit it if you want it to execute.
DELETE PROCESS	Enables you to specify a nonexecuting Process and remove it from the TCQ. Note: FLUSH PROCESS stops an executing Process.

If you do not specify FORCE for the FLUSH or SUSPEND command, an indicator notifies the program executing on behalf of the Process that a FLUSH or SUSPEND command was issued for the Process. If that program is not in control (for example, if it is waiting on a request outside of IBM Connect:Direct code to complete), then it does not see the FLUSH or SUSPEND indicator and the Process is not flushed or suspended; otherwise, the program looks for the FLUSH or SUSPEND indicator and takes the appropriate action.

When you specify FORCE, the action taken depends on the STATE and SUBSTATE of the Process for which you issued the FORCE FLUSH or FORCE SUSPEND. Determine the STATE and SUBSTATE of the Process by doing a SELECT PROCESS or SELECT TASK command.

Note: For a Process in a VTAM I/O STATE and a SUBSTATE of SEND or RECEIVE, the FLUSH or SUSPEND command is implemented whether you specify FORCE or not. That is not true for the other states.

SUSPEND, FLUSH, or DELETE PROCESS Command Format

The SUSPEND PROCESS, FLUSH PROCESS, and DELETE PROCESS commands have the following format and associated parameters. Required parameters and keywords are in bold print.

Label	Command	Parameters
(optional)	FLUSH PROCess DELETE PROCess SUSPEND PROCess	WHERE (PNAME= name (list) PNUMBER= number (list) SUBmitter= (node name, user ID) (list) FORCE) Note: The FORCE parameter is not valid for DELETE PROCESS.
		CASE = Yes No

The following table describes the parameters:

Parameter	Description
WHERE (PNAME = name (list) PNUMber = number (list) SUBmitter = (nodeid, user ID) (list) FORCE)	<p>Specifies which Process to suspend, delete, or flush. Name multiple Processes in the command using the search criteria method described on “Setting Selection Criteria” on page 579 .</p> <p>WHERE is the only required parameter for the SUSPEND PROCESS, DELETE PROCESS, and FLUSH PROCESS commands. Its subparameters are optional. However you must specify at least one of the PNAME, PNUMBER, and SUBMITTER subparameters.</p> <p>PNAME = name (list) specifies the name of the Process or a list of Process names enclosed in parentheses and separated by commas.</p> <p>PNUMber = number (list) specifies the number of the Process selected or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999.</p> <p>SUBmitter = (nodeid, user ID) (list) specifies the nodeid and user ID of the user that submitted the Process. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.</p> <p>FORCE enables you to suspend or flush a Process waiting for unavailable resources. If the initial suspend or flush fails, retry the command with FORCE.</p> <p>Note: FORCE parameter is not valid for DELETE PROCESS. If FORCE is specified for a Process executing on an LU6.2 session, it can terminate the session immediately, and IBM Connect:Direct does not exchange the Process statistics between nodes.</p>
CASE = Yes No	Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not the Process itself. This parameter is optional.

Note: Processes with a QUEUE or STATUS of PR cannot be flushed or suspended with the FLUSH or SUSPEND PROCESS command.

SUSPEND, FLUSH, and DELETE Command Examples

Examples of the SUSPEND, DELETE, and FLUSH commands are shown in the following:

SUSPEND Process

The following command suspends any executing Process with a submitter node ID of DALLAS and a submitter user ID of SMITH:

```
SUSPEND PROC WHERE (SUB=(DALLAS, SMITH))
```

The following command suspends the Process named PAYROLL, which is Process number 514 and the Process named COPY100, which is Process number 575.

```
SUS PROC WHERE ( -
PNAME=(PAYROLL,COPY100), -
PNUM=(514,575))
```

FLUSH Process

The following command flushes executing Processes submitted by SMITH at the node DALLAS:

```
FLUSH PROC WHERE (SUB=(DALLAS, SMITH))
```

The following command flushes the Process named PAYROLL which has Process number of 514 and the Process named COPY100 which has the Process number of 575:

```
FLUSH PROC WHERE ( -  
    PNAME=(PAYROLL,COPY100), -  
    PNUM=(514,575))
```

DELETE Process

The following command deletes Processes with the number 60584:

```
DEL PROC WHERE (PNUM=60584)
```

The following command deletes all Processes submitted by CHUCK at the node DALLAS.MVS:

```
DEL PROC WHERE (SUB=(DALLAS.MVS, CHUCK))
```

The following command deletes all Processes in the PR queue. When specifying QUEUE= PR, no other selection criteria are allowed.

```
DELETE PROCESS WHERE (QUEUE=PR)
```

Issuing the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS Commands through the Batch Interface

About this task

To use the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS commands from the Batch Interface, perform the following steps:

Procedure

1. Place commands in the DGADBATC job stream as described in [“Sample job stream to run the batch interface”](#) on page 513.
2. Submit the job while IBM Connect:Direct is running.
3. Verify the results.

Issuing the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS Commands through the IUI

About this task

To issue commands through the IUI:

Procedure

1. Access the function that you want as described in the following table.

Function	Access Method
Suspend an executing Process	<p>Select option PS from the Primary Options Menu.</p> <p>Type SUS at the Primary Options Menu command prompt</p> <p>Type a P next to the Process Name on the Operator Table, the Operator Table - Executing Queue, or the Selected Process screens.</p>
Delete a nonexecuting Process	<p>Select option DP from the Primary Options Menu.</p> <p>Type a D next to the Process Name on the Operator Table or the Selected Process screens.</p>
Flush a Process	<p>Select option FP from the Primary Options Menu.</p> <ul style="list-style-type: none"> Type an F next to the Process Name on the Selected Process screen.

The following screen shows an example of the Suspend an Executing Process screen. The Delete A Nonexecuting Process screen and the Flush Process screen are similar except that the FORCE option is not on the Delete A Nonexecuting Process screen.

```

node.name          SUSPEND AN EXECUTING PROCESS          hh:mm
CMD ==>
  PROCESS NUMBERS:
    ==> _____ ==> _____ ==> _____ ==> _____
    ==> _____ ==> _____ ==> _____ ==> _____
  PROCESS NAMES:
    ==> _____ ==> _____ ==> _____ ==> _____
    ==> _____ ==> _____ ==> _____ ==> _____
  FORCE: ('Y'-YES, 'N'-NO) FORCE SUSPENDING A PROCESS ON AN LU 6.2 SESSION
    ==> - MAY TERMINATE THE SESSION IMMEDIATELY WITHOUT
    STATISTICS OF THE PROCESS BEING EXCHANGED
  USER IDS:          USER NODES:
    ==> _____ ==> _____
    ==> _____ ==> _____
    ==> _____ ==> _____
    ==> _____ ==> _____
DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> NO

```

IBM Connect:Direct builds the SUSPEND PROCESS command from this screen, and selects the corresponding Process names and numbers and submitters (user ID and node) from left to right to construct the list form of this command. This list fully qualifies your search criteria.

2. Type at least one of the parameters or a combination of parameters as described in the SUSPEND PROCESS command syntax section in [Suspending, Flushing, and Deleting Processes](#), or press the **PF1** key for Help. IBM Connect:Direct suspends the Processes which match your search criteria.

Note: Case sensitivity designation refers only to the command parameters on screen, not to the parameters of the Process.

3. To minimize the risk of deleting, suspending, or flushing a Process accidentally, you may be required to confirm a Process request before it is executed. If you are required to confirm delete, flush, and suspend commands, a panel similar to the following screen is displayed. Do one of the following:
 - To confirm the operation, type Y on the command line. IBM Connect:Direct performs the function and returns to the previous screen.
 - To return to the previous screen without performing the function, press **PF3**. To change the Process numbers, select this option before executing the command.

```
DMDELCN2                                DATE: YYYY/MM/DD
                                         TIME: HH:MM
                                         Connect:Direct Confirm Delete Command
CMD ==>
You have requested the following process(es) be deleted, are you sure this is
what you want to do?
Reply Yes, No or CANCEL or Press PF3 to Cancel the Request
Process Numbers ==> 123                ==>                ==>
                                         ==>                ==>
Process Names   ==>                  ==>                ==>
                                         ==>                ==>
USER IDS:      USER NODES:
==>
                                         ==>
==>
                                         ==>
==>
                                         ==>
==
                                         ==>
__ Do not display this Confirm Delete prompt again.
```

Note: If you see the option, Do not display this Confirm Delete prompt again, you can type an X next to this option to turn off the Confirm prompt so that it will not display again during the current session.

Examining Processes in the TCQ

Use SELECT PROCESS to look over Processes in the TCQ. You can specify the search criteria and the form in which the information is presented (file, printout, table, or screen display).

Note: Use the SELECT STATISTICS command to determine the outcome of a completed Process.

SELECT PROCESS Command Format

The SELECT PROCESS command has the following format and parameters. The required parameters and keywords are in bold print. Default values are underlined>.

Note: The syntax and parameters for the VIEW PROCESS command are identical.

Label	Command	Parameters
(optional)	SEl ect PRO cess or View PRO cess	WHERE (DEST= node (list) PNAME= name (list) PNUMber= number (list) QUEUE= All queue name SERVER= server name STATUS= Process status (list) SUBmitter= (node name, user ID) (list)) CASE= Yes No FILE P R int <u>TAble</u>

The following table describes the parameters.

Parameter	Description
<p>WHERE (DEST = node (list) PNAME = name (list) PNUMber = number (list) QUEUE = All Exec Hold PR Timer Wait STATUS = Process status (list) SERVER = server name (list) SUBmitter = (node name, user ID) (list))</p> <p>Note: PR (Process Retention) requires additional setup, see <i>IBM Connect:Direct for z/OS Administration Guide</i>.</p>	<p>Specifies which Processes to select. Name multiple Processes in the command using the search criteria method described on “Setting Selection Criteria” on page 579 . The subparameters PNAME, PNUMber, and SUBmitter are optional, but you must specify at least one.</p> <p>WHERE is the only required parameter for the SELECT PROCESS command. Not all its subparameters are required.</p> <p>DEST = node (list) specifies the destination node name of the Process you are selecting or a list of destinations enclosed in parentheses and separated by commas.</p> <p>PNAME = name (list) specifies the name of the Process or a list of Process names enclosed in parentheses and separating by commas.</p>

Parameter	Description
	<p>PNUM = number (list) specifies the number of the Process you are selecting or a list of Process numbers enclosed in parentheses and separated by commas. The range is 1-99999.</p> <p>QUEUE = All queue name specifies Process selection based on the TCQ. Values are: All specifies selection of a Process from all queues. This value is the default. queue name can be one of the following:</p> <p>E = Executing queue H = Hold queue T = Timer queue W = Wait queue</p> <p>SERVER = server name (list) selects Processes on the specified IBM Connect:Direct Servers. The server name is a 1-8 character name assigned to each server in a IBM Connect:Direct/Plex through the CDPLEX.SERVER initialization parameter.</p> <p>STATUS = Process status (list) specifies Process selection by status value or a list of status values in parentheses separated by commas. Values include:</p> <p>H = All Held Processes HC = Held for Call HE = Held for Error HI = Held Initially HO = Held by Operator HP = Held due to Process error HR = Held Retain HS = Held for Suspension PR = Retained after Execution R = All Restart Processes RA = Held for Restart due to Allocation error RH = Restart Held (may be due to the Intelligent Retry feature) W = All Waiting Processes (including Retry) WC = Wait for Connection WT = Waiting for Transport WX = Waiting for IBM Connect:Direct Server</p> <p>SUBmitter = (node name, user ID) (list) specifies the nodeid and user ID of the user that submitted the Process. Specify a list of SUBmitter IDs by enclosing the IDs in parentheses and separating them by commas.</p>
CASE = Yes No	<p>Specifies whether parameters associated with accounting data, user ID, password, and data set name are case sensitive. The designation refers only to the command, not the Process itself. See Indicating Case Sensitivity for a general overview of case sensitivity. This parameter is optional.</p>
FILE PPrint TABLE	<p>Specifies the form in which the information is presented. TABLE is the default. See Indicating Output Destination for detailed information about the output format produced by each of the these parameters. This parameter is optional.</p>

SELECT PROCESS Command Example

The following command searches for all Processes submitted by SMITH at the node CD.DALLAS.

```
SEL PROC WHERE (SUB=(CD.DALLAS, SMITH))
```

Issuing SELECT PROCESS through the Batch Interface

About this task

To use the SELECT PROCESS command from the Batch Interface, perform the following steps:

Procedure

1. Place commands in the DGADBATC job stream as described in [“Sample job stream to run the batch interface”](#) on page 513.
2. Submit the job while IBM Connect:Direct is running.
3. Verify your results.

Issuing SELECT PROCESS through the IUI

About this task

Use the Select Process screen to specify the Processes that you want to display and how you want them displayed. To issue the SELECT PROCESS command in the IBM Connect:Direct IUI, perform the following steps:

Procedure

1. Select option SP from the Primary Options Menu to display the Select Process screen.

node.name		SELECT PROCESS		hh:mm	
CMD ==> 0					
CMD: 0 ... OPERATOR TABLE S ... OPERATOR TABLE/EXEC QUEUE STATUS					
P ... PRINT REPORT D ... DISPLAY REPORT V ... VIEW PROCESS					
QUEUE: ==> _ (A-ALL,W-WAIT,E-EXECUTE,H-HOLD,T-TIMER,P-PR)					
PROCESS NUMBERS: ==> _____ ==> _____ ==> _____ ==> _____					
PROCESS NAMES: ==> _____ ==> _____ ==> _____ ==> _____					
SERVER NAMES: ==> _____ ==> _____ ==> _____ ==> _____					
STATUS: (HO,HR,HI,HE,HC,HP,HS,PR,RH,RA,WC,WX,WT,H,R,W)					
==> _ _ ==> _ _ ==> _ _ ==> _ _					
DESTINATION NODES:					
==> _____ ==> _____					
==> _____					
USER ID: NODE ID:					
==> _____					
==> _____					
==> _____					
DO YOU WANT VALUES FOR THIS REQUEST TO BE CASE SENSITIVE? ==> NO					

2. To select the Processes, specify the parameters to use as search criteria. These parameters are described in [SELECT PROCESS Command Format](#).

Note: The case sensitivity designation refers only to the command parameters on the screen, not to the parameters of the Process itself.

3. To select the destination for your output, type the letter representing one of the following output options on the command line:
 - O accesses the Operator Table and creates a one line summary of each selected PNODE Process. For more information, see [Viewing and Controlling a Process through the Operator Table](#).

- S builds an Operator Table consisting only of Processes that are currently executing. The table shows how much data is transmitted for a COPY step. For more information, see [Accessing Execution Queue Status on the Operator Table](#).
- V displays the content of the matching Processes within the TCQ. For more information, see [Viewing the Content of Processes Matching Your Search Criteria](#).
- P prints a report of the selected Processes.
- D displays a report, similar to the following figure.

```

BROWSE -- temporary file name          ----- LINE 00000000 COL 001 080
COMMAND ==>                          SCROLL ==> PAGE
***** TOP OF DATA *****
=====
                        SELECT PROCESS
=====

Prc Name   => PROCESSA           Queue    => EXEC
Prc Num    => 3                  Status   => EX
Prc Debug  => 00000200           SMFID   => SYSA
Subnode    => CD.NODE1           Time     =>
Other Node => CD.NODE2           Date     =>
Jobname    =>                   Job Num  =>
                                      Day      =>

Userid     => USER01
Prty       => 10      Retain => N  Ret proc => NONE
Class      => 1              State  =>
Step Name  =>              Xmit St =>
Max Class  => 100           Ses id  => PNODE
Message Id =>              Rtnode  => 0
Direction => Sending Side  Applid  => 05610
Ses Rstrt  => 0             Dyn Rstrt=> 0
Snding DSN => SMSTEST.DSN1
Rcvng DSN  => SMSTEST.DSN2
Volseq No  => 001           Ttrn    => 00012434
Volser     => ARTS09
Function   => COPY          Member   =>
V2 Buffer Size      => 2,097,152
Negotiated V2 Buffer Size => 2,097,152
TCP Buffer Size Used    => 2,097,152
Estimated File Size => 10,000,050,560
Sent: Blks => 74,671 Recs => 0 RU's => 233
Bytes Read  => 2,081,966,480
Bytes Sent  => 464,633,653
Compression Factor => 77.7

```

Viewing and Controlling a Process through the Operator Table

About this task

The Operator Table displays the PNODE Processes which satisfied your selection criteria; it does not display information about Processes submitted from another node.

Procedure

1. After you type **O** on the command line of the Select Process screen, the Operator Table/Executing Queue screen is displayed.

The first two characters of the QUEUE indicate the queue, the second two characters indicate the status value. The OTHER.NODE is the name of the nonsubmitting node in the session.

2. Type an option in the OPTION column next to the Process name (PNAME) and press **Enter**. The following table describes each option.

Option	Description
H	Place a nonexecuting Process on the Hold queue.
D	Delete a nonexecuting Process from the queue. For more information, see Issuing the SUSPEND PROCESS, DELETE PROCESS, or FLUSH PROCESS Commands through the IUI .

Option	Description
R	Release a held Process.
P	Suspend a Process from the executing queue. For more information, see Issuing CHANGE PROCESS through the IUI .
S	Select a Process for detailed display. For more information, see Viewing and Controlling a Process through the Selected Process Screen .
V	View a Process in the executing queue. For more information, see Viewing the Content of Processes Matching Your Search Criteria .

Note: When you release a Process which contains RETAIN=YES, it is copied and assigned a new Process number.

3. If you are required to confirm a delete or suspend Process request before it is executed, respond to the Confirm Command prompt.

Note: If message ID SOPA006I is displayed on the Operator Table screen next to the Process name under the OPTION column, it indicates that the Process can no longer be found. The Process was probably in execute status when another function was selected, and the screen was not refreshed. Screens are not automatically refreshed. Press **Enter** to refresh the screen.

4. To view any message text, type **END** to leave the Operator Table and return to the Select Process screen. Then type **M** on the command line, and press **Enter**.

Accessing Execution Queue Status on the Operator Table

Procedure

1. After you type **S** at the command line of the Select Process screen, the Operator Table/Executing Queue displays Processes that are currently executing. The Operator Table/ Executing Queue screen shows how much data is transmitted by the COPY step, including:

- Blocks for block-mode transmissions
- Records for record-mode transmissions
- RUs (request/response units)
- I/O bytes
- VTAM bytes
- The compression factor

The following figure is a sample of the screen and shows that two Processes are executing.

node.name				Row 1 to 2 of 2	
-----OPERATOR TABLE/EXECUTING QUEUE-----					
==> Q		SCROLL ==> PAGE			
OPTION	PNAME	PNUMBER	SUBMITTER.NODE--	OTHER.NODE-----	QUEUE
SERVER	USERID				

S3	BENCHRC	272	Q1A.ZOS.V4600	Q1A.ZOS.V4600	-P EX
	USER01				
	Blks => 6	Recs => 0	RUs => 173		
	I/O bytes => 177,138	Compression			
	VTAM bytes => 177,152	Factor => 0.0%			
	BENCHRC	272	Q1A.ZOS.V4600	Q1A.ZOS.V4600	EX EX
S3	USER01				
	Blks => 7	Recs => 0	RUs => 174		
	I/O bytes => 195,440	Compression			
	VTAM bytes => 178,176	Factor => 8.8%			
***** Bottom of data *****					

2. Type an option in the OPTION column next to the Process name (PNAME) and press **Enter**. The following table describes each option.

Option	Description
P	Suspend a Process in the executing queue.
S	Select a Process for detailed display. For more information, see Viewing and Controlling a Process through the Selected Process Screen .

Viewing and Controlling a Process through the Selected Process Screen

About this task

To view and control a Process through the Selected Process Screen:

Procedure

1. From the Operator Table, type **S** in the OPTION column next to the name of the Process to display the Selected Process screen. This screen displays details about the Process that you selected from the Operator Table.

The following figure is a sample of the screen which displays all available information about the particular Process you selected from the Operator Table.

CD.NODE2		SELECTED PROCESS		16:09	
CMD ==>					
Process Name	=> PROCESS1	Number	=> 5	Step	=>
Other Node	=> CD.NODE2	DEBUG=>	00000200	Status	=> EX
Commid	=> 05610;11.12.13.192			Queue	=> EXEC
Function	=> COPY	Sub State	=> TCP	State	=> ST RUNNG
Server	=>	PLEXCLASS	=> ()
Submitter	=> CD.NODE1	SENDING SIDE		SMFID	=> SYSA
Userid	=> USER01				
Scheduled Time	=>	Date	=>	Day	=>
Queueing Prty	=> 10	Class	=> 1	Retain	=> NO
Submitted Class	=> 1	Max Class	=> 100	Sess.ID	=> PNODE
Session restrt	=> 0	Dyn rstirt	=> 0	RouteID	=>
Last Msgid	=>	Last RC	=> 00000000	RetProc	=>
Secure	=> UN-AUTHENTICATED	Signature	=> N	Encryption	=> NONE
Sending File	=> SMSTEST.DSN1				
Receiving File	=> SMSTEST.DSN2				
Volume seq no.	=> 1	Volser	=> ARTS09	TTRN	=> 0000BCD0
zFBA	=> ()	Current zFBA	=>
Blks	=> 48,165	Recs	=> 0	RUS	=> 150
Bytes Read	=> 1,341,639,760			MEMBER	=>
Bytes Sent	=> 299,412,598			Compression Factor	=> 77.7%

2. Type any of the following Process control commands at the command line, and press **Enter**. The following table describes each option:

Command	Description
P	Suspend the currently executing Process
D	Delete a nonexecuting Process
H	Place a nonexecuting Process in the HOLD queue
R	Release a nonexecuting Process
Enter	Monitor the currently executing Process execution
C	Change Other Node, Queueing Priority, Scheduled Time, Scheduled Day, and Retain Status in a nonexecuting Process

3. To change parameters, move the cursor to the appropriate field on the screen, make the changes that you want, and press **Enter**.
4. If you are required to confirm a Process request, respond to the Confirm Delete/Suspend Command prompt.

5. If the Process completes execution while the Selected Process screen is displayed, press **Enter** to return to the Operator Table screen.

Viewing the Content of Processes Matching Your Search Criteria

About this task

To view the content of Processes matching your search criteria:

Procedure

- To view output on the VIEW PROCESS screen, do one of the following:
 - Type a V on the command line on the Select Process screen after specifying the parameters to use as search criteria. These parameters are described in [SELECT PROCESS Command Format](#). Also, for more information, see [Issuing SELECT PROCESS through the IUI](#).
 - Type V in the OPTION column next to the Process name on the Operator Table/Executing Queue screen. (The PNODE Processes which satisfied your selection criteria on the Select Process screen are displayed on the Operator Table/Executing Queue screen.) See [Viewing and Controlling a Process through the Selected Process Screen](#).

The following shows the sample output of the VIEW PROCESS command:

```
=====
                        VIEW PROCESS
          PROCESS NAME: TOEND    PROCESS NUMBER: 1
=====
TOEND    PROCESS  SNODE=CD.NODE          -
          PNODE=CD.NODE          -
          HOLD=YES               -
          RETAIN=NO              -
          CLASS=1                -
          PRTY=2                 -
STEP01    COPY FROM              -
          (SNODE                 -
          DSN='USER01.JCL.LIB'   -
          DISP=(SHR)             -
          )                      -
          TO                     -
          (PNODE                 -
          DSN='USER01.JCL.LIBX6' -
          )                      -
=====
```

Process Queuing and Recovery

IBM Connect:Direct stores submitted Processes in the Transmission Control Queue (TCQ). The TCQ controls Process execution. It consists of two inter-dependent VSAM Relative Record Data Sets (RRDS), and an in-memory queue which controls access. The two data sets are the TCQ and the TCX, which is a space map for the TCQ.

Logical Queues

The TCQ has the following logical queues:

- Wait
- Execution
- Hold
- Timer
- Process Retention

Access the queues and manipulate Processes using the following IBM Connect:Direct commands:

- CHANGE PROCESS
- DELETE PROCESS

- FLUSH PROCESS
- SELECT PROCESS
- SUSPEND PROCESS

Refer to [Controlling Processes with Commands](#), for more information the on each of these IBM Connect:Direct commands.

Queuing Parameters

The RETAIN, HOLD, and STARTT parameters queue Processes as described in following table.

Parameters	Queue	Comments
None	Wait	Process remains on the Wait queue until IBM Connect:Direct can start a session with the SNODE. The Process then moves to the Execution queue.
RETAIN=INITIAL	Hold	Process automatically executes each time IBM Connect:Direct is initialized with TCQ=WARM. This setting is useful for Processes that contact other IBM Connect:Direct nodes each time IBM Connect:Direct completes initialization. This action causes any work queued on the remote node for the local node to begin.
RETAIN=YES	Hold	A copy of the Process is kept in the Hold queue after it has executed. The Process does not execute again until it is released by a CHANGE PROCESS command. You can use RETAIN=YES, combined with STARTT, to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(,12:00) starts the Process each day at noon.
HOLD=YES	Hold	Process remains in the Hold queue until someone releases the Process.
HOLD=CALL	Hold	Process is automatically moved from the Hold queue to the Wait queue when the SNODE contacts the node on which the Process resides.
STARTT	Timer	When the scheduled time and date arrive, the Process is put on the wait queue and is available for execution. You can use RETAIN=YES, combined with STARTT, to run a Process at a periodic interval. For example, RETAIN=YES and STARTT=(Tuesday, 3pm) starts the Process every Tuesday at 3 pm; RETAIN=YES and STARTT=(,12:00) starts the Process each day at noon. You can also use the Timer queue for session retry and file allocation retry based on IBM Connect:Direct initialization parameters specified by a particular installation. When you exhaust retry limits, the Process is moved to the Hold queue with an HE status.

When you submit a Process, IBM Connect:Direct puts it in the appropriate logical queue based on Process statement routing parameters listed in the preceding table. See [Process Execution Example](#) for an illustration of how IBM Connect:Direct executes a Process.

TCQ Status and State Values

The SELECT PROCESS command displays IBM Connect:Direct status values, task state values, and transport state values. These are defined in the following table:

Value	Definition
IBM Connect:Direct Status Value	Each Process on the TCQ has an associated IBM Connect:Direct status value. This status value has a unique meaning determined by which queue the Process is in. The SELECT PROCESS command displays IBM Connect:Direct status values.
IBM Connect:Direct Task State Values	When a Process is in the Execution queue, the SELECT PROCESS command also displays a IBM Connect:Direct task state value. The state values are provided for information purposes. You cannot modify or control them using IBM Connect:Direct commands. The task state value shows the current state of the Process. Usually, IBM Connect:Direct tasks are waiting for completion of a service such as File I/O, IBM Connect:Direct locked resource, or VTAM I/O.
Transport State Values	<p>When a Process is on the Execution queue, the SELECT PROCESS command can also display a transport state value, depending on timing. The state values are provided for informational purposes. You cannot modify or control them using IBM Connect:Direct commands. For example, if the Process is currently in a VTAM I/O state, the SELECT PROCESS output can show a further state value such as send or receive.</p> <p>While a Process is executing, the SELECT PROCESS command displays the number of file blocks or records and VTAM request/response units (RUs) sent or received. This display gives you an indication of the status of COPY statements on the Process.</p>

Wait Queue

The following table shows the IBM Connect:Direct status value for the Wait Queue:

Status Value	Explanation
WC	Waiting for Connection. This status is the initial queue status when a Process is submitted without HOLD or RETAIN specified. This status means the Process is ready to execute as soon as possible. Process is ready to run but other Processes are executing with the same SNODE, and no other sessions are available. This Process runs as soon as an eligible session is available. If you find a Process in this state, it is most likely for this reason. Processes are also in this state when IBM Connect:Direct is quiesced.
WT	Waiting for transport. This status is the status of a Process when the transport protocol is not available. This Process runs as soon as the transport protocol is available.
WX	Waiting for server. This status is the status of a Process waiting for a IBM Connect:Direct Server to become available. This Process runs as soon as an eligible IBM Connect:Direct Server is available. An eligible IBM Connect:Direct Server is an active server that supports the Process PLEXCLASS and the transport protocol (SNA, TCP, UDT, or CTCA). The transport protocol must also be available on the server for it to be eligible.

The following table shows the applicable commands for the Wait Queue:

Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Execution Queue

The following table shows the IBM Connect:Direct status values for the Execution Queue:

Status Value	Explanation
EX	Node is in Process control and executing the displayed Process.
WC (Waiting for Connection)	Process control is in negotiation while the two nodes determine which Process executes next, based on priority.
PR.CNTL	This node is not in Process control. This status occurs: (1) during Process negotiation where highest priority on either node runs next, or (2) when the current node is the SNODE during Process execution.
SS	Session with other node is being started.

The following table shows IBM Connect:Direct task state values for Processes in the Execution Queue:

Task State	Explanation
DISPATCH	Task is waiting to dispatch
INACTIVE	Task can be dispatched but inactive
VTAM I/O	Task is waiting on VTAM request
P=SNODE	PNODE equals SNODE task
VSAM I/O	Waiting on VSAM I/O request
MISC	Miscellaneous I/O, such as a WTO
FILE I/O	Non-VSAM I/O
LOCK	Waiting for IBM Connect:Direct locked resource
SUBTASK	Waiting on a subtask, such as open or close, allocation, security, or RUN TASK
RUNNING	Executing instructions
TCA SCAN	TCA scan
TIMER	Waiting for timer event
ATTACH	Waiting for attached tape drive
ALLOCATE	Waiting for allocation to complete
MOUNT	Waiting for tape mount
OPEN	Waiting for OPEN to complete

The following table shows the Subtask state values for the Execution Queue. The first characters of the subtask request state indicate the session protocol such as TCP or LU6.2.

Subtask State	Explanation
ADOPT V2	Performing Adopt
CALL V2	Performing Call
INIT V2	Performing Initialization
CLEANUP	Performing Cleanup
HANGUP	Performing Hangup

Subtask State	Explanation
SEND V2	Performing Send
SEND RSP	Performing Send Response
SEND SIG	Performing Send Signal
RECV V2	Performing Receive
ANSWER V2	Performing Answer
IOCTL V2	Performing I/O Control
GET BUF	Performing I/O Control Get Buffer
RDY RCV	Performing I/O Control Ready Receive

The following table shows the transport state values for the Execution Queue:

Transport State	Explanation
NO SESSION	No VTAM session
SESSION EST	Session with another node being established
NO REQUEST	No VTAM request outstanding
RECEIVE	Waiting on VTAM RECEIVE request
OPEN	Waiting on VTAM OPEN request
CLOSE	Waiting on VTAM CLOSE request
SETLOGON	Waiting on VTAM SETLOGON request
REQSESS	Waiting on a request session request
OPNDST	Waiting on an open destination request
CLSDST	Waiting on a close destination request
GENCB EXTLST	Waiting on a GENCB EXLIST
GENCB ACB	Waiting on a GENCB ACB
GENCB NIB	Waiting on a GENCB NIB
GENCB RPL	Waiting on a GENCB RPL
REJSESS	Waiting on a reject session request
SESSIONC	Waiting on a session cancel request
INQUIRE	Waiting on an inquire request
OPNSEC	Waiting on an open secondary request
RSHUTD	Waiting on a request shutdown request
SIMLOGON	Waiting on a simulate logon request
SND RESPONSE	Waiting on a send response request
WAIT FOR +DR	Waiting on a definite response request
VTAM I/O	Waiting on VTAM request
SEND	Waiting on VTAM send request
DACTSESS	waiting on LU6.2 deactivate session request

Transport State	Explanation
RCVFMH5	Waiting on LU6.2 receive FMH-5 request
REJECT	Waiting on LU6.2 reject conversation request
ACTSESS	Waiting on LU6.2 activate session request
ALLOC ALLOCD	Waiting on LU6.2 allocate conversation until available request
PRERECV	Waiting on LU6.2 prepare to receive request Note: In a IBM Connect:Direct/Plex environment, the LU6.2 connection protocol does not enable the IBM Connect:Direct Manager to redirect work to one of its servers. The remote node must address the server on which you want to run an LU6.2 Process. To do this, specify the node name and VTAM address of the IBM Connect:Direct Server on which the Process is to run in the remote server's network map. Use the same CDPLEX.SERVER.NODE and CDPLEX.VTAM specified for the local initialization parameters for the IBM Connect:Direct Server you are trying to address.
CNOS	Waiting on LU6.2 change number of session request
DELLOC	Waiting on LU6.2 deallocate conversation request
SEND CONFRMD	Waiting on LU6.2 send confirmation request
6.2 RECEIVE	Waiting on LU6.2 receive request
6.2 SEND	Waiting on LU6.2 send request
SEND ERROR	Waiting on LU6.2 send error request
DISPLAY	Waiting on LU6.2 display session limit request
DEFINE	Waiting on LU6.2 define session limits request
ALLOC IMMED	Waiting on LU6.2 allocate conversation immediately request

The following table shows the allocate state values for the Execution Queue:

Allocate State Value	Explanation
SVC99	Performing SVC99
CAT SEARCH1	CAMLST locate
READ VTOC	CAMLST search

The following shows the applicable command descriptions for the Execution Queue:

Applicable Command	Description
Flush Process	Terminates and deletes an executing Process
Select Process	Displays Process status and state
Suspend Process	Terminates and moves an executing Process on the hold queue

Hold Queue

The following table shows the IBM Connect:Direct Status values for the Hold Queue:

Status Value	Explanation
HC (Held for Call)	<p>Process is submitted with HOLD=CALL specified.</p> <p>A session started from either node causes IBM Connect:Direct to place this Process on the wait queue in WC status, and eventually in the execution queue (EX) when the first Process finishes.</p> <p>Note: If the SNODE is defined with a null IP address, the Processes will be released serially. Refer to the HOLD parameter in SUBMIT Command for more information.</p>
HE (Held in Error)	The Process was submitted and received an error unrelated to allocation or session errors. The Process is being checkpointed and REQUEUE=YES is specified. A common error that causes IBM Connect:Direct to place the Process in HE status is an out of space condition (Sx37 ABEND).
HI (Held Initially)	Process is submitted with HOLD=YES specified.
HO (Held by Operator)	An exception response is sent from the other node during FMH exchanges at Process negotiation or step termination. It also occurs if an FMH is invalid or is sent out of sync, or if the remote node is not defined in the network map.
HP (Held due to Process Error)	An exception response is sent from the other node during EXIT exchanges at Process negotiation or step termination. HP also occurs if an EXIT is invalid, sent out of sync, or if the remote node is not defined in the Network map.
HR (Held Retain)	Process is submitted with RETAIN=YES specified or being held for Process retry (intelligent retry).
HS (Held for Suspension)	The operator issued a SUSPEND PROCess command. You can release the Process later. The Process will run (Held for Suspension) when IBM Connect:Direct is recycled.
RA (Held for Restart Due to Allocation Error)	During Process execution, an allocation error occurred that matched those specified in the initialization parameters. This status enables the Process to restart after the allocation problem is resolved.
RH (Restart Held)	Either the Process is being held awaiting a connection to an inactive node by the Intelligent Session Retry feature (see Intelligent Session Retry) or a checkpointed Process is executing when an error such as a lost session or an I/O error occurred. This status enables the copy to restart when the session is reestablished.
WC (Wait For Connection)	Session establishment is attempted, including retries if specified, and failed. The current Process is put on the wait queue (and later EX Q) if a session with that node is established later. It also can be released.

The following table shows the applicable commands for the Hold Queue:

Applicable Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Timer Queue

The following table shows the IBM Connect:Direct Status values for the Timer Queue:

Status Value	Explanation
RE (Retry)	The session with the SNODE is in the retry state. The number of and interval between retries is specified in the initialization parameters. The Process can be in retry status for session establishment or for an allocation error.
WC (Wait For Connection)	The Process is submitted with a start time or date (STARTT) that has not expired. When the STARTT is reached, the Process is put on the wait queue for scheduling to the EX Q.

The following table shows the applicable commands for the Timer Queue:

Applicable Command	Description
Change Process	Modifies Process attributes
Delete Process	Removes the Process from the queue
Select Process	Displays Process status and state

Process Recovery

IBM Connect:Direct provides facilities to recover from most errors that occur during Process execution. Recovery from the point of failure is usually accomplished quickly. The following types of errors can occur during normal operation:

- Link failure terminates a session between IBM Connect:Direct systems
- File I/O error occurs during Process execution
- IBM Connect:Direct abends because of a hardware or other error
- TCQ Corruption

IBM Connect:Direct provides the following facilities to address errors:

Facility	Description
Session establishment retry	<p>When one or more Processes run with a node, IBM Connect:Direct establishes a session with that node and begins execution. If IBM Connect:Direct cannot start the session, IBM Connect:Direct retries the session establishment. The initialization parameters, MAXRETRIES and WTRETRIES, determine the number of retries and the interval between retries.</p> <p>If IBM Connect:Direct cannot establish a session after all retries are exhausted, the Process is placed in the Hold queue in the TCQ with a status of Waiting for Connection (WC). When a session is established with the other node, all other Processes are scanned and the highest priority Process is executed after the previous Process is finished.</p>
VTAM automatic session retry	<p>If Process execution is interrupted because of a VTAM session failure, IBM Connect:Direct automatically attempts to restart the session. This recovery facility uses the same parameter values as the session establishment retry facility.</p> <p>If IBM Connect:Direct cannot establish the session, the Process that is executing and any other Processes that are ready to run with the other node are placed in the Hold queue with a status of Waiting for Connection (WC).</p>

Facility	Description
TCQ/TCX Repair Utility	When the TCQ becomes corrupt because of an outage or other circumstance, IBM Connect:Direct may abend in production or during the next DTF initialization. The IBM Connect:Direct administrator can use the TCQ/TCX repair utility to remove ambiguous or corrupt data and avoid having to cold start the DTF and reinitialize the TCQ, thus losing any Processes left in the TCQ.
Process step checkpoint	As a Process executes, IBM Connect:Direct records which step is executing in the TCQ. If Process execution is interrupted for any reason, the Process is held in the TCQ. When the Process is available for execution again, IBM Connect:Direct automatically begins execution at that step.
COPY statement checkpoint/restart	<p>For physical sequential files and partitioned data sets, IBM Connect:Direct collects positioning checkpoint information at specified intervals as a COPY statement executes. Checkpoints are taken for each member that is transferred within a PDS, regardless of the checkpoint interval. If the copying procedure is interrupted for any reason, you can restart it at the last checkpoint position.</p> <p>Note: Whenever a Process step is interrupted and restarted, some data will be retransmitted. Statistics records for the Process step will reflect the actual bytes transferred, and not the size of the file.</p> <p>The COPY statement checkpoint/restart works in conjunction with step restart. The restart is automatic if IBM Connect:Direct can reestablish a session based on the initialization parameter values for MAXRETRIES and WTRETRIES. See COPY Statement Checkpoint/Restart Facility for more information.</p> <p>The CHANGE PROCESS command can also invoke the checkpoint/restart facility. See Controlling Processes with Commands for instructions on how to use the CHANGE PROCESS command.</p> <p>Note: Checkpoint/restart is not supported for I/O exits at this time.</p>

COPY Statement Checkpoint/Restart Facility

The checkpoint/restart facility includes the following elements:

- Initialization parameters
- Checkpoint/restart file
- Copy statement checkpoint parameters

Initialization Parameters

IBM Connect:Direct uses the following initialization parameters with the COPY statement checkpoint/restart facility:

Parameter	Description
CKPT.MODE	<p>Specifies whether the checkpoint function is performed when IBM Connect:Direct is transferring a file in record mode or block mode. (Record mode transfer is used when reblocking of the output file is specified.) It also specifies whether the checkpoint function is performed for partitioned data sets and if so, what type. Automatic checkpointing of VSAM files is supported.</p> <p>This parameter does not apply to TCP/IP and LU6.2 connections. With TCP/IP and LU6.2 connections, checkpointing is based on the checkpoint interval requested (regardless of the data set type or whether it is in block or record format) and on both nodes' ability to perform the checkpoint function. When both nodes agree to do checkpointing, the checkpoint interval is controlled by the sending node while checkpoint record filing is done on the receiving node.</p>
CKPT	<p>Specifies the default interval for checkpointing when it is not specified on the COPY statement.</p> <p>You should specify the CKPT value as a multiple of the value specified for the V2.BUFSIZE initialization parameter. If you do not, performance can seriously deteriorate.</p> <p>Note: Be aware of additional overhead associated with specifying too small a checkpoint interval, particularly when transferring large files.</p> <p>Note: For sequential files, try to avoid using a CKPT value less than:</p> $\text{BLKSIZE} * \text{NCP} * 10$ <p>where NCP is the number of buffers for reading data from or writing data to a sequential data set using BSAM</p> <p>For more information, search on Improving Performance in the IBM Connect:Direct for z/OS Administration Guide.</p>
CKPT.DAYS	Specifies the amount of time that checkpoint records are kept if they are not deleted. IBM Connect:Direct automatically deletes checkpoint records when a Process is restarted and runs to a successful completion.
CKPTDSN	Specifies the name of the checkpoint/restart file that holds checkpoint records during execution of the COPY statement.

For detailed information about these parameters, search on Global Initialization Parameters in the IBM Connect:Direct for z/OS Administration Guide.

Checkpoint/Restart File

The IBM Connect:Direct checkpoint/restart file contains positioning information for both files involved in executing a COPY statement. IBM Connect:Direct maintains the checkpoint records throughout data transmission and deletes them when a transmission completes successfully.

Note: Checkpoints take place on the receiving end of a transfer. During restart, this information is exchanged with the sender so that appropriate positioning can take place.

A checkpoint record can be left in the Checkpoint file if an interrupted Process is deleted by the operator. IBM Connect:Direct scans the checkpoint records during initialization and deletes records older than the value specified in the CKPT.DAYS initialization parameter.

COPY Statement Parameters

The CKPT and REQUEUE parameters of the COPY statement also control aspects of the Checkpoint/Restart facility. The following table describes these parameters.

COPY statement parameters

Parameter	Description
CKPT	<p>Specifies the interval IBM Connect:Direct uses to record checkpoint information. IBM Connect:Direct uses this CKPT value, rounded to the nearest block, in determining how many bytes to transfer before taking a checkpoint.</p> <p>If you do not specify the Copy statement CKPT parameter, IBM Connect:Direct uses the value specified in the CKPT initialization parameter. Specifying a value of CKPT=OK, or not specifying CKPT in the initialization parameters, disables checkpointing. Be aware of additional overhead associated with specifying too small a checkpoint interval, particularly when transferring large files.</p>
REQUEUE	<p>Indicates whether IBM Connect:Direct requeues Processes that end due to an ABEND, such as an Sx37, or enables any subsequent steps to run to Process termination. This parameter is only effective if checkpointing is in use. See the Connect:Direct Process Language help for more information.</p>

Checkpoint/Restart Examples for TCP/IP, UDT, or LU6.2 Transfers

The following examples describe how checkpoint/restart works for different transfer types.

TCP/IP, UDT, or LU6.2 Transfers

In the case of transfers using TCP/IP or UDT, IBM Connect:Direct sends approximately 30 bytes of overhead after sending an amount of data equal to the checkpoint interval in effect. Therefore, if the file size is 3 million bytes and the checkpoint interval is 10k, then 30 bytes of overhead is sent 300 times, resulting in a total overhead of 9000 bytes.

Other Transfers

For other transfers (LU0 or transfers to MS-DOS), IBM Connect:Direct sends an additional seven bytes per block or ten bytes per record of overhead during data transfer. Data is sent in record mode when reblocking is taking place during transfer (source and destination block sizes differ). When no reblocking is taking place, IBM Connect:Direct sends in block mode.

When you enable checkpointing, IBM Connect:Direct transfers positioning information in addition to the file. The positioning information enables IBM Connect:Direct to reposition the file in the event of an interruption. The following examples show how to determine this overhead in number of bytes when using Checkpoint/Restart for non-TCP/IP transfers.

In the following table, IBM Connect:Direct checkpoints the file in block mode and adds seven bytes to each block transmitted. This functionality adds only 2,100 bytes to the transmission of almost 6 million bytes.

File Attributes	Sending File	Receiving File
BLKSIZE	19,069	19,069
LRECL	0	0
DSORG	PS	PS
RECFM	U	U
FILE SIZE	300 blocks	300 blocks

In the following table, IBM Connect:Direct checkpoints the file in record mode because the data is being reblocked. IBM Connect:Direct adds 10 bytes to each record being transmitted. This functionality adds 150,000 bytes to the transmission, or 5 percent overall.

File Attributes	Sending File	Receiving File
BLKSIZE	5,000	2,000
LRECL	200	200
DSORG	PS	PS
RECFM	FB	FB
FILE SIZE	600 blocks	1,500 blocks

RUN TASK Checkpoint/Restart

Connect:Direct for z/OS also offers a checkpoint/restart feature with the RUN TASK Process statement.

If a RUN TASK program is executing on the SNODE and a session failure occurs, the PNODE recognizes the session failure and puts the Process in the Timer queue for retry. The SNODE, however, is still running the RUN TASK program and is not aware of the session failure until the program finishes. The checkpoint/restart feature for RUN TASK ensures that when the Process restarts on the PNODE, the RUN TASK program does not execute a second time on the SNODE.

RUN TASK Checkpoint Records

IBM Connect:Direct always writes a checkpoint record on the node where the RUN TASK program executes. The initial checkpoint record is written upon entry to DGADRNT\$, the module that handles the RUN TASK Process statement. IBM Connect:Direct updates the checkpoint record before attaching the program that is to execute. When the RUN TASK program finishes, IBM Connect:Direct updates the checkpoint record again.

RUN TASK Restart

If a RUN TASK Process step restarts, the node where the program executes attempts to find the checkpoint record in the checkpoint file. If the RUN TASK step is still executing, the Process that is running for the restart of the step waits for the RUN TASK program to finish the first task and then proceed to the next step of the Process.

At Process restart for a RUN TASK step, if the program is still executing, you see two Processes in the EX queue for the same Process step. The first Process is executing the program. The second Process is waiting for the first Process to complete. When the first Process completes, it determines that the session it was running under is lost and posts the second Process. The second Process records how the RUN TASK step that is still executing ended, and proceeds to the next step in the Process.

Re-execution of the RUN TASK

The RUN TASK step does not execute again if it is determined at restart that the RUN TASK step ended because it finished before the PNODE restarted the Process. However, if the RUN TASK program did not complete and is not currently running, then the RESTART parameter determines the restart of the Process.

Also, if at restart, IBM Connect:Direct cannot find the checkpoint record and the RUN TASK program is not executing, IBM Connect:Direct is unable to determine what action to take for the restart. If IBM Connect:Direct cannot determine what action to take for the restart, it uses the RESTART parameter. You can code the RESTART parameter on the RUN TASK step or in the initialization parameters.

Note: If you code the RESTART parameter on the RUN TASK step, it overrides the initialization parameter. When you code RESTART=YES, IBM Connect:Direct executes the program again. When you code RESTART=NO, the Process skips the RUN TASK step.

Search on Global Initialization Parameters in the IBM Connect:Direct for z/OS Administration Guide for detailed information about the RUNTASK.RESTART initialization parameter.

Process Results and Statistics

The SELECT STATISTICS command is used to retrieve statistics records. Use this command to see the results of a Process that has finished executing.

Note: The IBM Connect:Direct Activity Reporting System (ARS) produces detailed statistics reports.

You can also use the Connect:Direct Browser User Interface to perform some Connect:Direct for z/OS procedures.

Statistics Log Records

IBM Connect:Direct keeps a record of operations and activity in a Statistics Log file. This file contains the following record types:

Record Type	Provides Statistics About
Process Submit	The submitted Process
Member Copy	Transmission of a PDS member
Copy	How the COPY statement completed
Process Duration	How the Process completed
Run Job	A job submitted to the z/OS internal reader (JES)
Run Task	A program attached to the Process as a subtask
Submit within Process	A Process that is submitted within another Process
InterConnect Option Results	Processes created by the InterConnect Option

For a table of statistics record types, refer to Using IBM Connect:Direct Exits

Statistics Records Content

The statistics records include the following information:

- In all records:
 - Record type or function
 - Date and Time of record logging
 - Process name and number
 - SMFID
 - User ID
 - Other node (secondary or primary) involved in the Process
 - Start and stop time of the function
 - Highest or final completion code and message
 - Step label name, if one is specified
- In the Member Copy record, the input and output member names
- In the Process Submit and Submit Within records, the submitted PDS DSN and member name
- In the Copy record:
 - The number of records or blocks read from or written to the file, and whether the transfer was performed in block or record mode
 - The number of I/O bytes that were read or written from disk or tape
 - The number of VTAM bytes sent or received during the session, including IBM Connect:Direct and VTAM control information

- The RUSIZE

The size displayed is not the actual RUSIZE used by the transfer but rather the RUSIZE indicated by the LOGMODE. VTAM can negotiate for a smaller RUSIZE.

- The compression percentage

This percentage is calculated by evaluating the number of VTAM bytes and the number of I/O bytes. If the number of VTAM bytes is less than or equal to the number of I/O bytes, a positive compression percentage is calculated as:

$$\text{Compression Percentage} = (1 - \{\text{VTAM Bytes} \div \text{I/O Bytes}\}) \times 100$$

If the number of VTAM bytes is greater than the number of I/O bytes, a negative compression is calculated as:

$$\text{Compression Percentage} = - (1 - \{\text{I/O Bytes} \div \text{VTAM Bytes}\}) \times 100$$

The compression percentage values displayed in the Copy Termination record differ depending on the file attributes specified. A negative compression percentage means that more bytes are sent than are read from the file. This includes IBM Connect:Direct control information.

Elapsed Time Significance

Use the following table to interpret the elapsed time between the start and stop times for the different record types:

Record Type	Interpretation
For Copy Termination	The elapsed time includes file allocation on both nodes and the time required to mount a tape, when required.
For Run Task or Run Job	The elapsed time indicates the time it took to run the task or job before control returned to IBM Connect:Direct.
For Submit within Process	The elapsed time indicates the time it took to submit the Process to the TCQ.
For an ESF Process Submit	The start time is when the submit was performed (the Process was added to the TCQ). The stop time is when the Process was first added to the Process queue.

InterConnect Option Records

The InterConnect Option (ICO) routes files from a IBM Connect:Direct node to an IBM Connect:Enterprise® Secure Client node for distribution, distributes batches to a Connect:Direct node, and provides notification of success or failure of the Process. The InterConnect software copies the entire content of the SYSPRINT file of the IBM Connect:Enterprise Secure Client batch utility, STOUTL, into the IBM Connect:Direct Statistics file.

This report from STOUTL is displayed as a block by IBM Connect:Direct. The InterConnect Option software imposes a limit of 500 lines for the report. These reports are retrievable only through the SELECT STATISTICS command FILE option for batch, and the DISPLAY option for the IUI. Search on Offline Utilities in the IBM Connect:Enterprise for z/OS User's Guide for samples of the output reports from the ADD and EXTRACT functions.

Additional statistics records related to the InterConnect Option are written to the statistics facility. For a list of those statistics records, search on Using IBM Connect:Direct Exits in the IBM Connect:Direct for z/OS Administration Guide.

Note: If you specify a statistics exit in the initialization parameters, the LF and RO records are passed to the exit.

SELECT STATISTICS Command

Use the SELECT STATISTICS command to retrieve and examine statistic log records. You can select records based on certain conditions and indicate if you want the output displayed, printed, or saved in a file for later processing. See information about the FILE subparameter in [Indicating Output Destination](#) for how to save your output in a file for processing later.

SELECT STATISTICS Command Format

SELECT STATISTICS has the following format and parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameters	subparameters
(optional)	SElect STATistics	WHERE (subparameters)	ARCHDSN = dsname (list) CASE = YES NO CCODE = (condition, completion code) EXCLUDE = (MEMber MCR) (WTO) (NOTWTO) (list) FNAME = dsname (list) * MSGID = ID (list) PNAME = name (list) PNUMBER = number (list) SERVER = server name (list) SNODE = name (list) STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) TYPE = ID (list) * USER = name (list)
		FILE PRint TABLE SUMmary	Note: These parameters are mutually exclusive - only one of them must be specified.
(optional)	SElect STATistics	WHERE2 (subparameters)	ANODE = name (list) ARCHDSN = dsname (list) CASE=YES NO CCODE = (condition , completion code) EXCLUDE = (MEMBer) (MCR) (WTO) (NOTWTO) (list) FNAME = dsname (list) FNODE = name (list) LNODE = name (list) MSGID = ID (list) PNAME = name (list) PNODE = name (list) PNUMBER = number (list) RNODE = name (list) SERVER = server name (list) SNODE = name (list) STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) TNODE = name (list) TYPE = ID (list) UNODE = name (list) USER = name (list)
		FILE PRint TABLE SUMmary	Note: These parameters are mutually exclusive - only one of them must be specified.

The following table describes the SELECT STATISTICS command parameters:

Parameter	Description
WHERE	Specifies which statistics records to examine. This parameter or WHERE2 is required. The subparameters, such as PNAME, PNUMBER, and STARTT, are optional, but you must include at least one subparameter. WHERE name subparameters only support the generic character asterisk (*) and it is limited to the end of the name.
WHERE2	Specifies which statistics records to examine. This parameter or WHERE is required. WHERE2 supports the same subparameters as WHERE, in addition to the following sub-parameters: ANODE, FNODE, LNODE, PNODE, RNODE, TNODE and UNODE. When these subparameters are specified, they can use extended generics support. Also, when specified as a WHERE2 subparameter, SNODE can use extended generics support. Ex-tended generics support allows one or more asterisks (*) and/or percent signs (%) anywhere in the name. An asterisk stands for any number of characters. A percent sign stands for one character. The default record types selected by WHERE2 are the same as those for WHERE.
FILE PRint TABLE SUMmary	<p>Specifies the output format. Indicate only one. TABLE is the default. This parameter is optional.</p> <p>See Indicating Output Destination for an explanation of the output produced by the selection of the FILE, PRINT, and TABLE parameters. The SUMMARY parameter produces a 3-line summary per statistics record, similar to the example on Statistics Log Records.</p> <p>The reports from Connect:Enterprise are retrievable only through the FILE option for batch and the DISPLAY option for the IUI.</p>

The following table describes the SELECT STATISTICS command sub-parameters:

Parameter	Description
ANODE=name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name is any of the following: PNODE, SNODE or UNODE. You can specify a list of names by enclosing them in parentheses. ANODE names can contain lowercase characters and supports extended generics.</p> <p>ANODE supports extended generics. For example, if you specify ANODE=CD.TEST%.NODE, then records with CD.TEST1.NODE in the PNODE, SNODE or UNODE field are selected. To select CD.TEST12.NODE, specify ANODE=CD.TEST%%.NODE or ANODE=CD.TEST*.NODE or CD.TEST* or any of other extended generic specification.</p> <p>Note: ANODE is only allowed as a WHERE2 subparameter.</p>
ARCHDSN=dsname (list)	Specifies that IBM Connect:Direct is to search the archived statistics files named, instead of the statistics log currently in use by the DTF. Archived statistics files store old statistics data that is no longer active in the statistics log. See your system administrator for information about the availability of archived statistics at your site, and for the data set names you can specify with the ARCHDSN subparameter.

Parameter	Description
CASE=YES NO	<p>Specifies whether lowercase or mixed-case data is permitted for the USER, ANODE, FNODE, LNODE, PNODE, RNODE, SNODE, TNODE, UNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the purpose of the SELECT STATISTICS command.</p> <p>YES folds the data in USER, ANODE, FNODE, LNODE, PNODE, RNODE, SNODE, TNODE, UNODE, and FNAME to uppercase regardless of the actual data specified.</p> <p>NO preserves the actual case typed for the USER, ANODE, FNODE, LNODE, PNODE, RNODE, SNODE, TNODE, UNODE, and FNAME subparameters. The CASE defaults to the setting defined within the session defaults if nothing is specified.</p>
CCODE = (condition, completion code)	<p>Specifies selection by completion code.</p> <p>condition specifies a relationship to the completion code given in the subsequent positional parameter.</p> <p>The options for specifying condition include:</p> <p>GT for greater than LT for less than EQ for equal to NE for not equal to GE for greater than or equal to LE for less than or equal to</p> <p>completion specifies a decimal value 1 to 2,147,483,647 to allow for all completion codes that the RUN TASK statement can pass. This last value represents a maximum 31-bit binary number.</p> <p>For example, if CCODE = (EQ,12) is specified, records that have a Comp Code of 0000000C are selected.</p> <p>For another example, if CCODE = (GT,0) is specified, you see statistics records in which the step completion code is greater than zero, as long as the records also meet other specified criteria.</p>
EXCLUDE = (MEMBer MCR) (WTO) (NOTWTO) (list)	<p>Specifies to exclude certain statistics from selection. Specify a list of excluded options by enclosing them in parentheses and separating them by a space or a comma. To select all statistics, omit the EXCLUDE parameter from the SELECT STATISTICS command. The TYPE parameter overrides the EXCLUDE specifications.</p> <p>MEMBer MCR specifies whether or not to exclude the PDS member copy record for PDS copies on the statistics report.</p> <p>WTO specifies that Connect:Direct write-to-operator (WTO) messages are excluded from the SELECT STATISTICS command. Dynamic allocation messages are represented as WTOs in the statistics file.</p> <p>NOTWTO specifies that only WTO records are displayed that is, exclude everything that is NOT a WTO record.</p>

Parameter	Description
FNAME=dsname (list)	<p>Limits the selected statistics records to those that contain the specified file name. The FNAME subparameter is meaningful for the following record types:</p> <p>CT Copy Termination RJ Run Job SD Start IBM Connect:Direct SW Submit within Process</p> <p>The meaning of the file name within these records is unique for each record type. For example, the Run Job record contains the file name of the submitted JCL. Filenames can be up to 254 characters in length and can contain lowercase characters.</p> <p>FNAME supports use of limited generic specification using an asterisk (*) at the end of the dsname. For example, if you specify TEST.F1*, then all dsnames with first seven characters as TEST.F1 are selected.</p>
FNODE =name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name was the FROM (sending) node (CI, CT, FA, MC records), or was the execution node (RJ, RT records). You can specify a list of names by enclosing them in parentheses. FNODE names can contain lowercase characters and supports extended generics.</p> <p>Note: FNODE is only allowed as a WHERE2 subparameter.</p>
LNODE=name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name was the LOCAL (or this) node. You can specify a list of names by enclosing them in parentheses. LNODE names can contain lowercase characters and supports extended generics.</p> <p>Note: LNODE is only allowed as a WHERE2 subparameter.</p>
MSGID = ID (list)	<p>Specifies selection by message ID. You can specify a list of message IDs by enclosing them in parentheses. You can use a generic specification in the MSGID parameter using an asterisk (*) at the end of the ID. For example, if you specify SCPA*, then MSGIDs with SCPA in the first four characters of the message ID are selected.</p>
PNAME=name (list)	<p>Specifies selection by Process name. You can specify a list of Processes by enclosing them in parentheses. You can use a generic specification in the PNAME parameter by using an asterisk (*) at the end of the name. For example, if you specify PNAME=TEST*, then records with TEST in the first four characters of the Process name field are selected regardless of the contents of the remaining characters. Records having TEST, TEST123, and TESTX all satisfy this selection criterion.</p>
PNODE=name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name was PNODE. You can specify a list of names by enclosing them in parentheses. PNODE names can contain lowercase characters and supports extended generics.</p> <p>Note: PNODE is only allowed as a WHERE2 subparameter</p>
PNUMber = number (list)	<p>Specifies selection by Process number. You can specify a list of Processes by enclosing them in parentheses. The range is 1–199999.</p>

Parameter	Description
RNODE=name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name was a REMOTE (or the other) node. You can specify a list of names by enclosing them in parentheses. RNODE names can contain lowercase characters and support extended generics.</p> <p>Note: RNODE is only allowed as a WHERE2 subparameter.</p>
SERVER = server name	<p>Specifies the name of the IBM Connect:Direct member where the statistics are generated. The server name is a 1–8 character name assigned to each IBM Connect:Direct Server through the CDPLEX.SERVER initialization parameter. This parameter is required in a IBM Connect:Direct/Plex.</p>
SNODE = name (list)	<p>Limits the selected statistics records to those that are written for processes where the specified node name was the SNODE. You can specify a list of names by enclosing them in parentheses. SNODE names can contain lowercase characters and supports extended generics when specified as a WHERE2 subparameter.</p> <p>SNODE supports limited generics when specified as a WHERE subparameter. Use limited generic specifications by placing an asterisk (*) at the end of the name. For example, if you specify SNODE=DALLAS*, then records with DALLAS in the first six characters of the SNODE field are selected regardless of the contents of the remaining characters. Records having DALLAS.PROD, DALLAS.TEST, and DALLAS all satisfy this selection criterion.</p>

Parameter	Description
STARTT = ([date day] [,hh:mm:ssXM])	<p>Specifies selection by designated starting date and time. The date or day and time are positional parameters. If you do not specify the date or day, precede the time with a comma.</p> <p>date specifies the date to execute the Process. You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). You can use periods or backslashes (/) to separate the components of a date value. You can omit the separators only for transfers between mainframe nodes. Use separators to guarantee transfers between all platforms. You can use the following date formats, according to which date order is specified in the DATEFORM initialization parameter:</p> <ul style="list-style-type: none"> DATEFORM=MDY specifies the date format as mm/dd/yy, mm/dd/yyyy, mm.dd.yy, or mm.dd.yyyy DATEFORM=DMY specifies the date format as dd/mm/yy, dd/mm/yyyy, dd.mm.yy, or dd.mm.yyyy DATEFORM=YMD specifies the date format as yy/mm/dd, yyyy/mm/dd, yy.mm.dd, or yyyy.mm.dd DATEFORM=YDM specifies the date format as yy/dd/mm, yyyy/dd/mm, yy.dd.mm, or yyyy.dd.mm <p>Valid Julian dates formats are yyddd, yyyyddd, yy/ddd, yyyy/ddd, yy.ddd, or yyyy.ddd.</p> <p>If you do not specify the DATEFORM parameter, Connect:Direct for z/OS defaults to MDY date format.</p> <p>day specifies the day of the week to select. Valid names include Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. You can also specify YESTER to search for statistics records created yesterday or TODAY to start the search at records created today.</p> <p>hh:mm:ssXM indicates the time of day in hours (hh), minutes (mm), and seconds (ss) to select. Set XM to AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, 1:00 hours are expressed as 1:00AM, and 13:00 hours are expressed as 1PM. If you use neither AM or PM, IBM Connect:Direct assumes the 24-hour clock. You do not have to specify minutes and seconds.</p> <p>You can also specify NOON, which searches for the statistics records at noon, or MIDNIGHT, which searches for the statistics records at midnight. The time default is 00:00:00, the beginning of the day.</p> <p>If you specify time of day but not date, the output shows the first available entry in the statistics log for that time of day. Statistics from all later times and dates are output up to and including the stop time.</p> <p>If you do not specify the STARTT parameter, the search criterion consists of any time before the STOPT parameter.</p>
STOPT = ([date day] [,hh:mm:ssXM])	<p>Specifies a search for statistics records up to and including the designated date, day, and time positional parameters. If you do not specify the date or day, place a comma before the time.</p> <p>The date and time formats are the same as the STARTT parameter. The time default is 24:00:00, the end of the day.</p> <p>If you specify time of day but not date, all statistics through the most recent entry for that time of day are output.</p> <p>If you do not specify the STOPT parameter, IBM Connect:Direct selects statistics through the present time.</p>

Parameter	Description
TNODE =name (list)	Limits the selected statistics records to those that are written for processes where the specified node name was the TO (receiving) node (CI, CT, FA, MC records), or was the non-execution node (RJ, RT records). You can specify a list of names by enclosing them in parentheses. TNODE names can contain lowercase characters and supports extended generics. Note: TNODE is only allowed as a WHERE2 subparameter.
TYPE = ID (list) *	Specifies the statistics record types to select. Every statistics record IBM Connect:Direct generates has an associated record-type identifier. Each identifier is two characters long and indicates the event or function that generated the record. The identifier also indicates the record format and contents. Use TYPE= * to view all record types. The TYPE subparameter specifies which record types to select, and overrides the EXCLUDE parameter. When you do not specify a TYPE, the record types selected are determined by the output option chosen, and can be affected by the EXCLUDE subparameter. When you specify the PRINT or TABLE output option, the types in the following table are selected. Specify FILE for the output option to select all available types. Use the EXCLUDE subparameter to exclude certain types that the output option included. For a list of statistics record type identifiers, search on Statistics Records in the IBM Connect:Direct for z/OS Administration Guide.
UNODE = name (list)	Limits the selected statistics records to those that are written for processes where the specified node name was the UNODE. You can specify a list of names by enclosing them in parentheses. UNODE names can contain lowercase characters and supports extended generics. Note: UNODE is only allowed as a WHERE2 subparameter. Note: For Process oriented records, UNODE is the Submitter's node and for command oriented records, UNODE is the node where the command was issued.
USER = name (list)	Limits the selected statistics records to those that are written for users with the specified name. You can specify a list of names by enclosing them in parentheses. Use generic specifications by placing an asterisk (*) at the end of the name. For example, if you specify USER = SYS\$, then records with SYS\$ in the first four characters of the USER field are selected regardless of the contents of the remaining characters. Records having SYS\$BOB, SYS\$ADM, and SYS\$0001 all satisfy this selection criterion. USER names can be up to 64 characters in length and can contain lowercase characters.

Using SELECT STATISTICS through the Batch Interface

About this task

To use the SELECT STATISTICS command from the Batch Interface, perform the following steps:

Procedure

1. Place commands in the DGADBATC job stream as described in [“Sample job stream to run the batch interface” on page 513](#).
2. Submit the job while IBM Connect:Direct is running.

SELECT STATISTICS Command Examples

The following command searches for statistics records based on start and stop date and time, and excludes write-to-operator statistics messages.

```
SEL STAT WHERE ( -
  EXCLUDE=(WTO) -
  STARTT=(11/24/2003,06:45:00) -
  STOPT=(11/25/2003,8AM) -
)
```

The following command searches for statistics records based on start and stop date and time, and excludes write-to-operator and PDS member copy statistics records.

```
SEL STAT WHERE ( -
  EXCLUDE=(WTO, MEMB) -
  STARTT=(11/21/2003,15:25:00) -
  STOPT=(11/25/2003,15:30:00) -
)
```

The following command searches for statistics records based on Process name, and start and stop date and time. It excludes write-to-operator statistics messages.

```
SEL STAT WHERE ( -
  EXCLUDE=(WTO) -
  PNAME=BILLING -
  STARTT=(12/28/2003,06:00) -
  STOPT=(12/28/2003,23:00) -
)
```

The following command searches for signon records for the user Mary.

```
SEL STAT WHERE (USER=MARY, TYPE=SI) TABLE
```

The following command searches are identical and both select only WTO records.

```
SEL STAT WHERE (TYPE=W0) TABLE
SEL STAT WHERE (EXCLUDE=(NOTWTO)) TABLE
```

The following command searches for statistics records based on Process start/stop date and time that ran on a IBM Connect:Direct Server named SERVER1.

```
SEL STAT WHERE ( -
  SERVER=SERVER1 -
  STARTT=(12/28/2003,06:00) -
  STOPT=(12/28/2003,23:00) -
)
```

The following command searches for Process submission records for Processes in which the SNODE is any node in Dallas, and prints the output. Assume that the site administrator indicates that the November statistics records are contained in a file pair whose key sequenced data set is named SYS.ARCH.STATS.M9611. For an explanation of statistics file pairs, search on Administering Statistics in the IBM Connect:Direct for z/OS Administration Guide.

```
SEL STAT WHERE ( -
  TYPE=(PS,SW) -
  SNODE=DALLAS* -
  ARCHDSN=SYS.ARCH.STATS.M9611 -
) PRINT
```

The following command searches for signon and signoff records that match today's date and the User ID SEAN.

```
SEL STAT WHERE ( -  
                TYPE=(SI,S0) -  
                USER=SEAN -  
                STARTT=(TODAY) -  
                STOPT=(TODAY,24:00:00) -  
                ) TABLE
```

The following command searches for the message ID SCMG010I and all message IDs that begin with SCPA.

```
SEL STAT WHERE ( -  
                MSGID=(SCMG010I, SCPA*) -  
                )
```

The following figure illustrates the commands for archiving statistics for a single day using the Batch Interface. These commands archive the statistics from the previous day into the preallocated data set of the API referenced by the DD DMTMPDSN.

```
SIGNON      USERID=(USERID,PASSWORD) TMPDD=DMTMPDSN -  
            NETMAP=HLQ.CD.NETMAP  
SELECT STAT WHERE (STARTT=(YESTERDAY,00:00:00) -  
                  STOPT=(YESTERDAY,24:00:00)) FILE  
SIGNOFF
```

SELECT STATISTICS Sample Output

The following shows sample output from a SELECT STATISTICS command.

```

=====
CD.NODEA                SELECT STATISTICS                DATE : 08.28.2018
=====
Function    => PROCESS SUBMIT          Step Start Time => 08:52:46
Process Name => IOEXIT1                 Step End Time   => 08:52:46
Process Num  => 2                       Comp Code       => 00000000
UserID       => USER11                 Comp Msg        => SSPA001I
Job Name     => USER11                 Job ID          => TSU04387
SnodeID      => No
Secondary Node => CD.NODEB              Step Name       =>
Submitted DSN=> USER11.CD.PROCESS

-----
Function    => COPY                     Step Start Time => 08:52:46
Process Name => IOEXIT1                 Step End Time   => 08:52:50
Process Num  => 2                       Comp Code       => 00000000
SMFID        => SYSK                    Comp Msg        => SCPA000I
Userid       => USER11                 Job ID          => TSU04387
Job Name     => USER11                 Step Name       => PUSH1
Secondary Node => CD.NODEB              Snode Version=> 6.00.00
Pnode Version=> 6.00.00
Other addr   => 10.120.131.92
Other port   => 05610
Session Class=> 001

Session Protocol    => TCP
FASP                => No
  Session is not with SSP
V2 Buffer Size      => 262,144
Negotiated V2 Buffer Size => 262,144
TCP Buffer Size Used    => 262,144
Compression Control Feature: ON
Negotiated: NO Compression

From ( Pnode
Dsn=>IOEXIT (CDINPUT ) SERVICES USED)
DCB => LRECL => 00080  BLKSIZE => 00000  RECFM => FB  DSORG => PS
DISP=> (S,K,K)
      recs => 0                      blks => 5,000
Bytes Read => 10,000,000,000
Bytes Sent => 10,000,020,000
Cmpr Perc => 0.0%
Time on CP => 00:00:01.947
Time on zIIP => 00:00:00.000
zIIP Qualify => 00:00:00.000

```

```

To ( Snode
Dsn=>IOEXIT (CDOUTPUT) SERVICES USED)
DCB => LRECL => 00080  BLKSIZE => 00000  RECFM => FB  DSORG => PS
DISP=> ( , , )
      recs => 0                      blks => 5,000
Bytes Written => 10,000,000,000
Bytes Received=> 10,000,020,000
Cmpr Perc => 0.0%
Time on CP => 00:00:02.874
Time on zIIP => 00:00:00.000
zIIP Qualify => 00:00:00.000

-----
Function    => Process Term          Process Start Time => 08:52:46
Process Name => IOEXIT1              Process End Time   => 08:52:50
Process Num  => 2
Session Class=> 001                  Comp Code       => 00000000
SMFID        => SYSK                  Comp Msg        => SVTM100I
Userid       => USER11
Job Name     => USER11              Job ID          => TSU04387
Secondary Node => CD.NODEB
Time on CP   => 00:00:01.948
Time on zIIP => 00:00:00.000
zIIP Qualify => 00:00:00.000

```

Using SELECT STATISTICS through the IUI

You can use the Select Statistics screen to select, display, and print statistics information from IBM Connect:Direct activities.

About this task

To select, display, or print statistics information using the IUI:

Procedure

1. Select option **SS** from the Primary Options Menu to display the Select Statistics screen.

node.name		SELECT STATISTICS	
CMD ==> P			
CMD: S ... SUMMARY TABLE	D ... DISPLAY REPORT	hh:mm	
P ... PRINT REPORT		yyyy.mm.dd	
		yyyy.ddd	
PROCESS NUMBERS:			
==> -----	==> -----	==> -----	
PROCESS NAMES:			
==> -----	==> -----	==> -----	
START DATE	==> TODAY-----	(YYYY.MM.DD)	
START TIME	==> -----	(HH:MM:SSXM)	
STOP DATE	==> TODAY-----	(YYYY.MM.DD)	
STOP TIME	==> -----	(HH:MM:SSXM)	
CONDITION CODE:	==> -----		
EXCLUDE (MEMBER RECS	==> Y	WTO RECS ==> Y	^WTO RECS ==> N)
CHANGE EXTENDED OPTS:	==> N		

2. To access the Select Statistics Extended Options screen, type Y in the CHANGE EXTENDED OPTS field. On the command line, type the command for the output format that you want. The following table describes each option:

Command	Description
D	Display the output on the screen.
P	Send the output to the printer.
F	View unformatted statistic records.
S	Display a two-line summary per statistics record on the screen. See the Statistics Summary figure on Statistics Summary .

Note: The records from the InterConnect Option are only retrievable through the DISPLAY option.

The following table lists the statistics record types, their corresponding record type identifiers, and whether they display through the Display Report command, or the Change Extended Opts field of the Select Statistics screen.

Record ID	Description	Display Report	Change Extended Options
CE	Copy I/O Start		Y
CH	Change Process	D	Y
CI	Copy Step Start		Y
CS	Statistics Command		Y
CT	Copy Termination	D	Y
CX	Check Certificate Validity		Y
DC	Directory Commands		Y

Record ID	Description	Display Report	Change Extended Options
DP	Delete Process	D	Y
DT	Select Task		Y
DU	Delete User		Y
EI	Event Services Start Command		Y
ET	Event Services Stop Command		Y
EV	Event Services Command		Y
FA	IGWFAMS Message		Y
FI	Long File Name Record	D	Y
FP	Flush Process	D	Y
FS	Suspend Process	D	Y
FT	Flush Task	D	Y
GO	Process Modal - GOTO, ELSE, or EXIT Statement		Y
HW	High Concurrent Session Count		Y
IA	Inquire Statistics		Y
IB	Inquire Debug		Y
ID	Inquire STATDIR		Y
IF	Process Modal - IF Statement		Y
IP	Inquire Initialization parameters		Y
IT	Inquire SNMP Trap Table		Y
IU	Insert User		Y
IX	Inquire IBM Connect:Direct/Plex		Y
JI	Run Job Start		Y
LF	ICO Log File Record		Y
MC	PDS Member Copy	D	Y
M2	Multiple Copy Record	D	Y
NL	Process modal - EIF or PEND statement		Y
NM	NETMAP Updated		Y
PE	IBM Connect:Direct/Plex Error Record		Y
PI	PNODE Process Start		Y
PR	Performance Measurement Record		Y
PS	Process Submit	D	Y
PT	PNODE Process Termination	D	Y
PX	IBM Connect:Direct/Plex Activity (Leave or Join IBM Connect:Direct/Plex)		Y
QE	Queue Change to EXEC Queue		Y

Record ID	Description	Display Report	Change Extended Options
QH	Queue Change to HOLD Queue		Y
QP	Queue Change to Process Retention Queue		Y
QT	Queue Change to TIMER Queue		Y
QW	Queue Change to WAIT Queue		Y
RE	ICO Report Record		Y
RF	Refresh/Update initialization parameters		Y
RJ	Run Job	D	Y
RO	ICO Event Record		Y
RT	Run Task	D	Y
SB	Session Begin		Y
SC	Statistics Control Record		Y
SD	Start IBM Connect:Direct		Y
SE	Session End		Y
SF	Statistics Format		Y
SI	Signon		Y
SN	Select Netmap		Y
SO	Signoff		Y
SP	Select Process		Y
SS	Select Statistics		Y
ST	Stop IBM Connect:Direct		Y
SU	Select User		Y
SW	Submit within a Process	D	Y
SY	SYSOPTS		Y
S2	Statistics Logging Statistics		Y
TF	TCQ Threshold Full		Y
TI	Run Task Start		Y
TL	TCQ Threshold Low		Y
TP	Throughput Record Statistics	D	Y
TR	Trap Event Record		
TS	Suspend Task		Y
TW	TCQ Threshold Warning		Y
UM	Update Network map		Y
UU	Update User		Y
VP	View Process		Y
WO	WTO	D	Y

Record ID	Description	Display Report	Change Extended Options
WS	Select Stat Command		Y
XO	Trace On/Off		Y
ZI	SNODE Process Start	D	Y
ZT	SNODE Process Terminated	D	Y

The following screen is displayed if you selected **Y** in the CHANGE EXTENDED OPTS field.

```

node.name          SELECT STATISTICS EXTENDED OPTIONS
CMD==>  -----
                                                    hh:mm
                                                    yyyy.mm.dd
                                                    yyyy.ddd

RECORD              ==> --      ==> --      ==> --
TYPES:              ==> --      ==> --      ==> --
SNODE NAME:         ==> ----- MESSAGE ID:  ==> -----
USER ID:            ==> -----
FILENAME:           ==> -----
SEARCH              ==> -----
ARCHIVED            ==> -----
DATASETS:           ==> -----
SERVER:             ==> -----
DO YOU WANT THE VALUES OF THIS REQUEST TO BE CASE SENSITIVE? ==> N

```

See the SELECT STATISTICS parameters description on [SELECT STATISTICS Command](#) for information on how to complete the fields, or press PF1 for Help.

Statistics Summary

If you selected **S** to display a statistics summary, the following Statistics Summary screen is displayed:

```

==>  -----STATISTICS SUMMARY-----
                                     ROW 1 TO 3 OF 3
                                     SCROLL ==> PAGE
FUNCTION  PNAME      PNUMBER      ----SUBMITTER NODE-----OTHER.NODE-----
USERID    RTNCD      END DATE/TIME    P|SNODE      SERVER
MSGID
-----
* RUN-TASK  TESTMBR2 5                SC.MVS.QA5A      AS400.CDQA62
  RBELL1                                     SERVER1
  ARTT003I 000000008 MM/DD/YY HH:MM:SS PNODE
S COPY     TESTMBR2 5                SC.MVS.QA5A      AS400.CDQA62
  RBELL1                                     SERVER2
  SCPA001I 000000000 MM/DD/YY HH:MM:SS PNODE
  SUB-CMD  TESTMBR2 5                SC.MVS.QA5A      AS400.CDQA62
  RBELL1                                     SERVER2
  SSPA001I 000000000 MM/DD/YY HH:MM:SS PNODE

```

The first and second lines of each entry identify a Process by function, Process name and number, submitter node, other node, and user ID. The third line is the message ID, return code, ending date and time of that Process, P|SNODE, and IBM Connect:Direct Server that the Process ran on. The P|SNODE field will have either SNODE or PNODE depending on the record type as follows:

- Copy Term (CT) ----- Node type of this record.
- Process Submit (PS) ----- Node it was submitted on (always PNODE).
- Process Term (PT) ----- Node type of this record (always PNODE).
- Process Term (ZT) ----- Node type of this record (always SNODE).
- Run Job (RJ) ----- Node the job was submitted on.
- Run Task (RT) ----- Node the task was run on.

- Submit within Process (SW) - Node the process was submitted on.

An asterisk (*) before the Process name indicates a nonzero return-code. You can type M next to the Process name to display a description of the message ID associated with the Process as in the following figure.

```
node.name          Connect:Direct (TM) MESSAGE DISPLAY          hh:mm
CMD ==>
MSGID ==> SVSL003I
MODULE ==> DMVSOPEN
Copy requested DISP=(,CATLG) to already cataloged dataset.
The PROCESS COPY step requested a DISP=(,CATLG) on the TO
clause of the COPY statement. The requested dataset already
exists as a cataloged dataset.
System Action. The PROCESS COPY step is terminated with a
completion code of 8.
Response: Either correct the COPY dataset or uncatalog the
existing dataset and re-submit the PROCESS.
```

Display Statistics

If you typed **D** to display the output on screen and specified only the CT record type, it is formatted as shown in the following example.

Note: The statistics display function supports 133 character records and horizontal scrolling using the standard ISPF scrolling function keys.

```

=====
CD.NODEA                SELECT STATISTICS                DATE : 08.28.2018
=====
Function    => COPY                      Step Start Time => 14:15:05
Process Name => STATS60A                  Step End Time   => 14:15:47
Process Num  => 5
SMFID       => SYSK                      Comp Code      => 00000000
Userid      => USER11                   Comp Msg       => SCPA000I
Job Name    => USER11                   Job ID        => TSU04406
Secondary Node => CD.NODEB               Step Name     => STEP1
Pnode Version=> 6.00.00                  Snode Version=> 6.00.00
Other addr  => 10.120.131.92
Other port  => 05610
Session Class=> 001

Session Protocol      => TCP
FASP                  => No
  Session is not with SSP
V2 Buffer Size        => 262,144
Negotiated V2 Buffer Size => 262,144
TCP Buffer Size Used   => 262,144
Compression Control Feature: ON
Negotiated: NO Compression

***** CHECKPOINTED;      Interval => 300,000,000
From ( Pnode
  Dsn=>USER11.SMSTEST.DSN1)
  DCB => LRECL => 00080  BLKSIZE => 27920 RECFM => FB  DSORG => PS-E
  DISP=> (S, , )
    recs => 0                      blks => 358,167
  Bytes Read => 10,000,000,000
  Bytes Sent => 10,001,432,668
  Cmpr Perc => 0.0%
  VOL=SER => ARTS09 ARTS0F ARTS0D ARTS08 ARTS06 ARTS0B
            ARTS04 ARTS05
  Time on CP => 00:00:06.658
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000
To ( Snode
  Dsn=>USER11.SMSTEST.DSN2)
  DCB => LRECL => 00080  BLKSIZE => 27920 RECFM => FB  DSORG => PS-E
  DISP=> (R, , )
    recs => 0                      blks => 358,167
  Bytes Written => 10,000,000,000
  Bytes Received=> 10,001,433,436
  Cmpr Perc => 0.0%
  VOL=SER => ARTS0C ARTS0F ARTS0E ARTS0D ARTS08 ARTS09
            ARTS06 ARTS0B
  Time on CP => 00:00:08.773
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000

```

The following figure illustrates how output to the system console and messages in response to console commands are formatted.

```

=====
CD.NODEA                SELECT STATISTICS                DATE : 08.28.2018
=====

14:45:16 SVTM055I SESSION (001) ESTABLISHED WITH          SNODE=CD.NODEB
14:45:16 SVTM036I PROCESS STARTED  STATS60A(              7) SNODE=CD.NODEB
14:45:16 SDAA004I - OBTAIN TSODSN=NO DSN=USER11.SMSTEST.DSN1
14:45:16 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
14:45:16 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
14:45:16 SDAA004I - ALLOC DD=NDM00016 TSODSN=NO RETURN=(DSORG,DEVTYPE,
14:45:16 SDAA004I - VOL,STORCLAS,DATACLAS,MGMTCLAS) DSN=USER11.SMSTEST.DSN1
14:45:16 SDAA004I - DISP=(SHR ) DCB=(BLKSIZE=0000027920,DSORG=PS ,
14:45:16 SDAA004I - LRECL=00080,RECFM=FB ) UNIT=(3390 ,,,DEFER) VOL=(,,
14:45:16 SDAA004I - 001,,)
14:45:16 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFULLY.
14:45:22 SOPD049I Connect:Direct Process,(STATS60A , 7 ) suspended by USER11
14:45:22 SOPD023I SUSPEND Process command by USER11 completed.
14:45:22 SDAA004I - UNALLOC DISP=(,KEEP) DD=NDM00016
14:45:22 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFULLY.
14:45:22 SVTM052I STEP1 COPY STATS60A(              7) SNODE=CD.NODEB
14:45:22 SVTM052I FROM USER11.SMSTEST.DSN1
14:45:22 SVTM052I TO USER11.SMSTEST.DSN2
14:45:22 SVTM052I ##### COMPLETED 0000000C/SCPA046I
14:45:22 SVTM105I PNAME=STATS60A, PNUM= 7 MOVED TO Q=HOLD , QSTATUS=HS
14:45:22 SVTM037I PROCESS SUSPENDED STATS60A(              7) SNODE=CD.NODEB
14:45:22 SVTM056I SESSION (001) TERMINATED WITH          SNODE=CD.NODEB
14:45:24 SOPD049I Connect:Direct Process,(STATS60A , 00000007) changed by USER11
14:45:24 SOPB017I CHANGE Process command by USER11 completed.
14:45:24 SVTM055I SESSION (001) ESTABLISHED WITH          SNODE=CD.NODEB
14:45:24 SVTM036I PROCESS STARTED  STATS60A(              7) SNODE=CD.NODEB
14:45:24 SDAA004I - OBTAIN TSODSN=NO DSN=USER11.SMSTEST.DSN1
14:45:24 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
14:45:24 SDAC006I (LOCATE) - R15=0000, R0=0000, Function completed successfully.
14:45:24 SDAA004I - ALLOC DD=NDM00017 TSODSN=NO RETURN=(DSORG,DEVTYPE,
14:45:24 SDAA004I - VOL,STORCLAS,DATACLAS,MGMTCLAS) DSN=USER11.SMSTEST.DSN1
14:45:24 SDAA004I - DISP=(SHR ) DCB=(BLKSIZE=0000027920,DSORG=PS ,
14:45:24 SDAA004I - LRECL=00080,RECFM=FB ) UNIT=(3390 ,,,DEFER) VOL=(,,
14:45:24 SDAA004I - 001,,)
14:45:24 SDAB005I - ERR=0000, INFO=0000, DYNAMIC ALLOCATION COMPLETED SUCCESSFULLY.

```

Note: In a IBM Connect:Direct/Plex, all write-to-operator (WTO) records display the IBM Connect:Direct Server name (or XCF.NAME for a IBM Connect:Direct Manager) before the time.

Using ADVANCED SELECT STATISTICS through the IUI

You can use the Advanced Select Statistics (S2) screen to select, display, and print statistics information from IBM Connect:Direct activities. The Advanced Select Statistics panel incorporates all fields from both the Select Statistics and the Select Statistics Extended Options panels into one scrollable panel. This makes it easier for those using larger dimension screen sizes to use the dialog. The S2 panel also includes new fields supported by SELECT STATISTICS command specifying the WHERE2 parameter. For more information see, [“SELECT STATISTICS Command Format”](#) on page 611. Since Advanced Select Statistics (S2) screen uses The WHERE2 parameter, the S2 Detail report will differ from the SS report by those record types reported differently by WHERE2. Note that there is more information in the help panel for S2.

About this task

To select, display, or print statistics information using the IUI:

Procedure

Select option **S2** from the Primary Options Menu to display the Advanced Select Statistics screen.

CD.ART	Advanced Select Statistics	08.30.2018 2018.242 14:13:05
CMD ==> D		AutoSave==> YES

Cmnds: S ... Summary Table	D ... Display Report	P ... Print Report
SAVE Save Variables	REST Restore Variables	CLR Clear Variables

Start	Stop	
-----	-----	
Date: -----	-----	(MM.DD.YYYY SU MO TU WE TH FR SA TO YE)
Time: -----	-----	(HH:MM:SSXM)

Process Numbers==>	1 2
Process Names ==>	-----
Record Types ==>	CT SB
/PNODE ==>	-----
/SNODE ==>	-----
From ==>	CD.ART
To ==>	-----
Node/ Local ==>	-----
Names Remote ==>	-----
Submit ==>	-----
Any ==>	-----
Server Names ==>	-----
Message IDs ==>	-----
User IDs ==>	-----
File Names ==>	-----

Condition Code ==>	-- -----
--------------------	----------

Search ==>	-----
Archived ==>	-----
Datasets ==>	-----

Excl Memb Recs ==>	N
Excl WTO Recs ==>	Y
Excl ^WTO Recs ==>	N

Case Sensitive ==>	Y
--------------------	---

DTF Level: 6.00.00

Advanced SELECT STATISTICS Display Report

The IUI Advanced Select Statistics (S2) panel uses the WHERE2 parameter with SELECT STATISTICS command. The WHERE2 formatted statistics records have a common header that makes it easier to find frequently sought report values such as Time and Date, on Local and Remote nodes. The Display Report for record types supported by the WHERE2 parameter is different from Display Report for record types supported by the WHERE parameter.

Note: Record types not supported by WHERE2 parameter continue to display using the same format as the WHERE parameter.

The following table lists the statistics record types and their corresponding record type identifiers that display through the Display Report command of the Advanced Select Statistics screen for WHERE2 parameter.

Record ID	Description
CT	Copy Termination
FI	Long File Name Record
MC	PDS Member Copy
RJ	Run Job
RT	Run Task
PI	PNODE Process Start
PS	Process Submit

Record ID	Description
PT	PNODE Process Termination

To use S2 panel, follow the same procedure as the SS panel, except there are more node field types that you can specify. There is no second panel you must go to if you want to specify any of the less frequently used fields. If no record types are specified, the default types are the same as for Select Statistics (SS).

The following table describes Default records types that display through the Display Report command for Select Statistics (SS) and Advanced Select Statistics (S2) screens:

Record ID	Description
CH	Change Process
CT	Copy Termination
DP	Delete Process
FI	Long File Name Record
FP	Flush Process
FS	Suspend Process
MC	PDS Member Copy
M2	Multiple Copy Record
PS	Process Submit
PT	PNODE Process Termination
RJ	Run Job
RT	Run Task
SW	Submit within a Process
TP	Throughput Record Statistics
ZI	SNODE Process Start
ZT	SNODE Process Terminated

All other record types must be requested specifically. An asterisk (*) requests all types in both SS and S2.

Advanced SELECT STATISTICS sample output

The following figure illustrates an example SELECT STATISTICS WHERE2 report for PS, CT and PT records.

```

=====
CD.NODEA                SELECT STATISTICS                DATE : 08.28.2018
=====
Function (PS)=> Process Submit                Start    Time=> 08:52:46 08.28.2018
Process Name => IOEXIT1                        Stop      Time=> 08:52:46 08.28.2018
Process Num  => 2
Comp Cod+Msg => 00000000 SSPA001I
Submit Job+ID=> USER11    TSU04387
UserID       => USER11
SnodeID      => No
PNode        => CD.NODEA                This Node  => P
SNode        => CD.NODEB
Submit Node  => CD.NODEA
Submitted DSN=> USER11.CD.PROCESS

-----
Function (CT)=> COPY                Start    Time=> 08:52:46 08.28.2018
Process Name => IOEXIT1                Stop      Time=> 08:52:50 08.28.2018
Process Num  => 2
Comp Cod+Msg => 00000000 SCPA000I
Submit Job+ID=> USER11    TSU04387
SMFID        => SYSK
Step Name    => PUSH1
UserID       => USER11
Pnode Version=> 6.00.00                Snode Version=> 6.00.00
PNode        => CD.NODEA                This Node   => P
SNode        => CD.NODEB                From Node   => P
Submit Node  => CD.NODEA
Session Class=> 001
Other addr   => 10.120.131.92
Other port   => 05610

Session Protocol      => TCP
FASP                  => No
Session is not with SSP
V2 Buffer Size        => 262,144
Negotiated V2 Buffer Size => 262,144
TCP Buffer Size Used   => 262,144
Compression Control Feature: ON
Negotiated: NO Compression

From ( Pnode
Dsn=>IOEXIT (CDINPUT ) SERVICES USED)
DCB => LRECL => 00080  BLKSIZE => 00000  RECFM => FB  DSORG => PS
DISP=> (S,K,K)
      recs => 0                blks => 5,000
Bytes Read => 10,000,000,000
Bytes Sent => 10,000,020,000
Cmpr Perc  => 0.0%
Time on CP  => 00:00:01.947
Time on zIIP => 00:00:00.000
zIIP Qualify => 00:00:00.000

```



```

To ( Snode
Dsn=>IOEXIT (CDOUTPUT) SERVICES USED)
DCB => LRECL => 00080 BLKSIZE => 00000 RECFM => FB DSORG => PS
DISP=> ( , , )
      recs => 0          blks => 5,000
Bytes Written => 10,000,000,000
Bytes Received=> 10,000,020,000
  Cmpr Perc => 0.0%
  Time on CP => 00:00:02.874
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000

-----
Function (PT)=> Process Termination      Start   Time=> 08:52:46 08.28.2018
Process Name => IOEXIT1                   Stop    Time=> 08:52:50 08.28.2018
Process Num  => 2                         Submit   Time=> 08:52:46 08.28.2018
Comp Cod+Msg => 00000000 SVTM100I        Schedule Time=> (none)
Submit Job+ID=> USER11 TSU04387
SMFID       => SYSK
UserID      => USER11
PNode       => CD.NODEA                  This Node  => P
SNode       => CD.NODEB
Submit Node => CD.NODEA
Session Class=> 001
  Time on CP => 00:00:01.948
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000

```

SELECT MESSAGE Command

The SELECT MESSAGE command enables you to display message details using the message ID. The SELECT MESSAGE command has the following format and associated parameters. The required parameters and keywords are in bold print. Default values for parameters and subparameters are underlined.

Label	Command	Parameter
(optional)		SElect MeSsaGe WHERE (ID MSGID= message ID) [Short Long]

The following table describes the SELECT MESSAGE command parameters

Parameter	Description
WHERE ID MSGID = message ID	Specifies the 8-character message identification. You can specify a generic message ID.
[Short Long]	Specifies to display the message text in short or long format. The short format is the default.

The following screens show examples of the SELECT MESSAGE command.

To use SELECT MESSAGE with a specific ID:

```
SELECT MSG WHERE (MSGID=SCPA0001) LONG
```

To use SELECT MESSAGE with a generic ID:

```
SELECT MESSAGE WHERE (MSGID=SCPA*)
```

The Network Map

Retrieving Records from the Network Map File

Use the SELECT NETMAP command to retrieve records from the network map file for display or further processing. The network map file is maintained by your system administrator.

Issuing the SELECT NETMAP Command from the IUI

About this task

Use the Select network map screen to select a Network map record.

Procedure

1. Select option **NM** from the Primary Options Menu to display the Select Netmap or TCP Information screen.

```
node.name          SELECT NETMAP OR TCP INFORMATION          hh:mm
CMD ==>
NETMAP INFORMATION
NODE KEY(S): ==> ----- ==> -----
               ==> ----- ==> -----
               ==> ----- ==> -----
               ==> ----- ==> -----
TCP INFORMATION
ADDRESS:-----
NAME:-----
OUTPUT DESTINATION ==> DIS (DIS-DISPLAY,PR-PRINT)
```

2. Type the selection criteria for the nodes you want to examine and press **Enter**. To select all nodes, you can simply press **Enter**.

The following is an example of a network map showing two nodes: CD.P LX44 and CDSELECT.SCO.

```
BROWSE   SYS06097.T144239.RA000.USER01.NDMAPI.H02   Line 00000000 Col 001 080
Command ==>                                           Scroll ==> PAGE
*****
=====
                        SELECT NETWORK MAP
=====
Node Name   : CD.P LX44          VTAM Applid   : M1CDD97G
Max Parsess : 10                Def Ses Class : 1
Session Type: SNA                Environment   :
Node Status : INTERNAL, SEND, RECEIVE
ALT.COMM    : (ALT.DIR=BALANCE   -
                (ALT.ADDR=10.20.129.101,ALT.PORT=13641,
                ALT.TYPE=TCP , ALT.USE.OUT=YES ))
CRC          : DEFAULT          Session Snode Max : 255
PlexClass    : (                ,                ) Use.Server.Node: No
-----
Node Name   : CDSELECT.SCO      TCP PORT      : 13640
TCP Address  : 10.20.246.244
Max Parsess : 20                Def Ses Class : 2
Session Type: TCP                Environment   : UNIX
Source IP    : 10.20.201.2
Node Status  : INTERNAL, SEND, RECEIVE
ALT.COMM     : (ALT.DIR=TOP      -
                (ALT.ADDR=MVSLPARA.CSG.STERCOMM.COM,
                ALT.PORT=01364,ALT.TYPE=TCP ,
                SOURCEIP=10.20.202.22 ,
                ALT.USE.OUT=YES ))
CRC          : DEFAULT          Session Snode Max : 255
Contact      : Joe Dowe
Phone        : (xxx) xxx-xxxxx
Description   : Connect:Direct for z/OS XXXX Node
```

You can make generic requests by using an asterisk. For example, type an asterisk (*) in the first node key field to select all nodes and **D*** to select only those nodes that begin with **D**. See [SELECT NETMAP Command Format](#) or press **PF1** for Help.

Note: The TCP information fields are for creating the SELECT TCPXLAT command described in the next section. Do not fill them in for the SELECT NETMAP command.

Issuing the SELECT NETMAP Command from the Batch Interface

Procedure

- To use the SELECT NETMAP command from the Batch Interface, place your commands in the DGADBATC job stream and submit the job while IBM Connect:Direct is running. The following command displays all network map entries for node names that begin with SAN.

```
SIGNON
SEL NET WHERE (NODE=(SAN*))
SIGNOFF
```

In the following example, the only node name in the network map which begins with SAN is the SANFRAN node. This report contains a long DNS record, which you can define in the Adjacent Node Definition in the network map. For more information on the Adjacent Node Definition, search on Maintaining the Network Map in the *IBM Connect:Direct for z/OS Administration Guide*.

```
=====
                        SELECT NETWORK MAP
=====
NODE NAME   : SANFRAN           VTAM APPLID   : M1A66789
MAX PARSESS : 10                DEF SES CLASS : 1
SESSION TYPE: SNA               ENVIRONMENT   : OS390
NODE STATUS : INTERNAL, SEND, RECEIVE
LONG DNS    : REGIONAL.MANAGER.OFFICE.IN.THE.SAN.FRANCIS
              CO.BAY.AREA.CALIFORNIA.USA
ALT.COMM    : (ALT.DIR=BALANCE   -
              (ALT.ADDR=M1A66790,ALT.TYPE=SNA ,
              ALT.USE.OUT=YES ))
CRC         : DEFAULT           SESSION SNODE MAX : 255
CONTACT     : JOE SMITH
PHONE      : 345-999-6999
DESCRIPTION : SANFRAN TO TOKYO NODES PATH, SECONDARY P
              ATH TO OSAKA.
```

SELECT NETMAP Command Format

The SELECT NETMAP command uses the following format and parameters. The required parameters are in bold print. Default values for parameters and subparameters are underlined>.

Label	Command	Parameters
(optional)	SElect NETMAP	WHERE (NODE = (node generic (list))) P RI nt DI S play

The following table describes SELECT NETMAP parameters:

Parameter	Description
WHERE (NODE = (node generic (list)))	<p>Specifies which network map node definitions you want to examine. WHERE is the only required parameter for the SELECT NETMAP command.</p> <p>NODE=(node generic (list)) specifies the network map node names that are selected. Specify a list of node names by enclosing them in parentheses and separating each by a comma or a blank.</p> <p>node specifies a node name. The node is a 1–16 character alphanumeric string, with the first character alphabetic.</p> <p>generic specifies a generic selection of node names. To specify node names generically, type a 1–7 character alphanumeric string, with the first character alphabetic, plus an asterisk (*). For instance, if your network includes node names PHOENIX, SANDIEGO, SANFRAN, and TUCSON, a specification of SAN* provides information about the SANDIEGO and SANFRAN nodes. If you type only an asterisk (*), IBM Connect:Direct displays (or prints) all nodes that you are authorized to use.</p>
PRint DISplay	Indicates the output format. Indicate only one.

Translating TCP/IP Host Names to Network Addresses

Use the SELECT TCPXLAT command to translate TCP/IP host names to network addresses and TCP/IP addresses to host names.

Issuing the SELECT TCPXLAT Command from the IUI

The Select NETMAP or TCP Information screen issues the SELECT TCPXLAT command or the SELECT NETMAP command depending on your request.

Procedure

1. Select option **NM** from the Primary Options Menu to display the Select Netmap or TCP Information screen. (See Issuing the SELECT NETMAP Command from the IUI to see a screen sample.)
2. Press **Tab** until you reach the TCP INFORMATION portion of the screen. See the parameter descriptions in [SELECT TCPXLAT Command Format](#) or press **PF1** for Help.
3. Do one of the following:
 - To translate a host address to a fully qualified host name, type the address in dot notation form in the ADDRESS field.
 - To translate a host name to a real TCP/IP address, type the host name in the NAME field. You can also type an alias name in this field if the host name is too long for the space provided.

The following output is displayed:

```

xx.abc.node111          RESOLVED TCP NAME/ADDRESS          14:56
CMD ==>
TCP name/address translation successful.
TCP ADDRESS : 111.222.333.444
FULLY QUALIFIED TCP HOST NAME:
qaoptsol.csg.stercomm.com
----- $$$$ -----

```

Issuing the SELECT TCPXLAT Command with the Batch Interface

To use the SELECT TCPXLAT command from the Batch Interface, place your commands in the DGADBATC job stream and submit the job while IBM Connect:Direct is running. Use the batch version of the SELECT TCPXLAT command to make multiple requests for resolution. The maximum number of requests is 10.

Procedure

- To use the SELECT TCPXLAT command from the Batch Interface, place your commands in the DGADBATC job stream and submit the job while IBM Connect:Direct is running. Use the batch version of the SELECT TCPXLAT command to make multiple requests for resolution. The maximum number of requests is 10.

The following command produces the TCP Name/Address Translation Report. The first two names listed in the TRTCPNAM parameter are aliases, the third is an actual host name.

```
SIGNON
  SEL TCPXLAT WHERE (TRTCPNAM=(NAME1, NAME2, tcpip.host.name3))
  SEL TCPXLAT WHERE (TRTCPADR=(123.456.78.999, 123.456.789.012))
SIGNOFF
```

The following report shows the resolution of name to address or address to name. REQUEST is what is typed in the command.

```
=====
                        TCP Name/Address Translation Report
=====
REQUEST                  : NAME1
Host Address              : NNN.NNN.NN.NNN..
HOST NAME                 : tcpip.host.name1
REQUEST                  : NAME2
*** TCP Name/Address could not be resolved ***
REQUEST                  : tcpip.host.name3
Host Address              : NNN.NNN.NN.NNN..
HOST NAME                 : tcpip.host.name3
=====
                        TCP Name/Address Translation Report
=====
REQUEST                  : 123.456.78.999
Host Address              : 123.456.78.999,
HOST NAME                 : tcpip.host.name4
REQUEST                  : 123.456.789.012
Host Address              : 123.456.789.012,
HOST NAME                 : tcpip.host.name5
```

SELECT TCPXLAT Command Format

The SELECT TCPXLAT command uses the following format and parameters. The required parameter is in bold print. Default values for parameters and subparameters are underlined>.

Label	Command	Parameters
(optional)	SE lect TCPXLAT	WHERE (TRTCPNAM = (tcpip.host.name (list)) TRTCPADR = (tcp.net.adr (list))) PRint DISplay

The following table describes the SELECT TCPXLAT command parameters:

Parameter	Description
<p>WHERE (TRTCPNAM = (tcpip.host.name (list)) TRTCPADR = (tcp.net.adr (list)))</p>	<p>Specifies which TCP host name or network address resolution you want to examine. This parameter is required. Provide one of the following subparameters:</p> <p>TRTCPNAM = (tcpip.host.name (list)) specifies a translation from host name or alias to a real address. Type this name in the standard format with each of the qualifiers being from 1 to 64 alphanumeric characters, with a maximum name length of 64 characters. Specify a list of names by enclosing them in parentheses and separating each by a comma or a blank.</p> <p>TRTCPADR = (tcp.net.adr (list)) specifies a translation from an IP address to the host name for that network. Type this address in the dotted format (nnn.nnn.nnn.nnn) with a maximum length of 15 characters including periods. Specify a list of addresses by enclosing them in parentheses and separating each by a comma or a blank.</p>
PRint DISplay	Specifies where output is directed. Indicate only one. When you submit this command through DGADBATC, use the PRINT parameter only.

Utility Programs

IBM Connect:Direct provides several utility programs and one I/O exit that facilitate your use of the software. You can perform the following functions using these programs and the exit:

- Notify the user of Process success or failure using DGADTIFY and DGADTFY2
- Dynamically allocate data sets using DGADTDYN
- Place job stream in a wait status for a specified period of time and set return codes indicating whether the batch step or RUN TASK completed successfully or failed by using the DGADWAIT program.
- Dynamically invoke AMS (Access Method Services) to perform VSAM utility functions using DGADTAMS
- Resolve symbolic references using DGADTSUB and DGADGSUB
- Compress and decompress files stored in a ZLIB compressed format (DGASACMP)
- Determine how many concurrent sessions are running during a specific time period (DGADVITL)
- Interface with the IBM utility ADDRDSU, by using the IBM Connect:Direct I/O exit, DGADSIQX. The ADDRDSU program enables you to copy SMS-compressed data without having to decompress the data and also provides support for copying wildcard-named files.
- Replace existing FTP sessions within z/OS job streams with Connect:Direct for z/OS Processes using IBM Sterling Connect:Direct FTP+ for z/OS (CDFTP).

User Notification Programs

IBM Connect:Direct provides sample programs that notify the console operator or IUI users about the success or failure of any Process step. The sample programs are supplied in source statement form in the IBM Connect:Direct SDGASAMP library and in load module form in IBM Connect:Direct SDGALINK. Use RUN TASK and MODAL statements with the sample programs to notify users.

The messages sent are standard TRANSFER SUCCESSFUL or TRANSFER FAILED status messages.

Operator Console Notify Program (DGADTIFY)

The operator console program, DGADTIFY, works with the RUN TASK statement to notify the console operator of the success or failure of Processes by means of Write to Operator (WTO) messages.

The program must have the following parameters passed to it:

Parameter	Description
CL4'GOOD' CL4'FAIL'	The first positional parameter tells the program the correct message to output. Code the parameter as illustrated in the following example, including the single quotes.
file name	The second positional parameter is the name of the file that is being copied.

The following example shows a Connect:Direct for z/OS Process using the DGADTIFY program.

```

COPY1 PROCESS PNODE=CDA SNODE=CDB
STEP01 COPY -
        FROM ( -
                PNODE -
                DSN=FILE1 -
                DISP=SHR -
                ) -
        TO ( -
                SNODE -
                DSN=FILE2 -
                DISP=SHR -
                )
STEP02 IF (STEP01=0) THEN
NOTIFYG RUN TASK (PGM=DGADTIFY, -
                PARM=(CL4'GOOD',FILE1)) PNODE
        ELSE
NOTIFYF RUN TASK (PGM=DGADTIFY, -
                PARM=(CL4'FAIL',FILE1)) PNODE
EIF

```

The DGADTIFY program is attached on the PNODE. A parameter list containing the GOOD or FAIL message criteria and the FILE1 file name is passed to DGADTIFY. If the COPY statement is successful, the first RUN TASK step executes and you see the following message:

```

SNTB001I ** CONNECT:DIRECT SUCCESSFUL STEP COMPLETION **
FILE XFER BETWEEN - PLEX.JOE AND - PLEX.TOM SUCCESSFUL
REQUESTOR=CTENN1
DSN=CD.OUTPUT.data set

```

If the COPY statement is unsuccessful, the second RUN TASK step executes and you see the following messages:

```

SNTB002I ** Connect:Direct FAILED STEP COMPLETION **
***** Connect:Direct FILE TRANSFER FAILED *****
***** Connect:Direct FILE TRANSFER FAILED *****
FILE XFER BETWEEN - PLEX.JOE AND - PLEX.TOM FAILED
REQUESTOR=CTENN1
DSN=CD.OUTPUT.data set
***** Connect:Direct FILE TRANSFER FAILED *****
***** Connect:Direct FILE TRANSFER FAILED *****

```

User Notify Program (DGADTFY2)

Use the sample program, DGADTFY2, in conjunction with the RUN TASK statement to notify TSO users of various conditions. The program accepts a variable length parameter list to broadcast notification messages to users.

Note: The TSO user ID must be in the same z/OS image as the DTF where the DGADTFY2 RUN TASK is running.

The following table describes the parameters:

Parameter	Description
CL4'GOOD' CL4'FAIL'	This first positional parameter indicates the success or failure of the Process step. Code the parameter as illustrated in the following example, including the single quotes.
file name	This second positional parameter is the name of the file that is being copied.
user ID user ID list	The last parameter or list of parameters contains the TSO user IDs to notify. Separate the user IDs with a comma.

The following example demonstrates a Process using the DGADTFY2 program.

```

COPY  PROCESS  PNODE=CDA SNODE=CDB
STEP01COPY  -
      FROM (PNODE -
            DSN=FILE1 -
            DISP=SHR) -
      TO (SNODE -
          DSN=FILE2 -
          DISP=RPL )
STEP02 IF (STEP01=0) THEN
NOTIFY1  RUN TASK -
          (PGM=DGADTFY2,PARM=(CL4'GOOD',FILE1,CDID1,CDID2))SNODE
      ELSE
NOTIFY2  RUN TASK -
          (PGM=DGADTFY2,PARM=(CL4'FAIL',FILE1,CDID1,CDID2))PNODE
EIF

```

A GOOD value is relayed to the TSO users CDID1 and CDID2 on CDB in the following message:

```

SNTA001I **Connect:Direct FILE TRANSFER SUCCESSFUL**
TRANSFER BETWEEN - CDA AND - CDB
DSN=FILE1

```

A FAIL value is relayed to the TSO users CDID1 and CDID2 on CDA in the following message.

```

SNTA002I ****Connect:Direct FILE TRANSFER FAILED****
TRANSFER BETWEEN - CDA AND - CDB
DSN=FILE1

```

Dynamic Allocation Program (DGADTDYN)

The dynamic allocation program, DGADTDYN, performs several useful status checking and housekeeping functions in a Process. It is located in the IBM Connect:Direct SDGALINK.

When multiple actions are contained in the parameters, the completion code returned is the highest completion code encountered.

DGADTDYN Parameters

The RUN TASK command invokes DGADTDYN with parameters that define allocation actions to execute. The following rules apply to the syntax of the parameters:

- Perform each allocation action with as many parameters as required.
- Separate each unique allocation action by a parameter containing a fullword of minus one (F'-1').
- A leading blank on the first parameter is not required; however, you must include a leading blank on all subsequent parameters. If you do not include a leading blank, the program issues an RC=30 and an SRTA005I message for the Process.
- Parameters must not extend past column 72, or they are not read.

The following table shows the available dynamic allocation functions:

Function	Description
ALLOC	Use the ALLOC function to allocate a file. The allocation request can contain most job stream DD statement options, including DCB options applicable to allocation. This function is the default.
UNALLOC	Use the UNALLOC function to unallocate a file.
CONCAT	Use the CONCAT function to concatenate two files.
DECONCAT	Use the DECONCAT function to perform dynamic deconcatenation.
LOCATE	Use the LOCATE function to locate cataloged data sets.

Note: You cannot access HFS files using DGADTDYN.

DGADTDYN Sample Program Uses

The following examples demonstrate how you can use the functions of the DGADTDYN program.

Example 1: Locate and Allocate a Data Set

The following example calls the program DGADTDYN using the LOCATE parameter to determine if a cataloged file exists.

If the file does not exist, the program issues a nonzero return code and makes a call to allocate the file through the ALLOC parameter.

```

PROC1 PROCESS SNODE=BOSTON
LOCATE      RUN TASK(PGM=DGADTDYN -
      PARM=(C'LOCATE DSN=XYZ.CUSTOMER.FILE'))SNODE
      IF (LOCATE NE 0) THEN
ALLOC       RUN TASK (PGM=DGADTDYN -
      PARM=(C'ALLOC', -
      C' DSN=XYZ.CUSTOMER.FILE', -
      C' DISP=(NEW,CATLG)', -
      C' VOL=SER=DCM009', -
      C' SPACE=(CYL,(1,1))', -
      C' DSNTYPE=EXTDPREF', -
      C' DCB=(RECFM=FB,DSORG=PS,LRECL=80,BLKSIZE=3120)', -
      C' UNIT=3380'))SNODE
      EIF

```

Example 2: Delete and Uncatalog a Data Set

The following example deletes and uncatalogs a data set.

The first function, ALLOC, allocates the data set specifying a current status of OLD and a secondary (or normal step completion) disposition of DELETE, which deletes the data set.

The second function, UNALLOC, unallocates the data set. Because of the disposition (the data set is originally allocated with disposition DELETE), the data set is uncataloged.

Use the F'-1' as a separator between allocation functions.

```

PROC2 PROCESS SNODE=BOSTON
DELETE RUN TASK(PGM = DGADTDYN -
      PARM = (C'ALLOC DSN=XYZ.TEMP DISP=(OLD,DELETE)' -
      F'-1' -
      C'UNALLOC DSN=XYZ.TEMP'))

```

Example 3: Allocate and Concatenate Data Sets

The following example calls the program to allocate each data set through the ALLOC parameter. Then it calls DGADTDYN again to do the concatenation through the CONCAT parameter. The data set is concatenated under the first DDNAME in the CONCAT list.

```
RTDYNCON PROCESS SNODE=BOSTON
ALLOC1 RUN TASK (PGM=DGADTDYN -
  PARM=(C'ALLOC DSN=XYZ.TEMP DISP=SHR DD=OLD1')) PNODE
ALLOC2 RUN TASK (PGM=DGADTDYN -
  PARM=(C'ALLOC DSN=XYZ.TEMP2 DISP=SHR DD=OLD2')) PNODE
CONCAT RUN TASK (PGM=DGADTDYN -
  PARM=(C'CONCAT DD=(OLD1,OLD2)')) PNODE
```

Example 4: Deconcatenate and Unallocate Data Sets

The following example calls the program to deconcatenate by DDNAME using the DECONCAT parameter. Then it calls DGADTDYN again to unallocate each data set in the concatenation with the UNALLOC command.

```
RTDYNCON PROCESS SNODE=BOSTON
DECONCAT RUN TASK (PGM=DGADTDYN -
  PARM=(C'DECONCAT DD=(OLD1)')) PNODE
UNALLOC1 RUN TASK (PGM=DGADTDYN -
  PARM=(C'UNALLOC DD=OLD1')) PNODE
UNALLOC2 RUN TASK (PGM=DGADTDYN -
  PARM=(C'UNALLOC DD=OLD2')) PNODE
```

Run Task Wait Program (DGADWAIT)

Use the DGADWAIT program as a batch step or a RUN TASK to place a job stream in a wait status for a specified period of time and set return codes indicating when the step or task completes. To synchronize Processes submitted through the batch interface, use DGADWAIT with the MAXDELAY parameter in a SUBMIT command or PROCESS statement. For more information, see [Using the MAXDELAY Keyword Parameter to Synchronize Submitted Processes](#). This program is distributed via SDGASAMP and must be assembled and link-edited to the IBM Connect:Direct load library.

The following table describes the positional parameters. If you execute DGADWAIT as a RUN TASK, enclose the parameters in quotes because of the embedded commas, for example, PARM=('1,2').

Parameter	Description
VALUE1	The first positional parameter indicates the time to wait. The default is 0 or you can use a comma to omit. For seconds, specify (S)xxxx. The maximum is 9999 seconds or about 2.7 hours. For minutes, specify Mxxx. The maximum is 999 minutes or about 16.6 hours. For hours, specify Hxx. The maximum is H99 or about 4.1 days. You can also specify 'ABEND' and then use the actual user ABEND code for the second positional parameter.
VALUE2	The second positional parameter specifies the return code to set. The default is 0. The maximum is 4095. '-' causes a -1 value (if executed as a batch step, IBM displays the return code value as 4095).
VALUE3	The third positional parameter is only used to display diagnostics as WTO (write-to-operator) messages. Specify 'DIAG' or leave blank. Any other value is ignored.

The following examples show how to use DGADWAIT in a RUN TASK. In the first example, the program waits for one second, and then sets the return code to 2.

```
DGADWAIT PROCESS SNODE=NODE.NAME &PRM='1,2'
STEP01 RUN TASK (PGM=DGADWAIT PARM=(&PRM))
```

In this example, the program does not wait, but sets the return code to 8.

```
DGADWAIT PROCESS SNODE=NODE.NAME &PRM='0,8'
STEP01 RUN TASK (PGM=DGADWAIT PARM=(&PRM))
```

VSAM AMS Interface Program (DGADTAMS)

The VSAM AMS interface program, DGADTAMS, dynamically invokes AMS to perform VSAM utility functions. The program is located in the IBM Connect:Direct SDGALINK.

The highest completion code encountered in any AMS action is returned.

DGADTAMS Parameters

DGADTAMS is invoked by the RUN TASK Process statement with the following parameters that define allocation and AMS actions to execute.

Parameter	Description
sysprint output parameters	<p>The first positional parameter is special because it defines dynamic allocation parameters for the AMS SYSPRINT output. This parameter can allocate a spool file or a user file. The FREE and RETURN subparameters are required.</p> <p>Note: You must specify all of the sysprint output parameters on the same line. They cannot be on separate lines.</p> <p>The following example defines a CLASS A SYSOUT spool file.</p> <p>C'FREE=CLOSE, RETURN=(DD) SYSOUT=A'</p>
control statement parameters	<p>You can use any valid input to the VSAM AMS program as input to DGADTAMS, even multiple input parameters defining one or more AMS actions. You must begin all VSAM control statement parameters with a blank.</p>

DGADTAMS Sample Program

The following figure shows a sample of DGADTAMS using the Define Cluster AMS function. The ASM SYSPRINT output is routed to the existing user file XYZ.OUTPUT.

```
VADC1 PROCESS SNODE=BOSTON
DEFINE RUN TASK (PGM=DGADTAMS,
  PARM=(C'FREE=CLOSE,RETURN=(DD),DISP=OLD,DSN=XYZ.OUTPUT',
    C' DELETE(VSAM.PAY.CHECKS) CLUSTER',
    C' DEFINE CLUSTER',
    C' (NAME(VSAM.PAY.CHECKS))',
    C' RECORDS(1)',
    C' VOLUMES(DCM009)',
    C' OWNER(NDM)',
    C' NONINDEXED',
    C' RECORDSIZE(4089 4089)',
    C' SHAREOPTIONS (2))',
    C' DATA',
    C' (CONTROLINTERVALSIZE(4096))',
    C' NAME(VSAM.PAY.CHECKS.DATA))',
    C' CATALOG(USER.UCAT)')) SNODE
```

Symbolic Resolution Utilities (DGADTSUB and DGADGSUB)

The IBM Connect:Direct utility programs, DGADTSUB and DGADGSUB, enable you to submit jobs to the internal reader, much like RUN JOB. DGADTSUB has the added capability of symbolic resolution into the job. The RUN TASK statement invokes DGADTSUB or DGADGSUB, which allows you to substitute variables in the parameter list.

The following table lists DGADTSUB and DGADGSUB parameters, and DGADGSUB variables:

Parameter	Description
JCL source	The first parameter is an allocation string to point to the job to submit. Code it without a ddname because DGADTSUB acquires one. IBM Connect:Direct uses this value, along with the DISP parameter, to allocate the data set that contains the JCL.
Subsequent Parameters	<p>Subsequent parameters define resolutions for symbols used in the JCL itself. For each parameter, the first position is the name of the symbol found in the JCL and the second position is the value used. Separate the first and second positions by an equal sign (=), one or more spaces, or a comma. The value is terminated by the end of the parameter string, or a comma or space not within a quoted string. A parameter can also contain only a symbol name, in which case the value to substitute into the job stream is a null string.</p> <p>Note: To pass lowercase data to the DGADTSUB program, use SYSOPTS in the RUN TASK statement.</p>
Variables (for use with DGADGSUB)	<p>%SRCDSN The resolved 'FROM' data set name from the last COPY step executed in this Process</p> <p>%DSTDSN The resolved 'TO' data set name from the last COPY step executed in this Process</p> <p>%PROCNAME The name of this Process</p> <p>%PROCNUM The number of this Process</p> <p>Note: To use %SRCDSN or %DSTDSN, the RUN TASK must be run on the PNODE. The values for the %SRCDSN and %DSTDSN variables are not available when the RUN TASK is run on the SNODE.</p>

The following figure is an example of RUN TASK using the DGADTSUB program.

```
RUN TASK (PGM=DGADTSUB -
          PARM=('DSN=dsname,DISP=SHR', -
              'JOB X21JOB', -
              'RPTNAME REPORT01', -
              'CLASS A', -
              'FILE X21.FILE' -
          )) PNODE
```

The job stream source that corresponds to this RUN TASK statement is displayed in the following figure. The values requiring substitution are presented in bold type.

```
//&JOB JOB (79502),&RPTNAME,PTY=12,TIME=(1),CLASS=&CLASS,
// REGION=512K,MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=BSMITH1
//PRINT EXEC PGM=IEBGNER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY,DCB=BLKSIZE=80
//SYSUT1 DD DISP=SHR,
// DSN=&FILE
//SYSUT2 DD SYSOUT=&CLASS,DCB=(&FILE)
```

The following figure shows the resulting job stream after substitution has taken place. IBM Connect:Direct replaces the variables that began with & with the assigned values in the DGADTSUB parameter list.

```
//X21JOB JOB (79502),REPORT01,PRTY=12,TIME=(1),CLASS=A,
// REGION=512K,MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=BSMITH1
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY,DCB=BLKSIZE=80
//SYSUT1 DD DISP=SHR,
// DSN=X21.FILE
//SYSUT2 DD SYSOUT=A,DCB=(X21.FILE)
//*
```

Passing Variables Using DGADGSUB

You can use the DGADGSUB program, which substitutes variables in the parameter list and then calls the DGADTSUB program to process the parameter list further.

Note: DGADGSUB is not intended to concatenate variables, only to substitute them. Variable concatenation is handled by the parser at PROCESS submit time. DGADGSUB is intended to pass variables to JCL to be substituted into the job.

To do this, modify an existing RUN TASK statement that calls DGADTSUB to call DGADGSUB, and add the variables you want to use in the parameter list.

The following figure is an example of RUN TASK using the DGADGSUB program.

```
STEP01 RUN TASK(PGM=DGADGSUB -
    PARM=('DSN=DALLAS1.ZOS.SRCLIB(JCLTEST1),DISP=SHR', -
        'JOB &JOBNAME', -
        'SRC %SRCDSN', -
        'DST %DSTDSN', -
        'PROCNAME %PROCNAME', -
        'PROCNUM %PROCNUM', -
        'DSN DSNNAME')) PNODE
```

In this example, the following circumstances are in effect:

- The 'FROM' data set in the last COPY step before this RUN TASK is DALLAS1.FILE
- The 'TO' data set in the last COPY step before this RUN TASK is DALLAS1.TEST.GDG.G0006V00
- The name of the Process is 'TESTPR'
- The Process number is 29
- The &JOBNAME is XYZ

The DGADGSUB program makes the variable substitutions and calls the DGADTSUB program with the following parameters:

```
PARM=('DSN=DALLAS1.ZOS.SRCLIB(JCLTEST1),DISP=SHR', -
    'JOB XYZ', -
    'SRC DALLAS1.FILE', -
    'DST DALLAS1.TEST.GDG.G0006V00', -
    'PROCNAME TESTPR', -
    'PROCNUM 00029', -
    'DSN DSNNAME'))
```

Wherever &JOB, &SRC, &DST, &PROCNAME, &PROCNUM, and &DSN are found in DALLAS1.ZOS.SRCLIB(JCLTEST1), the corresponding values are substituted. Once all the substitutions have been made, the resultant JCL is submitted to JES.

Using Symbolic Variables

It can also be useful to specify the values to substitute as symbols rather than their actual value. You can then use a single Process for multiple purposes.

The following figure is an example of DGADTSUB that symbolically substitutes values into a z/OS job being submitted to the internal reader. IBM Connect:Direct supports any symbols that begin with an ampersand.

In the example, the second and subsequent parameters name the symbol, without the &, and value that the symbol becomes in the submitted job stream. IBM Connect:Direct substitutes the value itself when the Process is submitted.

```
DGADTSUB PROC SNODE=CD.OS39040.N1 NOTIFY=USER01 -
    &CLASS=A &DEST=LOCAL &JOB=JOB1SB -
    &RPTNAME='''RPT NAME''' -
    &FILE=X21.FILE -
    &JCL=X21.CNTL(SUBSUB)
*
*
    RUN TASK (PGM=DGADTSUB -
        PARM= ("DSN=dsname, DISP=SHR", -
            "FILE          &FILE", -
            "JOB           &JOB", -
            "CLASS        &CLASS", -
            "DEST         &DEST", -
            "RPTNAME      &RPTNAME", -
            )) PNODE
```

IBM Connect:Direct Exits and DGADTSUB

If your IBM Connect:Direct system uses any of the standard exits (Security, RUN JOB, or RUN TASK), DGADTSUB functions with any or all of them without further modification. Both a RUN TASK and a RUN JOB exit are invoked by DGADTSUB, and any processing that takes place is done when using DGADTSUB.

Error Output

DGADTSUB writes any internal error information to the data set specified on the CDESTAE DD statement. Always include this DD statement in the IBM Connect:Direct startup job stream.

Return Codes

The following table describes the DGADTSUB return codes:

Return Code	Meanings
04	<p>A job stream record is truncated after substitution. Symbolic substitution is performed on columns 1–71 of the JCL record. Truncation is indicated when substitution causes the data in columns 1–71 to expand beyond column 71.</p> <p>For non-JCL records, substitution is performed on columns 1–80. The most common cause of this error is sequence numbers in data statements.</p>
08	<p>An error occurred parsing the parameters or the job stream source.</p> <p>A symbol length is greater than 19.</p> <p>A symbol name is absent in the parameter string.</p>

12	<p>No parameters are passed.</p> <p>An I/O error occurred.</p> <p>An input data set allocation parameter is not found.</p> <p>The symbol table overflowed.</p> <p>A file open error occurred.</p> <p>An input record length is greater than 256.</p> <p>A member locate error occurred.</p>
----	---

Batch Compression Utility (DGASACMP)

DGASACMP is a batch utility that compresses and stores files in ZLIB-compressed format. You can transfer these compressed files using a IBM Connect:Direct Process and store them on the remote node as compressed data. A user can execute DGASACMP on the remote node to decompress the data and store it in the original format. These files cannot be decompressed using any other utility.

Note: You can also decompress files using a IBM Connect:Direct Process. Refer to [Automatic Decompression](#) for more information.

DGASACMP supports physical sequential files (PS), KSDS, ESDS, and RRDS VSAM files. DGASACMP cannot allocate VSAM files as DISP=(NEW,CATLG), so you must pre-define the VSAM output file when decompressing a file that was originally VSAM.

DGASACMP does not support PDS (or PDSE) files as PDS, but you can use IEBCOPY to unload the PDS, then compress or decompress using DGASACMP.

DGASACMP Parameters

You can pass the following parameters to DGASACMP:

Parameter	Description
MODE= COMP DECOMP	Specify COMP to compress an input file. Specify DECOMP to decompress a compressed file.
REPORT=NORMAL DEBUG	Specify NORMAL to instruct DGASACMP to produce a summary statistics report. This value is the default. Specify DEBUG to instruct DGASACMP to produce a detailed statistics report.
ZLIB=(CMP= <u>1</u> n, WIN= <u>13</u> nn, MEM= <u>4</u> n)	<p>These three parameters change the settings of the extended compression utility. CMP indicates the compression level. Valid values are 1–9. The default value is 1. WIN indicates the window size. Valid values are 9–15. The default value is 13. MEM indicates the memory. Valid values are 1–9. The default value is 4.</p> <p>Note: Changing these values can significantly impact performance and CPU overhead. Before you change the default values, see Testing the Effects of Changing Values for Extended Compression Parameters .</p>

DGASACMP Examples

The following sample JCL, found as member DGAXCMP in \$CD.SDGASAMP, compresses a physical sequential (PS) file and stores it as compressed data.

```
//CDCMP JOB (1004), 'BATCH COMPRESS', CLASS=N, MSGCLASS=X,
//          COND=(1,LT), REGION=0M
//*
//* ZLIB COMPRESS A FILE IN BATCH
//*
//COMP      EXEC  PGM=DGASACMP, PARM='MODE=COMP'
//STEPLIB   DD    DISP=SHR, DSN=$CD.SDGALINK
//INPUT      DD    DISP=SHR, DSN=$CD.FILE1
//OUTPUT     DD    DSN=$CD.FILE1.COMP, DISP=(NEW,CATLG),
//                UNIT=SYSDA, SPACE=(TRK,(1500,300),RLSE),
//                DCB=(BLKSIZE=27920, RECFM=U, DSORG=PS)
//SYSOUT     DD    SYSOUT=*, DCB=(BLKSIZE=133)
//*
```

The following sample JCL, found as member DGAXDCMP in \$CD.SDGASAMP, decompresses a compressed file and stores it in the original format.

```
//CDDECMP JOB (1004), 'BATCH COMPRESS', CLASS=N, MSGCLASS=X,
//          COND=(1,LT), REGION=0M
//*
//* ZLIB DE-COMPRESS A FILE IN BATCH
//*
//DECOMP     EXEC  PGM=DGASACMP, PARM='MODE=DECOMP'
//STEPLIB    DD    DISP=SHR, DSN=$CD.SDGALINK
//INPUT       DD    DSN=$CD.FILE1.COMP, DISP=SHR
//OUTPUT      DD    DSN=$CD.FILE1.NEW, DISP=(NEW,CATLG),
//                UNIT=SYSDA, SPACE=(TRK,(1500,300),RLSE),
//                DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SYSOUT      DD    SYSOUT=*
//*
```

DGASACMP Output

The following screen shows sample output from DGASACMP.

```
SAC0000I DGASACMP started, GMT =2001.275 19:39:25.00
SAC0001I OS Parm: MODE=COMP,
SAC0002I Options in effect: MODE=COMP  CMP= 1    WIN= 13    MEM= 4    REPORT=NORMAL
SAC0003I ZLIB Version=1.2.3
SAC0011I INPUT DSORG=PS    RECFM=FB    RECSZ=      80    DSN=CD.TESTFILE.M1
SAC0012I OUTPUT DSORG=PS   RECFM=U     RECSZ= 27,920    DSN=CD.TESTFILE.M1.COMP
SAC0021I Total Records Read      =      12,500
SAC0022I Total Records Written   =           5
SAC0023I Total Bytes Read        =    1,000,000
SAC0024I Total Bytes Written     =    124,584
SAC0031I Compression Percent     =      88%
SAC0032I Compression Ratio       =      8.0:1
SAC0033I TCB CPU Seconds         =      2.627
SAC0034I Elapsed Seconds         =      4.103
SAC0999I DGASACMP utility completed successfully
```

You can also invoke DGASACMP with the OUTPUT DD as DUMMY, indicating that the summary report is produced, but the output data is not stored. You must still provide the DCB attributes. The following screen shows an example.

```
//OUTPUT     DD  DUMMY,
//            DCB=(BLKSIZE=27920, RECFM=U, DSORG=PS)
```


Testing the Effects of Changing Values for Extended Compression Parameters

Before you change the default values of the extended compression parameters when using the DGASACMP utility, search on Improving Performance in the IBM Connect:Direct for z/OS Administration Guide and review the benchmark figures showing different test results. You can also use this utility offline to determine the benefits of changing the default values in a test environment before actually using compression on live data in your production environment.

About this task

Although extended compression using ZLIB is available on a global basis using the ECZ initialization parameters and on a Process basis using the EXT parameters in the COPY statement, performing compression online consumes significant CPU resources. The DGASACMP utility produces a report (see the sample report in [DGASACMP Output](#)), which shows how much the data was read, written, compressed, and how long it took to compress. Another report is produced when you use DGASACMP to decompress data and restore it to its original format.

To use DGASACMP to test compression, follow this procedure:

Procedure

1. Prepare a file containing the typical type of data you transfer and the average amount of data involved in a transfer.
2. Transfer the file without using compression and record the time it takes for the transfer to complete.
3. To test the results of compressing and decompressing the file, run DGASACMP three times on the test file using the following settings. After each compression run, run DGASACMP to decompress the file. Keep the output reports for each test to compare results.
 - CMP=1,WIN=13,MEM=4 (the default settings)
 - CMP=1,WIN=1,MEM=14)
 - CMP=8,WIN=1,MEM=14)
4. To determine if it is worthwhile to use compression and what extended compression parameter values are most beneficial for your environment, examine the test results taking the following factors into consideration:
 - Amount of CPU and elapsed time it takes to compress and decompress the data
 - Amount of time it takes to send the data
 - Virtual memory space used to maintain the ZLIB internal control blocks
 - Virtual memory space used to allocate the compression window or history buffer
 - Type of data being sent, including its compressibility

Automatic Decompression

As an alternative to storing the data as compressed data on the remote node, you can decompress a file during Process execution. This functionality is available between z/OS and any other platform that supports ZLIB compression.

To perform automatic decompression, specify PRECOMP=YES in the SYSOPTS parameter in the FROM clause of the COPY statement. PRECOMP=YES indicates that the FROM data set is precompressed and instructs IBM Connect:Direct to decompress the file as part of the Process. If you do not code SYSOPTS or set PRECOMP to NO, the file is sent in compressed format and the receiver must run DGASACMP with MODE=DECOMP.

Note: If you are decompressing a file using DGASACMP, you cannot allocate VSAM files as DISP=(NEW,CATLG). You must predefine the VSAM output file.

The following sample Process, found as member DGAPMPPS in \$CD.SDGAPROC, instructs IBM Connect:Direct to decompress a file that is compressed using DGASACMP.

Note: Refer to the \$CD.SDGASAMP data set for more sample Processes.

```
CDCOMPPS PROCESS SNODE=CD.OS390
STEP1 COPY FROM (DSN=$CD.FILE1.COMP -
              DISP=(SHR) -
              SYSOPTS='PRECOMP=YES' -
              PNODE) -
TO (DSN=$CD.FILE1.NEW -
   DISP=(RPL) -
   SNODE)
```

Determine “High-Water Mark” for a Period (DGADVITL)

The program called DGADVITL reports how many concurrent sessions (Processes) are running during the time period specified. Below is a sample RUN TASK that executes as a PNODE=SNODE Process when submitted. This Process must be force flushed to discontinue. The MCS console displays the number of sessions that ran during the specified period, and the time and date the high water mark was reached. These messages appear in the JESMSG LG segment of the IBM Connect:Direct SYSOUT.

```
/*****/
/* RUN TASK FOR HIGH WATER PROCESS NUMBER; PNODE=SNODE */ /*****/
*****/
RTBB PROCESS SNODE=your.local.node
STEP1 RUN TASK (PGM=DGADVITL,PARM=(C'60')) PNODE
```

The parameter, C, is the number of minutes desired in a measurement interval. The interval, 60, is an appropriate value to use. Since the Process that does the measuring runs as a PNODE=SNODE Process, the system adds a count of 2 to whatever the highest number of Processes that were running concurrently during a particular sample window. For example, if 6 other Processes were running simultaneously at a given time during the sample window, the value in the SVIT001I message will be '0008'. If no Processes were running during that sample time, the measuring Process is not counted, so the value in the SVIT001I message is '0000'.

The following example shows the output of the DGADVITL program:

```
SVIT001I SESSION HIGH WATER MARK OF: 0003 REACHED AT: 10:15:46 ON: 02000173
```

Note: Using the DGADVITL utility may not be required and Simultaneous Session Reporting (SSR) is the preferred application in most instances. Connect:Direct utilizes Simultaneous Session Reporting (SSR) to continually monitor sessions over the life of Connect:Direct and tracks the session high-water mark, generates an SMF133 record to document this value, and provides reports from the data. For CDPLEX, a DGADVITL Run Task must be running on each server in the CDPLEX to report that server's activity on the number of concurrent sessions reached during the specified period. For additional information about Simultaneous Session Reporting (SSR), see *Simultaneous Session Reporting* in the *System Administration Guide*.

ADRDSSU Interface Program (DGADSIOX I/O Exit)

The DGADSIOX I/O exit dynamically invokes the IBM ADRDSSU utility to enable you to copy:

- SMS-COMPRESSED data without first uncompressing the data
- Wildcard-named files

Running Traces for the DMDSSIOX I/O Exit

The DGADSIOX I/O exit automatically logs information to //ADRIOLG each time the ADRDSSU program is invoked regardless of the DEBUG bit settings. However, to log additional tracing information, use the setting, DEBUG=00008000 or DEBUG=04008000 in the DEBUG DDNames in the IBM Connect:Direct startup JCL to generate an I/O Buffer Trace or an I/O Buffer Trace combined with a separate trace per task. DGADSIOX I/O exit information logs to Rnnnnnnn while DSS exit information logs to Xnnnnnnn.

DGADSIIOX I/O Exit Sample Processes

Three sample Processes are provided in the \$CD.SDGAPROC library distributed with IBM Connect:Direct that you can use to prepare a Process to execute the DGADSIIOX I/O exit:

- DGAPIOX1, which copies the SMS-COMPRESSED data set, KSTIC1.SMSTEST.DMDSSU.COMP.OUTLIST, from the PNODE to a data set of the same name on the SNODE but renames the SNODE data set to MWATL1.SMSTEST.DMDSSU3.** as part of the execution of the ADRDSSU utility on the SNODE.
- DGAPIOX2, which is similar to the DGAPIOX1 process but uses symbolic substitution to name the input data set.
- DGAPWILD, which is set up to copy wildcard-named files

DGADSIIOX I/O exit limitations and security considerations

Checkpoint/restart is not supported when using DGADSIIOX I/O exit for the following reasons:

- I/O exits in general including do not support Checkpoint/restart
- The IBM ADRDSSU utility does not support Checkpoint/restart
- Control cards are allow for wild-carded input

In addition, the ADRDSSU utility does not support DUMP/RESTORE of z/OS UNIX System Services user files such as PATH='/u/kstic1/abc.txt'. However, ADRDSSU does support HFS files.

The DGADSIIOX I/O Exit utility program invokes DFSMSdss to perform requested functions by calling the ADRDSSU utility via the cross-memory application interface, documented in DFSMSdss Storage Administration. This cross-memory interface causes a DFSMSdss server address space to start under the ASPACE name of DGADSIIOX.

For HFS and ZFS type files, there are restrictions related to UNIX System Services that may result in failures or hangs without proper security. Please refer to DFSMSdss Storage Administration (Application programming interface, Cross-memory application interface restrictions) for a description of the restrictions and set up requirements related to HFS and ZFS files.

DGASCONV – Secure Parameter File Conversion Utility

When upgrading from a release of IBM Connect:Direct prior to 5.2 and using Secure Plus, the Secure Parameter and Secure Access files must be converted to the new file format. The SPAdmin tool cannot open a secure parameter file created in a release prior to 5.2.

The DGASCONV utility performs the conversion by allocating a new secure parameter file and using the old secure parameter file as input. Due to the possibility of secure passwords with Strong Password Encryption (SPE) being enabled in the previous release, this utility also uses the DGADTQFX and IDCAMS to convert the TCQ/TCX, copy the NETMAP and copy the AUTH file.

Sample JCL, DGAJCONV, in the SDGAJCL data set should be tailored to your environment and executed to perform this conversion. Use the instructions within the DGAJCONV JCL to assist with this JCL tailoring.

The execution of DGAJCONV will produce a report in the output DD called REPORT. Use this report to determine the scope of the changes made by the utility and to make recommended manual updates to any action produced in the report.

You must also specify the new secure parameter file as SECURE.DSN within your initialization parameters. This utility produces a diagnostic trace to the DD called TRACEO, which is a comment in the sample JCL. If you do not want to execute the diagnostic trace, leave the DD as a comment.

Using the TCQ/TCX Repair Utility (DGADTQFX)

The TCQ/TCX Repair Utility (DGADTQFX) can be used to help solve corruption problems related to the TCQ/TCX data sets.

Run the DGADTQFX utility in one of the following ways:

- Rebuild TCX mode, which creates a new TCX by using the current TCQ to indicate the existence of Processes.
- Use TCX mode, which creates a new TCQ using the current TCX to indicate the existence of Processes. When upgrading from Connect:Direct releases prior to 4.4, using TCX mode is recommended.

Note: Use different names to distinguish the original and new TCQ/TCX data sets in case you need to go back and reuse the original data sets.

The Rebuild TCX mode is the preferred mode for rebuilding the TCQ/TCX after encountering TCQ corruption problems, which cause U3083 abends.



CAUTION:

The DGADTQFX utility is rebuilding the TCX based on Processes that remain in the TCQ. The utility will determine the highest Process number in the TCQ and will set the next available Process number to the next number. It is possible for this utility to reuse Process numbers.

The Use TCX mode is recommended for TCQs associated with systems running versions of IBM Connect:Direct prior to Version 4.4. Prior to Version 4.4, completed Processes were retained in the TCQ and were not deleted.

The DGADTQFX program, located in \$CD.SDGALINK, has one execution parameter for specifying the report type.

Parameter	Description
PARM= SUMMARY DETAIL BACKLEVEL REMOVEPR	Specify SUMMARY to produce a report at the Process level. Specify DETAIL to produce a report, which shows steps within each Process, such as RUN TASK, RUN JOB, SUBMIT, and COPY. Specify BACKLEVEL to create a TCQ and TCX for pre-version 4.6 Connect:Direct systems. Specify REMOVEPR to remove all Processes on the PR queue.

Normally, you run DGADTQFX with IBM Connect:Direct shut down but you could run it in production. The data sets and reports created will be correct as long as no update activity to the input TCQ takes place while the utility executes. The program issues a warning message if the VSAM timestamp for the input TCQ is changed during execution.

The return codes associated with the DGADTQFX utility are described in the following table.

Return Code	Meaning
0	No errors were found in the input TCQ
4	At least one error was found and removed or a warning message was issued
8	A severe error occurred during execution and the utility was terminated

Initializing the DTF After DGADTQFX Has Found Errors

If errors were found and corrected when you ran the DGADTQFX utility, replace the original corrupted data sets in use with the new data sets created by DGADTQFX. To allocate the new data sets to IBM Connect:Direct, use IDCAMS ALTER or regenerate the network map

IBM Sterling Connect:Direct FTP+ for z/OS (CDFTP)

IBM Sterling Connect:Direct FTP+ for z/OS is a component of Connect:Direct for z/OS that provides a simple and dynamic migration path from unmanaged FTP sessions within z/OS job streams to a managed environment using Connect:Direct for z/OS with minimal change to the JCL.

IBM Sterling Connect:Direct FTP+ for z/OS provides a simple, reliable, and secure way to transfer files between a Connect:Direct for z/OS installation at a central processing center and remote IBM Connect:Direct sites. IBM Sterling Connect:Direct FTP+ for z/OS runs as the client and supports Connect:Direct for z/OS, Connect:Direct for Microsoft Windows, and Connect:Direct for UNIX as the remote system.

Components of IBM Sterling Connect:Direct FTP+

The major components of IBM Sterling Connect:Direct FTP+ are:

- IBM Sterling Connect:Direct FTP+ Manager—Starts as a job stream or started task just like a IBM Connect:Direct/Plex Manager. It handles IBM Sterling Connect:Direct FTP+ Clients and is only a PNODE. It cannot be contacted by a remote IBM Connect:Direct to start a process, and it cannot be contacted by a non- IBM Sterling Connect:Direct FTP+ Client. (The sample JCL, DGAXFTPJ, is in \$CD.SDGASAMP.)

The IBM Sterling Connect:Direct FTP+ Manager enables the TCP API and can be contacted by the ISPF UII, Control Center, the Connect:Direct Browser User Interface, and the IBM Connect:Direct JAI.

- IBM Sterling Connect:Direct FTP+ Client—Starts a special case of a IBM Connect:Direct Server. It builds and submits a Process which is scheduled by the IBM Sterling Connect:Direct FTP+ Manager back to the IBM Sterling Connect:Direct FTP+ Client using the server name for the PLEXCLASS. It is only a PNODE and cannot be contacted by a remote IBM Connect:Direct to start a Process. (The sample JCL, DGAXFCLI, is in \$CD.SDGASAMP.)
- IBM Sterling Connect:Direct FTP+ Plug-in—Provides support for FTP commands that IBM Connect:Direct does not support, such as CWD (change working directory), MKDIR (make a directory), RMDIR (remove directory), DIR (obtain a directory listing of files), and DELETE and MDELETE (deleting files).

When the JCL is changed to use IBM Sterling Connect:Direct FTP+ for z/OS instead of FTP, all of the supported FTP commands are interpreted into appropriate COPY or RUNTASK and COPY steps and executed within the IBM Sterling Connect:Direct FTP+ Client started in the user's address space.

Security Considerations for IBM Sterling Connect:Direct FTP+

The IBM Connect:Direct files, Connect:Direct Secure Plus parameter file and Netmap, are accessed through the IBM Sterling Connect:Direct FTP+ Manager under its authority. The only IBM Connect:Direct file that is accessed through the IBM Sterling Connect:Direct FTP+ Client is the IBM Connect:Direct Message File. All file accesses on the local system will be performed under the authority of the job userid (normal MVS operation). All file accesses on the remote system will be performed under the authority of the userid/password used for the SNODEID in the IBM Sterling Connect:Direct FTP+ for z/OS Process.

To use the IBM Sterling Connect:Direct FTP+ Client to connect to a remote Connect:Direct for z/OS system, the remote userid must have HFS file access on the remote Connect:Direct for z/OS system if the temporary file written by the plug-in on Connect:Direct for z/OS is an HFS file or if HFS files or directories are to be accessed.

Configure IBM Sterling Connect:Direct FTP+ for z/OS

Before you can use IBM Sterling Connect:Direct FTP+, certain installation and configuration tasks must be performed.

Complete the following tasks:

- [“Install IBM Sterling Connect:Direct FTP+ Plug-in” on page 652](#)
- [“Create CDFTP Procedure” on page 652](#)
- [“Create JCL to Start the IBM Sterling Connect:Direct FTP+ Manager” on page 654](#)
- [“Update Netmap for Remote Nodes” on page 655](#)
- [“Start IBM Sterling Connect:Direct FTP+ Manager” on page 655](#)

Install IBM Sterling Connect:Direct FTP+ Plug-in

The IBM Sterling Connect:Direct FTP+ Plug-in must be installed on the IBM Connect:Direct nodes that IBM Sterling Connect:Direct FTP+ for z/OS communicates with. The IBM Sterling Connect:Direct FTP+ Plug-in is already included in the Connect:Direct for z/OS installation. If the remote node is a Connect:Direct for UNIX or Connect:Direct for Microsoft Windows server, the IBM Sterling Connect:Direct FTP+ Plug-in must be deployed on the remote system. For IBM Sterling Connect:Direct FTP+ Plug-in installation instructions for Connect:Direct for Microsoft Windows and Connect:Direct for UNIX, refer to the latest IBM Sterling Connect:Direct FTP+ 1.2 Release Notes.

Note: Make a note of the fully-qualified installation directory of the plug-in (unless the remote node is Connect:Direct for z/OS) and the fully qualified name of the plug-in temporary file on the remote IBM Connect:Direct server. You need this information when you update the Netmap.

Create CDFTP Procedure

Build the CDFTP procedure and save it in your system's PROCLIB. The CDFTP procedure is executed in your modified FTP JCL. The sample JCL, DGAXFTPP, is in \$CD.SDGASAMP.

The following is an example of a CDFTP procedure:

```
//CDFTP      PROC
//CDFTP      EXEC PGM=DMINIT,REGION=0M,
//  PARM='USER1.INIT.PARMLIB(GLOBINIT),CDFTP'
//STDENV DD DUMMY
//SYSTCPT DD DISP=(,DELETE),DSN=*&RES,
//          SPACE=(CYL,(1,1)),DCB=(DSORG=PS,RECFM=FB,
//          LRECL=80,BLKSIZE=24000),UNIT=SYSDA
//NETRC DD DISP=SHR,DSN=USER1.OS390.SRCLIB(NETRC)
//SYSTCPD DD DISP=SHR,DSN=SYS2.TCPIP.PARMS(TCPDATA2)
//STEPLIB DD DISP=SHR,DSN=USER1.CD5000.LOADLIB
//CDPLEX DD DISP=SHR,DSN=USER1.INIT.PARMLIB(SERVER)
//CDESTAE DD SYSOUT=*
//ESTAE DD SYSOUT=*
//NDMLLOG DD SYSOUT=*
```

In this example, the Global initialization parameters file, **GLOBINIT**, and local initial initialization parameters file, **SERVER**, contain normal initialization parameters for starting a IBM Connect:Direct/Plex Server. If **CDPLEX.SERVER=** is specified in the local initialization parameters, it is ignored and the JES jobname is used as the server name. The specification of **CDFTP** as the first override parm after the specification of the location of the initialization parameters file identifies this DTF as a IBM Sterling Connect:Direct FTP+ Client.

The SYSTCPT DD is required so that the Resolver will write its trace information into this file when a request to the Resolver, such as GETHOSTNAME, is issued. IBM Sterling Connect:Direct FTP+ reads this trace information to get the resolved DATASETPREFIX from the Resolver to determine which FTP.DATA file to open and process.

A z/OS FTP client gets its configuration data from the FTP.DATA file and uses the following search order to find this file:

- DD card - //SYSFTPD
- Tsoprefix.FTP.DATA
- Tsoid.FTP.DATA
- /etc/ftp.data
- SYS1.TCPPARMS(FTPDATA)
- tcpip_hlq.FTP.DATA

IBM Sterling Connect:Direct FTP+ Client will attempt to find the correct FTP.DATA file using the following search order:

- DD card - //SYSFTPD
- JobUserid.FTP.DATA

- /etc/ftp.data
- SYS1.TCPPARMS(FTPDATA)
- tcpip_hlq.FTP.DATA

Note: /etc/ftp.data must not be File Format NA or it will be ignored.

Note: tcpip_hlq is determined from DATASETPREFIX in the RESOLVER trace information written in the SYSTCPT DD. If DATASETPREFIX is not found, the default is "TCPIP".

If the FTP.DATA file is found, the following configuration parameters are honored by IBM Sterling Connect:Direct FTP+:

- PRIMARY—used for CDFTP.TEMPFILE allocation
- SECONDARY—used for CDFTP.TEMPFILE allocation
- SPACETYPE—used for CDFTP.TEMPFILE allocation
- KEYRING—used to override Connect:Direct Secure Plus keyring

Note: The FTP.DATA file is used by both z/OS FTP and IBM Sterling Connect:Direct FTP+.

Use ;CDFTPKEYRING if IBM Sterling Connect:Direct FTP+ should use a different keyring than z/OS FTP.

- ;CDFTPKEYRING ;—used for a IBM Sterling Connect:Direct FTP+ specific override of Connect:Direct Secure Plus Keyring

Note: Do not forget to put a semi-colon (;) in front of CDFTPKEYRING and a space followed by a semi-colon (;) at the end of CDFTPKEYRING, which makes this statement a comment to z/OS FTP, but will still be recognized by IBM Sterling Connect:Direct FTP+ and processed.

CDFTPKEYRING ;(or KEYRING if ;CDFTPKEYRING ; is not specified) will override the Connect:Direct Secure Plus specification of the keyring only if the Connect:Direct Secure Plus Parmfile was defined with a keyring. If the Connect:Direct Secure Plus Parmfile is defined to use a key database, the FTP.DATA specification is ignored with a message to OUTPUT. To explicitly specify to use the Connect:Direct Secure Plus Parmfile definitions (keyring or key database, and to avoid the message to OUTPUT), code: ;CDFTPKEYRING ;.

- SECURE_DATACONN—used as default for Connect:Direct Secure Plus transfers

The IBM Sterling Connect:Direct FTP+ Client is a special case of a IBM Connect:Direct/Plex Server in a IBM Connect:Direct/Plex environment. The local initialization parameters file, **SERVER**, should specify:

```
CDPLEX.MANAGER=NO
```

For more information about global and local initialization parameters, see *Initialization Parameters* in the *IBM Connect:Direct for z/OS Administration Guide*.

Create JCL to Start the IBM Sterling Connect:Direct FTP+ Manager

Create JCL to start the IBM Sterling Connect:Direct FTP+ Manager. Refer to the following JCL as an example:

```
//CD$MGR JOB (CDHOSTD),'FTPUSER',PRTY=12,TIME=1440,CLASS=Q,
// REGION=0M,MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=&SYSUID
//*JOBPARM LINES=999999
//*****
//*
//* CONNECT:DIRECT (Connect:Direct for z/OS MANAGER) *
//*
//*****
// SET PGM=DMINIT,P=,
// E='STAT.INIT=COLD,TCQ=COLD'
//NDMITST EXEC PGM=&PGM,REGION=0M,
// PARM='&P.USER1.INIT.PARMLIB(GLOBINIT),CDFTP,&E '
//CDPLEX DD DISP=SHR,DSN=USER1.INIT.PARMLIB(CDFTPMGR)
//CDSVCDMP DD DUMMY
//STEPLIB DD DSN=USER1.CD5000.LOADLIB,DISP=SHR
//DMPUBLIB DD DSN=USER1.PROCESS.LIB,DISP=SHR
//ESTAE DD SYSOUT=*
//CDESTAE DD SYSOUT=*
//NDMLOG DD SYSOUT=*
```

In this example, the JCL that starts the IBM Connect:Direct Manager is similar to JCL that starts the IBM Connect:Direct/Plex Manager. The initparm override PARM **CDFTP** as the first override parm after the location of the global initialization parameters makes this a IBM Sterling Connect:Direct FTP+ Manager.

The following local and global initialization parameters are forced with no message:

- TCP=OES
- SNA=NO
- CTCA=NO
- UDT=NO
- SECURITY=OFF (IBM Sterling Connect:Direct FTP+ Client only. The Security parameter is allowed for the IBM Sterling Connect:Direct FTP+ Manager)
- TCQ=COLD
- CONFIRM.COLD.START=NO
- TCP.FMH.TIMER=00:00:00
- TCP.RUNTASK.TIMER=00:00:00
- PROCESS.RETENTION=NO

If you specify any of the following Global Initialization Parameters, they are ignored:

- TCP.LISTEN
- UDT33.LISTEN
- MAXPROCESS
- MAXPRIMARY
- MAXSECONDARY

The local initialization parameters file for the IBM Sterling Connect:Direct FTP+ Manager, **CDFTPMGR**, should specify:

```
CDPLEX.MANAGER=YES
```

For more information about global and local initialization parameters, see *Initialization Parameters* in the *IBM Connect:Direct for z/OS Administration Guide*.

Update Netmap for Remote Nodes

IBM Sterling Connect:Direct FTP+ for z/OS gets its configuration information about remote systems it connects to from the IBM Connect:Direct Netmap.

Configure the following parameters in your Connect:Direct for z/OS Netmap for each remote system that IBM Sterling Connect:Direct FTP+ for z/OS connects to:

- **CDFTP.PLUGIN**=<fully qualified installation directory of the IBM Sterling Connect:Direct FTP+ Plug-in>
This parameter is not required when the Netmap entry defines a z/OS system.
- **CDFTP.TEMPFILE**=<fully qualified name of the IBM Sterling Connect:Direct FTP+ temporary file>
The IBM Sterling Connect:Direct FTP+ Plug-in creates a temporary file to store the results of a directory command RUNTASK operation.

Note: If no variables are used in the **CDFTP.TEMPFILE**, IBM Sterling Connect:Direct FTP+ for z/OS appends the job userid and the ASID and CPUID to the **CDFTP.TEMPFILE** parameter as a further specification of uniqueness.

Several variables are available for substitution in the **CDFTP.TEMPFILE** specification. The following variables allow you to build an appropriate unique temporary file as a z/OS file:

- **&userid**;—The remote userid (lower case)
- **&USERID**;—The remote userid (upper case)
- **&luserid**;—The local userid padded to eight characters with '\$'
- **&lcpuid**;—The first eight bytes of the local CPUID with "C" overlaying the first byte
- **&asid**;—The four character Address Space ID prefixed with "AS"

For example, you can specify **CDFTP.TEMPFILE**= with a variable ("**&userid**"). IBM Sterling Connect:Direct FTP+ for z/OS inserts the remote userid in place of the **&userid**. In the following example, the userid specified in the script or NETRC file is **user1**.

```
CDFTP.TEMPFILE=/u/&userid;/cdftp/temp.file
```

CDFTP.TEMPFILE resolves to:

```
CDFTP.TEMPFILE=/u/user1/cdftp/temp.file
```

In the script or NETRC file, specify the userid in the case expected for the substitution.

Start IBM Sterling Connect:Direct FTP+ Manager

Start the IBM Sterling Connect:Direct FTP+ Manager as a job stream or started task the same way you start IBM Connect:Direct/Plex Manager using the JCL created in the previous step.

The IBM Sterling Connect:Direct FTP+ Manager must be running before the IBM Connect:Direct FTP+ Client can successfully contact the IBM Sterling Connect:Direct FTP+ Manager to perform its work.

For more information, see the *IBM Connect:Direct for z/OS Administration Guide*.

Modify FTP JCL

Use the following procedure to modify your FTP JCL to use IBM Sterling Connect:Direct FTP+ :

Procedure

1. Change **EXEC PGM=FTP** to **EXEC PROC=CDFTP**.
2. Change the INPUT and OUTPUT DD statements and other step statements to specify the procedure step name. For example, change **//INPUT** to **//CDFTP.INPUT** and change **//OUTPUT** to **//CDFTP.OUTPUT**.
3. Change the first statement in the INPUT data stream to specify the remote IBM Connect:Direct node name instead of the machine name for the FTP server.

The following is an example of JCL that executes the CDFTP procedure:

```
//CD$FTPJC JOB (CDHOSTD),'-CDFTP+- ',TIME=(1440),CLASS=Q,
// REGION=0M,MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=&SYSUID
/*JOBPARM LINES=999999
//LIBS JCLLIB ORDER=(USER1.OS390.SRCLIB)
//CDFTP EXEC PROC=CDFTP
//CDFTP.NETRC DD DISP=SHR,DSN=USER1.OS390.SRCLIB(NETRC)
//CDFTP.FTPTRACE DD SYSOUT=*
//CDFTP.OUTPUT DD SYSOUT=*
//CDFTP.INPUT DD *
MVSBB (exit
ebcdic
cd /u/user3
delimit
mkdir cdftptestlib
cd cdftptestlib
lcd 'USER1.CDFTP.TESTLIB'
delimit
bin
site sbdataconn=(IBM-037,IS08859-1)
locsite sbdataconn=(IBM-037,IS08859-1)
put CDFTPJC1
ebcdic
put CDFTPJC1 CDFTPJC2
dir
ls
mdelete CDFTPJC*
cdup
rmdir cdftptestlib
close
quit
//
```

You must provide the following information in the JCL:

- Remote host—In this example, the hostname of the FTP server is changed to specify the IBM Connect:Direct node name, **MVSBB**, that is in the Netmap. If the DNS name or IP address used by FTP is the nodename in the Netmap, no change is required. However, duplicating the entry in the Netmap and giving it the IP address or the DNS name as the node name may cause Netmap checking problems. In this case, the script should be changed to use the Netmap entry node name.

The remote host statement can also specify **(exit**, as shown in this example, or **(exit=nn** to specify that any error exits the script with a reason code (default is RC=8).

The remote host statement can also specify **(secure=off|ssl|tls** to specify security options.

If both **exit** and **secure=** are specified; only one open parenthesis can be used. For example, **MVSBB (exit 16 secure=off**.

- Remote userid and password—In this example, they are obtained from the NETRC file. If the remote userid and password are not specified in the NETRC file, they must be included in the FTP script after the remote host.

In the following sample excerpt of JCL that executes the CDFTP procedure, the remote host is **PROD.ZOS1**, the remote userid is **user1** and the password is **S1VX\$25B**:

```
...
//CDFTP.INPUT DD *
PROD.ZOS1
user1 S1VX$25B
cd /u/daily
...
```

You do not specify the use of Connect:Direct Secure Plus in the JCL. It is determined by the initialization parameters and the Connect:Direct Secure Plus parameter file definitions for the node named in the script.

To simplify matching up statistics in the remote node with IBM Sterling Connect:Direct FTP+ for z/OS, the PROCESS name is the jobname and the process number is the jobid. The jobname is also used for

the IBM Sterling Connect:Direct FTP+ Client server name when contacting the IBM Sterling Connect:Direct FTP+ Manager.

Note: Only the JCL interface is implemented. The ISPF interface (option 6;FTP) is not supported.

New \$CD.SDGAPARM and \$CD.SDGASAMP Members

The IBM Connect:Direct \$CD.SDGAPARM and \$CD.SDGASAMP contain Initialization Parameters, JCL, and other samples that you can use and modify in your IBM Sterling Connect:Direct FTP+ environment.

Member	Description	Location
DGAIPGBL	Global Initialization Parameters	\$CD.SDGAPARM
DGAIPMGR	Manager Local Parameters	\$CD.SDGAPARM
DGAIPSVR	Server (FTP+ Client) Local Parameters	\$CD.SDGAPARM
DGAXFTPJ	CD FTP+ Manager Startup JCL	\$CD.SDGASAMP
DGAXFCLI	CD FTP+ Client JCL	\$CD.SDGASAMP
DGAXFTPP	CDFTP PROC	\$CD.SDGASAMP
DGAXNETC	Sample NETRC	\$CD.SDGASAMP
DGAXFTPM	Sample NETMAP Entry	\$CD.SDGASAMP

Supported FTP Commands

IBM Sterling Connect:Direct FTP+ for z/OS separates the z/OS FTP commands into three categories:

- Commands completely supported
- Commands partially supported
- Commands accepted, but ignored.

All commands are case insensitive and can be abbreviated as specified in the z/OS FTP documentation.

Fully Supported FTP Commands

The following FTP commands are fully supported by IBM Connect:Direct FTP+ for z/OS as described in the following:

Command	Description
APpend	Appends a data set on the local host to a file on the foreign host
BINary	Sets the transfer type to IMAGE
CD	Changes the working directory
CDUp	Changes to the parent of the current working directory
CLEar	Sets the protection level for data transfers to CLEAR
CLOSE	Disconnects from the foreign host
COMpress	Sets the data transfer mode to compressed mode
CWd	Changes the working directory (Synonymous with CD)
DELEte	Deletes a single file on the foreign host
DELImit	Displays the delimiter character between the file_name and file_type
DIr	Lists the directory entries for files on the foreign host Note: Format is from Connect:Direct for z/OS Plug-in

Command	Description
EBcdic	Sets the transfer type to EBCDIC
Get	Copies a file from the foreign host to your local host Note: "(REPLACE" supported
GLob	Toggles globbing (the expansion of metacharacters in file names) for the MDelete, MGet, and MPut subcommands
LCd	Changes the working directory on the local host
LPwd	Displays the name of the active working directory on the local host
LS	Lists the names of files on the foreign host
MDelete	Deletes multiple files on the foreign host
MGet	Copies multiple files from the foreign host to the local host Note: "(REPLACE" supported
MPut	Copies multiple files on the local host to the foreign host
PRIVate	Sets the protection level for data transfers to PRIVATE
PROtect CLEAR	PROTECT sets the protection level for data transfers on data connections. CLEAR sets SECURE=(ENC=N) in COPY steps.
PROtect PRIVATE	PROTECT sets the protection level for data transfers on data connections. PRIVATE sets SECURE=(ENC=Y) in COPY steps.
PUt	Copies a file on the local host to the foreign host
PWd	Displays the name of the active working directory on the foreign host
QUIt	Leaves the FTP command environment
REName	Renames a file on the foreign host
RMdir	Removes a directory
SENDSite	Enables or disables automatic transmission of the SItE subcommand Note: Used to determine whether to propagate DCB or use SITE/LOCSITE

Partially Supported FTP Commands

The following FTP commands are supported by IBM Sterling Connect:Direct FTP+ with limitations described in the Notes:

Command	Description	Note
AScii	Sets the transfer type to ASCII	Treated as EBCDIC
BIG5	Sets the transfer type to BIG5. BIG is the minimum abbreviation for BIG5.	Only to set DBCS= table in IBM Connect:Direct
HAngeul	Sets the transfer type to HANGEUL	Only to set DBCS= table in IBM Connect:Direct
JIS78kj	Sets the transfer type to JIS78KJ	Only to set DBCS= table in IBM Connect:Direct
JIS83kj	Sets the transfer type to JIS83KJ	Only to set DBCS= table in IBM Connect:Direct
Ksc5601	Sets the transfer type to KSC5601	Only to set DBCS= table in IBM Connect:Direct

Command	Description	Note
LMkdir	Create a directory on the local host	"(LIKE" ignored
LOCSite	Specifies information that is used by the local host to provide service specific to that host system	<p>Ignores all but:</p> <ul style="list-style-type: none"> • BLOCKS • BLKSIZE • BLOCKSIZE • CONDISP • CYLINDERS • DIRECTORY • LRECL • PRIMARY • RECFM • SECONDARY • TRACKS <p>Errors out on:</p> <ul style="list-style-type: none"> • TRUNCATE • WRAPRECORD
MKdir	Creates a directory on the foreign host	"(LIKE" ignored
MOfde	Specifies the mode or data format of the transfer	MODE C honored as COMPRESS. All others ignored.
Open	Opens a connection to a foreign host	Port ignored
Pass	Supplies a password to the foreign host	No :userdata or account_information supported (ignored)
PROMpt	Toggles interactive prompting for MDelete, MGet, and MPut commands	Toggles prompt switch
PROXy	Executes an FTP subcommand on a secondary control connection	Treated as OPEN statement
PROtect SAFE	Sets SECURE=(ENC=Y) in COPY step	Treated as PROTECT PRIVATE
QUOte	Sends an uninterpreted string of data	Interprets command locally
SAfe	Sets the protection level on data transfers to safe	Treated as PRIVATE. Sets SECURE=(ENC=Y) in the COPY step.
SChinese	Sets the transfer type to SCHINESE	Only to set DBCS= table in IBM Connect:Direct

Command	Description	Note
Site	Sends information to the foreign host using site-specific commands	Ignores all but: <ul style="list-style-type: none"> • BLOCKS • BLKSIZE • BLOCKSIZE • CYLINDERS • DIRECTORY • FILETYPE JES • LRECL • PDSTYPE • PRIMARY • RECFM • TRACKS Errors out on: <ul style="list-style-type: none"> • TRUNCATE • WRAPRECORD
SUnique	Toggles the storage methods	NAME/NONAME not supported Used to set DISP=NEW or DISP=RPL for output file
TChinese	Sets the transfer type to TCHINESE	Only to set DBCS= table in IBM Connect:Direct
Type	Specifies the transfer type	Ignores all but: <ul style="list-style-type: none"> • TYPE I = BINARY • TYPE A = ASCII • TYPE E = EBCDIC • TYPE B 2 = EUCKANJI • TYPE F = IBMKANJI • TYPE B 5 = HANGEUL • TYPE B 4 = JIS78KJ • TYPE B 3 = JIS83KJ • TYPE B 6 = KSC5601 • TYPE B 1 = SJISKANJI • TYPE B = SJISKANJI
User	Identifies you to a foreign host	No :userdata and account_information supported (ignored)

FTP Commands Accepted, But Ignored

The following commands are recognized, accepted, and ignored by IBM Sterling Connect:Direct FTP+. These commands do not produce errors.

FTP commands that are not supported or recognized, are ignored with appropriate messages. However, if IBM Sterling Connect:Direct FTP+ is in exit mode (**exit** or **exit=** specified in the remote host statement), these commands stop the script and terminate the step. This is how z/OS FTP behaves.

(LOC)SITE subcommands that are not supported or recognized are ignored and do not terminate the step, even in exit mode. The exceptions to this z/OS FTP behavior are the **(LOC)SITE WRAPRECORD** and **TRUNCATE** subcommands. If these subcommands are specified, they always terminate the script (**RC=8** when not in exit mode and with the user's return code when in exit mode).

- ?
- !
- ACCount
- BLock
- CCc
- CProtect
- DEBug
- DUMP
- EUCKANJI
- FEature
- FIle
- HElp
- Ibmkanji
- LANGuage
- LOCSTat
- NOop
- RECord
- REStart
- SENDPort
- SJiskanji
- SRestart
- STAtus
- STREam
- STRucture
- SYstem
- TSO
- UCs2
- Verbose

Chapter 8. Facilities Guide

IBM Connect:Direct for z/OS Facilities Overview

IBM Connect:Direct for z/OS links technologies and moves information between networked systems and computers. It manages high-performance transfers by providing:

- Automation
- Reliability
- Efficient use of resources
- Application integration
- Ease of use

Note: Examples and source are mentioned in this document. Make any modifications and customizations in a site private PDS and not in an SMP/E target PDS, unless you make them via a USERMOD.

Important: When you are using CICS API in Connect:Direct, pair SIGNONs with SIGNOFFs to avoid using memory and rendering the interface inactive.

IBM Connect:Direct offers choices in communications protocols, hardware platforms, and operating systems. It provides the flexibility to move information among mainframes, midrange systems, desktop systems, and LAN-based workstations.

The following facilities are provided to fill specific needs:

- The Activity Reporting System (ARS) works with Connect:Direct for z/OS to produce reports of IBM Connect:Direct activity using the SAS software.
- The Console Operator interface enables you to issue all Connect:Direct for z/OS commands from a z/OS console by using a MODIFY command.
- Event Services Support (ESS) generates events asynchronously in IBM Connect:Direct and can be used by external management and automated operations applications that require real-time notification of IBM Connect:Direct activities.
- The Spool Transfer facility enables you to transfer Job Entry Subsystem (JES) spool files by copying from JES spool files, to JES print queues, and to JES reader queues.

ARS Report Examples

IBM Connect:Direct for z/OS Activity Report

You can request reports in the following ways:

- Through ARS screens using TSO/ISPF
- Through a IBM Connect:Direct Process that is defined to automatically request reports
- Through a pre-defined batch job stream that is run by a job scheduling subsystem

The IBM Connect:Direct for z/OS Activity Report lists IBM Connect:Direct activity by Process step. The ARS software sorts the steps for each secondary node (SNODE) location by user ID and then by Process number, for a specified time period.

You can specify the start date and time and the stop date and time. Following is an example of the IBM Connect:Direct for z/OS Activity Report:

Connect:Direct ACTIVITY REPORT (ARS00001) 17:12 WEDNESDAY, JUNE 24, 2010 1										
NODE = CD.BOSTON										
USERID	PROC #	PROC NAME	STEP NAME	FUNCTION	ELAPSE TIME	PNODE	CC	EXC	MSG ID	DATE
ALPHA	46	COPY1	STEP01	COPY	0:00:01.89	CD.BOSTON	00000000		SCFA000I	06/17/2010
ALLEN	63	COPY2	STEP01	COPY	0:00:02.08	CD.CHICAGO	00000000		SRFA000I	06/17/2010
BETA	87	RUNJOB	LAB	RUNJOB	0:00:01.26	CD.CHICAGO	00000000		SRJA000I	06/17/2010
ETHAN	89	RUNJOB	SUBSTEP	RUNJOB	0:00:00.68	CD.JERSEY	00000000		SSJA000I	06/17/2010
LEWIS	9	RUNT	STEP1	RUNTASK	0:00:00.11	CD.JERSEY	00000000		SRTA000I	06/17/2010
LOUIS	95	SUB	SUBSTEP	SUBMIT	0:00:01.34	CD.JERSEY	00000008		SSUB000I	06/17/2010
MULLEN	96	COPY05	STEP1	COPY	0:00:01.57	CD.MIAMI	80013000		SCFA000I	06/17/2010
WARD	99	COPY05	STEP1	COPY	0:00:02.06	CD.MIAMI	80806000		SCFA000I	06/17/2010
WARREN	379	SUB	SUBSTEP	SUBMIT	0:00:00.96	CD.MIAMI	00000008	***	SSUB000I	06/17/2010
-----SNODE=CD.JERSEY-----										
JONES	544	RUNT	STEP1	RUNTASK	0:00:00.18	CD.BOSTON	80806000	***	SRTA004I	06/18/2010
-----SNODE=CD.MIAMI-----										
USERID	PROC #	PROC NAME	STEP NAME	FUNCTION	ELAPSE TIME	PNODE	CC	EXC	MSG ID	DATE
ALLEN	413	COPY05	COPY01	COPY	0:00:06.96	CD.BOSTON	00000000		SCFA000I	06/17/2010
ALLEN	414	RUNTASK	LAERNJ2	RUNJOB	0:00:01.13	CD.BOSTON	00000000		SRJA000I	06/17/2010
BURNS	415	RUNT	STEP1	RUNTASK	0:00:00.28	CD.BOSTON	00000000		SRTA000I	06/17/2010
BURNS	418	SUB	SUBSTEP	SUBMIT	0:00:01.79	CD.BOSTON	00000000		SSUB000I	06/17/2010
GEORGE	420	PDS COPY	COPY01	COPY	0:00:08.42	CD.BOSTON	00000000		SCFA000I	06/17/2010
JONES	421	PDS COPY	COPY01	COPY	0:00:08.21	CD.BOSTON	00000008	***	SDE5078I	06/17/2010
LISA	422	RUNJOB	LAERNJ2	RUNJOB	0:00:00.34	CD.BOSTON	80013000	***	SVGS010I	06/17/2010
MILLER	425	RUNT	STEP1	RUNTASK	0:00:00.18	CD.BOSTON	80806000	***	SRTA004I	06/17/2010
RYAN	426	SUB	SUBSTEP	SUBMIT	0:00:00.68	CD.BOSTON	00000008	***	SSUB005I	06/17/2010
RYAN	427	RUNJOB	LAERNJ2	RUNJOB	0:00:00.39	CD.BOSTON	80013000	***	SVSG010I	06/19/2010
RYAN	428	RUNJOB	LAERNJ2	RUNJOB	0:00:01.54	CD.BOSTON	00000000	***	SRJA000I	06/19/2010
RYAN	429	RUN10	STEP1	RUNTASK	0:00:00.25	CD.BOSTON	80806000	***	SRTA004I	06/19/2010
SMITH	430	RUN20	STEP1	RUNTASK	0:00:00.09	CD.BOSTON	00000000		SRTA000I	06/19/2010
SMITH	431	SUBACCT	SUBSTEP	SUBMIT	0:00:01.19	CD.BOSTON	00000004	***	SEFA016I	06/19/2010
TURNER	432	SUBSTAT	SUBSTEP	SUBMIT	0:00:01.59	CD.BOSTON	00000008	***	SPFG002I	06/19/2010
WARREN	433	SUBSALE	SUBSTEP	SUBMIT	0:00:02.47	CD.BOSTON	00000000	***	SSUB000I	06/19/2010

The following table contains a description of the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is being examined.
SNODE	IBM Connect:Direct secondary node.
USERID	IBM Connect:Direct user ID.
PROC #	IBM Connect:Direct Process number.
PROC NAME	IBM Connect:Direct Process name.
STEP NAME	Name (or label) assigned to the Process step.
FUNCTION	IBM Connect:Direct activity. Valid Process steps are COPY, RUN JOB, RUN TASK, and SUBMIT (within a Process).
ELAPSE TIME	Function duration time in hh:mm:ss.hh format.
PNODE	IBM Connect:Direct primary node.
CC	IBM Connect:Direct completion code in hexadecimal format.
EXC	Exception field is displayed "****" when a Process step does not complete successfully.
MSG ID	IBM Connect:Direct message ID.
DATE	When activity began. Date format is mm/dd/yyyy. Process start date does not conform to the DATEFORM initialization parameter.
CLASS	IBM Connect:Direct Session Class

IBM Connect:Direct for z/OS Summary Report

The IBM Connect:Direct for z/OS Summary Report summarizes activity for the node where the statistics file are examined. The report categorizes IBM Connect:Direct activity by SNODE for a specified time period and a summary for all SNODEs.

Information includes:

- Total Process steps run
- Total COPY steps run
- Total RUN JOB steps run, and RUN JOB steps submitted at this node
- Total RUN TASK steps run, RUN TASK steps executed at this node
- Total SUBMIT (within a Process) steps run and number of Processes submitted at this node
- Number of successful steps and unsuccessful steps
- Total bytes tested, written, sent, and received by this node
- Total elapsed time for all Process steps in this time period
- Average time to complete a Process step
- Total elapsed time for all COPY steps
- Average time to complete a COPY step
- Effective send and receive rates

You can specify the start date and time and the stop date and time.

Following is an example of the IBM Connect:Direct for z/OS Summary Report

```

-----IBM Sterling Connect:Direct for z/OS Summary Report
                                NODE = xx.xxx.xxxxxx
0  SNODE = SC.DUB.MWATL1
-  FIRST FUNCTION BEGAN EXECUTING ON    02/25/2011 AT 12:40:41.51
-  LAST FUNCTION FINISHED EXECUTING ON  02/25/2011 AT 12:44:27.90
-
0  18  PROCESS STEPS WERE COMPLETED FOR THIS TIME PERIOD.
-
-  6    TOTAL COPY STEPS OR 33.3    % OF TOTAL STEPS RUN WERE COPIES
-  1    TOTAL RUNJOB STEPS OR 5.6    % OF TOTAL STEPS RUN WERE RUNJOBS
0  1    RUNJOBS WERE SUBMITTED ON SC.DUB.MWATL1
-
-  4    TOTAL RUNTASK STEPS OR 22.2  % OF TOTAL STEPS RUN WERE RUNTASKS
0  4    RUNTASKS WERE ATTACHED ON SC.DUB.MWATL1
-
-  7    TOTAL SUBMIT STEPS OR 38.9   % OF TOTAL STEPS RUN WERE SUBMITS
0  7    SUBMITS (WITHIN A PROCESS) ON SC.DUB.MWATL1
-
0  14   SUCCESSFUL STEPS OR 77.8    % OF TOTAL STEPS RUN COMPLETED WITH A ZERO RETURN CODE
0  4    UNSUCCESSFUL STEPS OR 22.2  % OF TOTAL STEPS RUN COMPLETED WITH A NON-ZERO RETURN CODE
1
                                IBM Sterling Connect:Direct for z/OS Summary Report
                                NODE = SC.DUB.MWATL1
0  SNODE = SC.DUB.MWATL1
-
0  183,726,180  TOTAL BYTES WERE READ
0  143,315,988  TOTAL BYTES WERE SENT
0  0            TOTAL BYTES WERE WRITTEN
0  0            TOTAL BYTES WERE RECEIVED
-
0  0:00:47.82  WAS THE TOTAL "FUNCTION" TIME (SPENT PERFORMING FUNCTIONS)
-  0:00:02.66  WAS THE AVERAGE TIME TO COMPLETE A FUNCTION
-  0:00:47.38  WAS THE TOTAL "COPY" TIME (SPENT TRANSFERRING DATA)
-  0:00:07.90  WAS THE AVERAGE TIME TO COMPLETE A COPY
-
-  1514808.033 BYTES/SECOND = EFFECTIVE SEND RATE
0  0          BYTES/SECOND = EFFECTIVE RECEIVE RATES

```

The following table contains a description of the report fields.

Report Field	Description
NODE	Node where the statistics file is being examined.
SNODE	IBM Connect:Direct secondary node.
FIRST FUNCTION	Actual date/time that first Process step began executing during specified time period. Includes any step that started before specified time, but is under way or completes during the time period.
LAST FUNCTION	Actual date/time last Process step finished executing during specified time period. If a step started during specified time period but ended after requested stop time or stop date, it is not included in this report.
PROCESS STEPS COMPLETED	Total number of Process steps completed in time period.

Report Field	Description
TOTAL COPY STEPS	Total number of COPY steps run.
% TOTAL STEPS RUN WERE COPIES	Percentage of total steps run that are COPY steps.
TOTAL RUNJOB STEPS	Total number of RUN JOB steps run.
% TOTAL STEPS RUN WERE RUNJOB	Percentage of total steps run that are RUN JOB steps.
# RUNJOBS SUBMITTED	Number of RUN JOB steps submitted to run at "NODE."
TOTAL RUNTASK STEPS	Total RUN TASK steps run.
% TOTAL STEPS RUN WERE RUNTASKS	Percentage of total steps run that are RUN TASK steps.
# RUNTASKS ATTACHED	Number of RUN TASK steps that are attached on "NODE."
TOTAL SUBMIT STEPS	Total number of SUBMIT steps run.
% TOTAL STEPS RUN WERE SUBMITS	Percentage of total steps run that are SUBMIT (within a Process) steps.
# SUBMITS (WITHIN A PROCESS)	Number of SUBMIT (within a Process) steps submitted to run at "NODE."
SUCCESSFUL STEPS	Total number of successful steps.
% OF TOTAL STEPS RUN WITH ZERO RETURN CODE	Percentage of total successful steps.
UNSUCCESSFUL STEPS	Total number of unsuccessful steps.
% OF TOTAL STEPS RUN WITH NON-ZERO RET CD	Percentage of total unsuccessful steps.
TOTAL BYTES WERE READ	Total number of bytes read by "NODE."
TOTAL BYTES WERE SENT	Total number of bytes sent by "NODE."
TOTAL BYTES WERE WRITTEN	Total number of bytes written by "NODE."
TOTAL BYTES WERE RECEIVED	Total number of bytes received by "NODE."
WAS TOTAL "FUNCTION" TIME	Sum of elapsed times of all individual Process steps. The elapsed time for this calculation is the time period between when the step started and when the step completed.
WAS AVERAGE TIME TO COMPLETE A FUNCTION	Total function time divided by number of Process steps run.
WAS TOTAL "COPY" TIME	Sum of elapsed times for all individual COPY steps. The elapsed time for this calculation is the time period between when the step started and when the step completed.
WAS AVERAGE TIME TO COMPLETE A COPY	Total COPY time divided by the total number of COPY steps run.
EFFECTIVE SEND RATE	Total bytes sent divided by elapsed time. For this calculation, the elapsed time is the time between when the first COPY step in the data set began and the last COPY step in the data set completed.

Report Field	Description
EFFECTIVE RECEIVE RATE	Total bytes received divided by elapsed time. For this calculation, the elapsed time is the time between when the first COPY step in the data set began and the last COPY step in the data set completed.

IBM Connect:Direct Exception Report

The IBM Connect:Direct Exception Report lists Process steps that did not complete successfully. The ARS software sorts Process steps by SNODE, by user ID, and by Process number. The report displays information unique to the executed Process step. For example, information displayed for a failed COPY step differs from that of a failed RUN TASK step. A summary of exception cases is listed at the end of the report.

-----				IBM Sterling Connect:Direct for z/OS Exception Report	
- USERID				NODE = SC.DUB.MRATL1	
*****				EXCEPTION INFORMATION	
- MRATL1				*****	
PROC #				** NON-PDS COPY **	
PROC NAME				02/25/2011 PNODE = SC.DUB.MRATL1	
STEP NAME				SNODE = SC.DUB.MRATL1	
				COMPLETION CODE = 0000000C	
				MSG ID = SCFA046F	
				SENDING NODE ==> SC.DUB.MRATL1	
				SRC DSNNAME ==> DEVSTG2.HOST.DMS001.TSTFILE	
				RECEIVING NODE ==> SC.DUB.MRATL1	
				DEST DSNNAME ==> MRATL1.SMSTEST.HOST.DMS001.A9A0669	
				PROCESS HAS BEEN SUSPENDED BY USER/OPERATOR REQUEST	

MRATL1				** NON-PDS COPY **	
2				02/25/2011 PNODE = SC.DUB.MRATL1	
OS9901				SNODE = SC.DUB.MRATL1	
STEP01				COMPLETION CODE = 0000000C	
				MSG ID = SCFA046F	
				SENDING NODE ==> SC.DUB.MRATL1	
				SRC DSNNAME ==> DEVSTG2.HOST.DMS001.TSTFILE	
				RECEIVING NODE ==> SC.DUB.MRATL1	
				DEST DSNNAME ==> MRATL1.SMSTEST.HOST.DMS001.A9A0669	
				PROCESS HAS BEEN SUSPENDED BY USER/OPERATOR REQUEST	

MRATL1				** PDS COPY **	
4				02/25/2011 PNODE = SC.DUB.MRATL1	
PDS				SNODE = SC.DUB.MRATL1	
STEP01				COMPLETION CODE = 00000008	
				MSG ID = SVS6016F	
				SENDING NODE ==> SC.DUB.MRATL1	
				SRC DSNNAME ==> MRATL1.JCL.LIB	
				RECEIVING NODE ==> SC.DUB.MRATL1	
				DEST DSNNAME ==> MRATL1.JCL.LIB	
				PDS ERROR - DIRECTORY FULL/NON-EXISTENT MEMBER/WRONG DBCRG	

MRATL1				** PDS COPY **	
4				02/25/2011 PNODE = SC.DUB.MRATL1	
PDS				SNODE = SC.DUB.MRATL1	
STEP01				COMPLETION CODE = 00000004	
				MSG ID = SCFA000F	
				SENDING NODE ==> SC.DUB.MRATL1	
				SRC DSNNAME ==> MRATL1.JCL.LIB	
				RECEIVING NODE ==> SC.DUB.MRATL1	
				DEST DSNNAME ==> MRATL1.JCL.LIB	
				COPY STEP SUCCESSFUL	

1				IBM Sterling Connect:Direct for z/OS Exception Report	
				NODE = SC.DUB.MRATL1	
				SUMMARY INFORMATION	

				-#	
				NON-SUCCESSFUL STEPS IN 18 PROCESS STEPS RUN OR	
				0--> 22.2 % OF TOTAL PROCESS STEPS RUN COMPLETED WITH A NON-ZERO RETURN CODE	
				-#	
				NON-SUCCESSFUL COPIES IN 6 COPY STEPS RUN OR	
				0--> 66.7 % OF TOTAL COPY STEPS RUN COMPLETED WITH A NON-ZERO RETURN CODE	
				-#	
				NON-SUCCESSFUL RUNJOBS IN 1 RUNJOB STEPS RUN OR	
				0--> 0 % OF TOTAL RUNJOB STEPS RUN COMPLETED WITH A NON-ZERO RETURN CODE	
				-#	
				NON-SUCCESSFUL RUNTASKS IN 4 RUNTASK STEPS RUN OR	
				0--> 0 % OF TOTAL RUNTASK STEPS RUN COMPLETED WITH A NON-ZERO RETURN CODE	
				-#	
				NON-SUCCESSFUL STEPS IN 4	
				-#	

You specify the start date and time and stop date and time. Following are the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is being examined.
USERID	IBM Connect:Direct user ID.
PROC #	IBM Connect:Direct Process number.
PROC NAME	IBM Connect:Direct Process name.
STEP NAME	Name (or label) assigned to the Process step.
function type	IBM Connect:Direct activity. Valid Process steps are PDS COPY, NONPDS COPY, RUN JOB, RUN TASK, and SUBMIT (within a Process).
mm/dd/yyyy	Data Process step began executing.
PNODE	IBM Connect:Direct primary node.
SNODE	IBM Connect:Direct secondary node.
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format.

Report Field	Description
MSG ID	IBM Connect:Direct message ID.
SENDING NODE	(COPY) Sending IBM Connect:Direct node location name.
SRC DSNAME	(COPY) Source data set name.
RECEIVING NODE	(COPY) Receiving IBM Connect:Direct node location name.
DEST DSNAME	(COPY) Destination data set name.
DATASET CONTAINING JOB	(RUNJOB) Data set containing job.
INTENDED NODE FOR JOB SUBMISSION	(RUNJOB) Intended node location where the job is submitted.
PROGRAM NAME	(RUNTASK) Program name.
INTENDED NODE FOR PROGRAM EXECUTION	(RUNTASK) Intended attach node where the program is run.
DATASET CONTAINING PROCESS	(SUBMIT) Data set containing Process.
INTENDED NODE FOR PROCESS SUBMISSION	(SUBMIT) Intended node location where the Process is submitted.
message	Error message for IBM Connect:Direct message ID.
NON-SUCCESSFUL STEPS	Total number of unsuccessful Process steps.
PROCESS STEPS RUN	Total number of Process steps run.
% TOTAL PROCESS STEPS WITH NON-ZERO CODE	Percentage of Process steps unsuccessful.
NON-SUCCESSFUL COPIES	Number of unsuccessful COPY steps.
COPY STEPS RUN	Total number of COPY steps run.
% TOTAL RUN JOB STEPS WITH NON-ZERO CODE	Percentage of RUN JOB steps unsuccessful.
NON-SUCCESSFUL RUNTASKS	Number of unsuccessful RUN TASK steps.
RUNTASK STEPS RUN	Total number of RUN TASK steps run.
% TOTAL RUNTASK STEPS WITH NON-ZERO CODE	Percentage of RUN TASK steps unsuccessful.
NON-SUCCESSFUL SUBMITS	Number of unsuccessful SUBMIT (within a Process) steps.
SUBMIT STEPS RUN	Total number of SUBMIT (within a Process) steps run.
% TOTAL SUBMIT STEPS WITH NON-ZERO CODE	Percentage of SUBMIT (within a Process) steps unsuccessful.

Security Violations Report

The IBM Connect:Direct for z/OS Security Violations Report lists the following types of violations for a specified time period:

Signon security failures—Failure caused by an invalid user ID or password that signed on to IBM Connect:Direct. Each violation is recorded.

Process security failures—This failure is caused when a Process does not run due to an invalid security authorization. An example is a Process that does not run because the user is not defined in the IBM Connect:Direct authorization file at the remote node.

Data set access security failures—Failure due to insufficient authority to access a data set.

A sample report follows:

1	IBM Connect:Direct Security Violations Report					2
	NODE = SC.DUB.MWATL1		13:35 Friday, February 25, 2011			
	VIOLATION					
	USERID	CC	MSG ID	DATE	TIME	TYPE
	MyCCente	00000008	RACF001I	02/25/2011	12:21:12.5	SIGNON
	MWATL1	00000008	RACF097I	02/25/2011	12:23:04.4	SIGNON

You can specify the start date and time and the stop date and time.

Included in these reports are security message IDs generated by any security subsystem used with IBM Connect:Direct. Security subsystems supported include:

- IBM Resource Access Control Facility (RACF)
- CA-ACF2 and CA-TOP SECRET by Computer Associates, Inc.

The following table contains a description of the report fields:

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is examined.
USERID	IBM Connect:Direct user ID that created the security violation.
CC	IBM Connect:Direct completion code in hexadecimal format.
MSG ID	Security system message ID: IBM Connect:Direct, ACF2, RACF, or TOP SECRET.
DATE	Date security violation occurred.
TIME	Time security violation occurred.
VIOLATION TYPE	Type of security violation. Valid types are SIGNON, PROCESS, and DATASET.

Connect:Direct for z/OS Function Reports

The Function Reports provide detailed information about specific Process steps for a specified time period. The following reports are included:

- IBM Connect:Direct for z/OS Non-PDS Copy Report
- IBM Connect:Direct PDS Copy Report
- IBM Connect:Direct Run Job Report
- IBM Connect:Direct Run Task Report
- IBM Connect:Direct for z/OS Submit Within a Process Report

You can request the start date and time and the stop date and time when requesting one of the Connect:Direct for z/OS Function Reports.

IBM Connect:Direct for z/OS Non-PDS Copy Report

The IBM Connect:Direct for z/OS Non-PDS Copy Report provides information about COPY steps involving these transfers:

- Non-PDS data set <-----> Non-PDS data set
- Non-PDS data set <-----> PDS data set member
- PDS data set member <-----> Non-PDS data set

Following is an example of the IBM Connect:Direct for z/OS Non-PDS Copy Report.

				IBM Sterling Connect:Direct for z/OS Non-PDS Copy Report	
USERID	PROC #	PROC NAME	STEP NAME	TRANSMISSION INFORMATION	
*****	*****	*****	*****	*****	
MMWATL1	1	Q83903	PUSH01	02/25/2011 PNODE = SC.DUB.MWATL1	SNODE = SC.DUB.MWATL1
				TRANSMISSION TIME = 0:00:00.04	COMPLETION CODE = 00000000 MSG ID = SCFA0001
				COPY STEP SUCCESSFUL.	

				SRC DSNAME IS MMWATL1.TCPIF.DATA.FILE	
				DEST DSNAME IS MMWATL1.FTST.A7BE04E	
				SENDING NODE ==> SC.DUB.MWATL1	RECEIVING NODE ==> SC.DUB.MWATL1
				BYTES READ ==> 266	BYTES WRITTEN ==> 266
				BLOCKS READ ==> 1	BLOCKS WRITTEN ==> 1
				RECS READ ==> 0	RECS WRITTEN ==> 0
				BYTES SENT ==> 270	BYTES RECEIVED ==> 270
				COMPRESSION% ==> -1.5	COMPRESSION% ==> -1.5
				VOLSER ==> USER15	VOLSER ==> WRKPK2

MMWATL1	1	Q83903	PUSH01	02/25/2011 PNODE = SC.DUB.MWATL1	SNODE = SC.DUB.MWATL1
				TRANSMISSION TIME = 0:00:00.04	COMPLETION CODE = 00000000 MSG ID = SCFA0001
				COPY STEP SUCCESSFUL.	

				SRC DSNAME IS MMWATL1.TCPIF.DATA.FILE	
				DEST DSNAME IS MMWATL1.FTST.A7BE04E	
				SENDING NODE ==> SC.DUB.MWATL1	RECEIVING NODE ==> SC.DUB.MWATL1
				BYTES READ ==> 266	BYTES WRITTEN ==> 266
				BLOCKS READ ==> 1	BLOCKS WRITTEN ==> 1
				RECS READ ==> 0	RECS WRITTEN ==> 0
				BYTES SENT ==> 270	BYTES RECEIVED ==> 270
				COMPRESSION% ==> -1.5	COMPRESSION% ==> -1.5
				VOLSER ==> USER15	VOLSER ==> WRKPK2

MMWATL1	2	Q83901	STEP01	02/25/2011 PNODE = SC.DUB.MWATL1	SNODE = SC.DUB.MWATL1
				TRANSMISSION TIME = 0:00:03.83	COMPLETION CODE = 0000000C MSG ID = SCFA0461
				PROCESS HAS BEEN SUSPENDED BY USER/OPERATOR REQUEST.	

				SRC DSNAME IS DEVSTG2.HOST.DMDSSU1.TSTFILE	
				DEST DSNAME IS MMWATL1.SMSTEST.HOST.DMDSSU1.A9A0663	
				SENDING NODE ==> SC.DUB.MWATL1	RECEIVING NODE ==> SC.DUB.MWATL1
				BYTES READ ==> 31426504	BYTES WRITTEN ==> 31403484
				BLOCKS READ ==> 1366	BLOCKS WRITTEN ==> 1365
				RECS READ ==> 0	RECS WRITTEN ==> 0
				BYTES SENT ==> 11199488	BYTES RECEIVED ==> 11199488
				COMPRESSION% ==> 64.4	COMPRESSION% ==> 64.3
				VOLSER ==> SMSUS1	VOLSER ==> SMSUS3

This report includes all NonPDS COPY step transmissions for a specified time period. The ARS software sorts COPY information for each SNODE by user ID and then in ascending order by Process number. The following table contains a description of the report fields.

Report Field	Descriptions
NODE	IBM Connect:Direct node where the statistics file is examined
USERID	IBM Connect:Direct user ID
PROC #	IBM Connect:Direct Process number
PROC NAME	IBM Connect:Direct Process name
STEP NAME	Name (or label) assigned to a COPY step
mm/dd/yyyy	Date COPY step began executing
PNODE	IBM Connect:Direct primary node
SNODE	IBM Connect:Direct secondary node
TRANSMISSION TIME	Elapsed time between the COPY step start and the COPY step completion
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format
MSG ID	IBM Connect:Direct message ID
message	Short message for IBM Connect:Direct message ID
SRC DSNAME	Source data set name
DEST DSNAME	Destination data set name
REPORT FIELD	Description of report
SENDING NODE	Name of node sending the data set
RECEIVING NODE	Name of node receiving the data set

Report Field	Descriptions
BYTES READ	Number of bytes read by sending node
BYTES WRITTEN	Number of bytes written by receiving node
BLOCKS READ	Number of blocks read by sending node
BLOCKS WRITTEN	Blocks written by receiving node (either blocks or records is displayed)
RECS READ	Number of records read by sending node
RECS WRITTEN	Number of records written by receiving node
BYTES SENT	Number of bytes sent by sending node
BYTES RECEIVED	Number of bytes received by receiving node
COMPRESSION%	Compression percentage for sending data set
COMPRESSION%	Compression percentage for receiving data set
VOLSER	Sending volume serial number
VOLSER	Receiving volume serial number

IBM Connect:Direct for z/OS PDS Copy Report

The IBM Connect:Direct for z/OS PDS Copy Report provides information for each COPY step involving the PDS data set to PDS data set transfer.

Following is an example of the IBM Connect:Direct for z/OS PDS Copy Report.

```

IBM Sterling Connect:Direct for z/OS PDS Copy Report
  NODE = SC.DUB.MWATL1
  TRANSMISSION INFORMATION
- USERID      PROC #   PROC NAME   STEP NAME
  *****
- MWATL1      4        PDS         STEP01
  02/25/2011 PNODE = SC.DUB.MWATL1   SNODE = SC.DUB.MWATL1
  TRANSMISSION TIME = 0:00:19.82      COMPLETION CODE = 00000008 MSGID= SVSG016I
  PDS ERROR - DIRECTORY FULL/NON-EXISTENT MEMBER/WRONG DSORG.
  =====
  SRC DSNAME IS MWATL1.JCL.LIB
  DEST DSNAME IS MWATL1.JCL.LIB3
  SENDING NODE ==> SC.DUB.MWATL1      RECEIVING NODE ==> SC.DUB.MWATL1
  BYTES READ   ==> 60436320           BYTES WRITTEN  ==> 60436320
  BLOCKS READ  ==> 5479               BLOCKS WRITTEN ==> 5479
  RECS READ    ==> 0                  RECS WRITTEN   ==> 0
  BYTES SENT   ==> 60458236           BYTES RECEIVED ==> 60458236
  COMPRESSION% ==> 0                  COMPRESSION%   ==> 0
  VOLSER       ==> USER34             VOLSER         ==> USER03

1
IBM Sterling Connect:Direct for z/OS PDS Copy Report
  NODE = SC.DUB.MWATL1
  TRANSMISSION INFORMATION
- USERID      PROC #   PROC NAME   STEP NAME
  *****
- MWATL1      4        PDS         STEP01
  02/25/2011 PNODE = SC.DUB.MWATL1   SNODE = SC.DUB.MWATL1
  TRANSMISSION TIME = 0:00:19.82      COMPLETION CODE = 00000004 MSGID= SCPA000I
  COPY STEP SUCCESSFUL.
  =====
  SRC DSNAME IS MWATL1.JCL.LIB
  DEST DSNAME IS MWATL1.JCL.LIB3
  SENDING NODE ==> SC.DUB.MWATL1      RECEIVING NODE ==> SC.DUB.MWATL1
  BYTES READ   ==> 60436320           BYTES WRITTEN  ==> 60436320
  BLOCKS READ  ==> 5479               BLOCKS WRITTEN ==> 5479
  RECS READ    ==> 0                  RECS WRITTEN   ==> 0
  BYTES SENT   ==> 60458236           BYTES RECEIVED ==> 60458236
  COMPRESSION% ==> 0                  COMPRESSION%   ==> 0
  VOLSER       ==> USER34             VOLSER         ==> USER03
  =====
  MEMBER LIST
  =====
  $$$SPACE    TO    $$$SPACE
  $$$SPACE    TO    $$$SPACE

```

This report lists all PDS COPY step transmissions by Process number for a specified time period. It lists sending data set members names with each destination data set member name.

The accuracy of the data in this report is not guaranteed if you restart IBM Connect:Direct using the TCQ=COLD initialization parameter during the time period specified. ARS uses the Process number in the statistics file to match PDS member names with a specific COPY step. Because a COLD restart begins

numbering Processes from 1, ARS may not associate PDS member names correctly with the appropriate PDS Copy step.

The following table describes the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is examined
USERID	IBM Connect:Direct user ID
PROC #	IBM Connect:Direct Process number
PROC NAME	IBM Connect:Direct Process name
STEP NAME	Name (or label) assigned to a COPY step
mm/dd/yyyy	Date COPY step began executing
PNODE	Name of primary node
SNODE	Name of secondary node
TRANSMISSION TIME	Time between the COPY step start and the COPY step completion
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format
MSG ID	IBM Connect:Direct message ID
message	Short message for IBM Connect:Direct message ID
SRC DSNAME	Source data set name
DEST DSNAME	Destination data set name
SENDING NODE	Name of node sending the data set
REPORT FIELD	Description of report
RECEIVING NODE	Name of node receiving the data set
BYTES READ	Number of bytes read by sending node
BYTES WRITTEN	Number of bytes written by receiving node
BLOCKS READ	Number of blocks read by sending node
BLOCKS WRITTEN	Number of blocks written by receiving node (blocks or records is displayed)
RECS READ	Number of records read by sending node
RECS WRITTEN	Number of records written by receiving node
BYTES SENT	Number of bytes sent by sending node
BYTES RECEIVED	Number of bytes received by receiving node
COMPRESSION%	Compression percentage for sending data set
COMPRESSION%	Compression percentage for receiving data set
VOLSER	Sending volume serial number
VOLSER	Receiving volume serial number
MEMBER LIST	List of sending member names and receiving member names

IBM Connect:Direct for z/OS Run Job Report

The IBM Connect:Direct for z/OS Run Job Report lists the jobs submitted for execution using the IBM Connect:Direct RUN JOB statement. Jobs are listed for a specified time period. Following is an example of the IBM Connect:Direct for z/OS Run Job Report.

1	IBM Connect:Direct for z/OS RUNJOB Report				
	USERID	PROC #	PROC NAME	STEP NAME	RUNJOB INFORMATION
	*****	*****	*****	*****	

	MWATL1	6	MWCOPY	RJ	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE =
	SC.DUB.MWATL1				COMPLETION CODE = 00000000 MSG ID =
	SRJA000I				C:D RUN JOB FUNCTION COMPLETED SUCCESSFULLY.
					=====
					DSNAME ==> MWATL1.JCL.LIB(NDMBAT1)
					JOB # ==> 93960
					JOB SUBMITTED ON SC.DUB.MWATL1

The ARS software sorts all RUN JOB steps for each SNODE by userid and then in ascending order by Process number.

The following table contains a description of the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is being examined
USERID	IBM Connect:Direct user ID
PROC #	IBM Connect:Direct Process number
PROC NAME	IBM Connect:Direct Process name
STEP NAME	Name (or label) assigned to a RUN JOB step
mm/dd/yyyy	Date RUN JOB step submitted to run
PNODE	IBM Connect:Direct primary node
REPORT FIELD	Description of report
SNODE	IBM Connect:Direct secondary node
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format
MSG ID	IBM Connect:Direct message ID
message	Short message for IBM Connect:Direct message ID
DSNAME	Data set name containing job stream to be submitted
JOB #	Number assigned to job by operating system
JOB SUBMITTED ON	IBM Connect:Direct node where the job is submitted to run

Run Task Report

The IBM Connect:Direct for z/OS Run Task Report tracks all tasks (programs) for a specified time period that executed under the control of the IBM Connect:Direct RUN TASK statement. Following is an example of the IBM Connect:Direct for z/OS Run Task Report.

IBM Sterling Connect:Direct for z/OS RUNTASK Report				NODE = SC.DUB.MWATL1	
				RUNTASK INFORMATION	
USERID	PROC #	PROC NAME	STEP NAME	*****	
*****	*****	*****	*****	*****	
MMATL1	7	RTBR14	STEP1	02/25/2011	PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1
				COMPLETION CODE = 00000000	MSG ID = SRTA000I
				C:D RUN TASK FUNCTION SUCCESSFULLY COMPLETED.	
				=====	
				PROGRAM NAME ==> IEFBR14	
				PROGRAM ATTACHED ON SC.DUB.MWATL1	
0				-----	
MMATL1	8	RTBR14	STEP1	02/25/2011	PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1
				COMPLETION CODE = 00000000	MSG ID = SRTA000I
				C:D RUN TASK FUNCTION SUCCESSFULLY COMPLETED.	
				=====	
				PROGRAM NAME ==> IEFBR14	
				PROGRAM ATTACHED ON SC.DUB.MWATL1	
0				-----	
MMATL1	9	RTBR14	STEP1	02/25/2011	PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1
				COMPLETION CODE = 00000000	MSG ID = SRTA000I
				C:D RUN TASK FUNCTION SUCCESSFULLY COMPLETED.	
				=====	
				PROGRAM NAME ==> IEFBR14	
				PROGRAM ATTACHED ON SC.DUB.MWATL1	
0				-----	
MMATL1	10	RTBR14	STEP1	02/25/2011	PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1
				COMPLETION CODE = 00000000	MSG ID = SRTA000I
				C:D RUN TASK FUNCTION SUCCESSFULLY COMPLETED.	
				=====	
				PROGRAM NAME ==> IEFBR14	
				PROGRAM ATTACHED ON SC.DUB.MWATL1	

The ARS software sorts all RUN TASK steps for each SNODE by userid and then in ascending order by Process number.

The following table contains a description of the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is being examined
USERID	IBM Connect:Direct user ID
PROC #	IBM Connect:Direct Process number
PROC NAME	IBM Connect:Direct Process name
STEP NAME	Name (or label) assigned to a RUN TASK step
mm/dd/yyyy	Date RUN TASK step attached and executed
PNODE	IBM Connect:Direct primary node
REPORT FIELD	Description of report
SNODE	IBM Connect:Direct secondary node
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format
MSG ID	IBM Connect:Direct message ID
message	Short message for IBM Connect:Direct message ID
PROGRAM NAME	Name of program module attached
PROGRAM ATTACHED ON	IBM Connect:Direct node where the task is attached and executed

IBM Connect:Direct for z/OS Submit Within a Process Report

The IBM Connect:Direct for z/OS Submit Within a Process Report lists Processes that are submitted to Connect:Direct for execution using the IBM Connect:Direct Submit (within a Process) statement for a specified time period. Following is an example of the IBM Connect:Direct for z/OS Submit Within a Process Report.

IBM Connect:Direct for z/OS Submit within a Process Report				
USERID	PROC #	PROC NAME	STEP NAME	NODE = SC.DUB.MWATL1 SUBMIT INFORMATION
*****	*****	*****	*****	

- MWATL1	1	OS3903	STEP01	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
0				DSNAME ==> MWATL1.NDM.PROCESS.LIB(OS3903) PROCESS SUBMITTED ON SC.DUB.MWATL1
- MWATL1	4	PDS	STEP2	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
0				DSNAME ==> MWATL1.NDM.PROCESS.LIB(PDS) PROCESS SUBMITTED ON SC.DUB.MWATL1
- MWATL1	4	PDS	STEP2	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
0				DSNAME ==> MWATL1.NDM.PROCESS.LIB(PDS) PROCESS SUBMITTED ON SC.DUB.MWATL1
- MWATL1	7	RTBR14	STEP2	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
0				DSNAME ==> MWATL1.NDM.PROCESS.LIB(RTBR14) PROCESS SUBMITTED ON SC.DUB.MWATL1
- MWATL1	8	RTBR14	STEP2	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
0				DSNAME ==> MWATL1.NDM.PROCESS.LIB(RTBR14) PROCESS SUBMITTED ON SC.DUB.MWATL1
- MWATL1	9	RTBR14	STEP2	02/25/2011 PNODE = SC.DUB.MWATL1 SNODE = SC.DUB.MWATL1 COMPLETION CODE = 00000000 MSG ID = SSUB000I THE SUBMIT CONTROL BLOCK HAS BEEN SUCCESSFULLY CONSTRUCTED.
=====				
				DSNAME ==> MWATL1.NDM.PROCESS.LIB(RTBR14) PROCESS SUBMITTED ON SC.DUB.MWATL1

The ARS software sorts all SUBMIT steps for each SNODE by userid and then in ascending order by Process number.

The following table contains a description of the report fields.

Report Field	Description
NODE	IBM Connect:Direct node where the statistics file is being examined
USERID	IBM Connect:Direct user ID
PROC #	IBM Connect:Direct Process number
PROC NAME	IBM Connect:Direct Process name
STEP NAME	Name (or label) assigned to a SUBMIT (within a Process) step

Report Field	Description
mm/dd/yyyy	Date SUBMIT (within a Process) step ran
PNODE	IBM Connect:Direct primary node
REPORT FIELD	Description of report
SNODE	IBM Connect:Direct secondary node
COMPLETION CODE	IBM Connect:Direct completion code in hexadecimal format
MSG ID	IBM Connect:Direct message ID
message	Short message for IBM Connect:Direct message ID
DSNAME	Data set name containing the IBM Connect:Direct Process Only available when SUBNODE=PNODE.
PROCESS SUBMITTED ON	IBM Connect:Direct node where the Process was submitted to run

Activity Reporting System

The Activity Reporting System (ARS) produces reports of IBM Connect:Direct activity. While IBM Connect:Direct produces statistics, ARS enables you to access more information and provides sorting capabilities.

You can request ARS reports in three ways:

- Through ARS screens using IBM Time Sharing Option/Interactive System Productivity Facility (TSO/ISPF)
- Automatically through a IBM Connect:Direct Process
- Schedule a batch job through a job scheduling subsystem

With ARS, you do not have to pass data to z/OS system management facilities (SMF) or write SAS requests to print IBM Connect:Direct activity reports. Use ARS as a standard reporting format throughout a network for tracking IBM Connect:Direct activity.

Note: ARS requires a SAS base running under z/OS and Connect:Direct for z/OS must be active.

ARS and IBM Connect:Direct

IBM Connect:Direct automatically collects statistical data and stores it in a data set. ARS can access this statistical data and produce the following reports for a selected time period:

- IBM Connect:Direct Activity Report lists activity by Process step
- IBM Connect:Direct Summary Report summarizes all activity

IBM Connect:Direct Exception Report lists all Process steps that are not successful

IBM Connect:Direct Security Violations Report lists security violations by signon security failures, Process security failures, and data set access security failures

IBM Connect:Direct Function Reports provides the activity reports for the following types of Process steps:

- Non-Partitioned Data Set (PDS) COPY
- PDS COPY
- RUN JOB
- RUN TASK
- Submit Within a Process

Following are two methods to request ARS reports:

- Through the ARS screens where you can create a job stream that requests reports. The screens automatically build the job or allow you to edit the sample report job stream.
- By using the sample JCL member that is provided with ARS.

Business Solutions Using ARS

ARS provides data center management with additional tools to monitor and track IBM Connect:Direct usage. Use the information in the reports to track attempted security violations, analyze capacity planning related data, examine IBM Connect:Direct utilization, and isolate failed IBM Connect:Direct Processes.

The following table describes how to use ARS.

Function	Description
Data Center Management	The data center manager wants to keep track of IBM Connect:Direct activities. The IBM Connect:Direct Activity Report lists all Process steps that occurred in this data center during a time period. The IBM Connect:Direct NonPDS Copy Report and IBM Connect:Direct PDS Copy Report provide additional information on all COPY steps performed.
Security Violations	A system administrator requests a list of any IBM Connect:Direct-related security violations on a daily basis. The IBM Connect:Direct Security Violations Report lists the following violations: <ul style="list-style-type: none"> • Sign on security failures • Processes not run due to lack of authorization • Data set access security violations
Capacity Planning	The capacity planner at a data center needs to know how many bytes are transferring between this data center and other data centers and the total transmission times. The IBM Connect:Direct Summary Report provides this type of information by remote data center for a specified time period.
IBM Connect:Direct Utilization	The data center administrator studies IBM Connect:Direct utilization by examining how many application programs are submitted or invoked using IBM Connect:Direct. The administrator uses this information with a job scheduling system to produce a comprehensive analysis of all jobs run at a data center. Two ARS reports provide this type of utilization summary. The IBM Connect:Direct Run Job Report provides information about all jobs submitted by IBM Connect:Direct. The IBM Connect:Direct Run Task Report provides information about all programs executed under the control of IBM Connect:Direct.
Problem Isolation	The IBM Connect:Direct Exception Report is an excellent tool for researching why a Process does not run. This report lists each failed Process for a requested time period along with the reason for the failure.

Requesting ARS Statistical Reports

To request ARS statistical reports, complete the following procedures:

- Request the ARS report using ARS screens
- Run the job

ARS screens build a job stream to produce any of the ARS reports. For this reason, the screens are most useful in cases where you need individual reports instead of volume reports.

After ARS screens build the job stream and submit it to run, the job performs the following:

- Signs on to IBM Connect:Direct
- Creates a temporary data set to hold the specified statistics
- Accesses the IBM Connect:Direct statistics file and copies statistics to a temporary data set
- Executes the ARS report routine using the IBM Connect:Direct statistics as input data
- Sends the report to the selected output location
- Deletes the temporary data set

Requesting Reports Using ARS Screens

About this task

The ARS screens identify the information to build the job. Certain information is retained between sessions to prevent the need to enter the same information each time you request a report.

To access the ARS screens through the IUI:

Select ADMIN from the Primary Options Menu.

```

View  Modify  Control  Create  Execute  Help
-----
CD.ART          IBM Connect:Direct for z/OS          Primary Options Menu
Option ==>

Select one of the following:

CF   - Copy a file                                     *****
SB   - Submit a predefined process                     *
DF   - Define a process using ISPF Edit                 * Today: 08.30.2018 *
SS   - View statistics for a completed process          * Time: 14:00      *
S2   - Advanced view statistics                       * UID:  USER12    *
MB   - Submit a batch to Connect:Enterprise for z/OS   *
CP   - Change characteristics of a process             * OPT Enabled     *
DP   - Delete a non-executing process                 * OPT Part-Enabled *
FP   - Flush an executing process                     * OPT Disabled    *
SP   - View data about an executing process            *
PS   - Suspend an executing process                   *****
MSG  - View Connect:Direct message text
SW   - Swap among concurrent Connect:Direct sessions
SD   - View/Change your Connect:Direct signon defaults
NM   - View information in the Connect:Direct network map
WHO  - View characteristics of your Connect:Direct IUI environment
SPF  - Enter ISPF/PDF
AUTH - View your Connect:Direct function authorization
MS   - Sign on to multiple Connect:Direct nodes concurrently
ADMIN - Perform Connect:Direct administrative functions

```

Procedure

1. From the Administrative Options Menu, select ARS.


```

View  Modify  Control  Delete  Secure+  Help
-----
CD.ART          Connect:Direct Administrative Options Menu
Option ==>

Select one of the following:

ST  - View type record
IT  - Insert/Update type record
DT  - Delete type record
SU  - View user authorization record
IU  - Insert/Update user authorization record
DU  - Delete user authorization record
TS  - View Connect:Direct tasks
TF  - Flush a Connect:Direct task
S   - Execute Secure Plus Commands
MD  - Modify Connect:Direct trace characteristics
C   - Enter a native Connect:Direct command
SN  - Terminate Connect:Direct
ARS - ARS reporting facility
NM  - View information in the Connect:Direct network map
UNM - Update the Connect:Direct network map
INQ - Inquire about DTF internal status
STAT - Perform statistics functions

*****
*                                     *
* Today: 08.30.2018 *
* Time: 14:05      *
* UID:  USER12    *
* OPT Enabled      *
* OPT Part-Enabled *
* OPT Disabled     *
*                                     *
*****

```

2. To request the report, provide signon data, and specify the time period for the report, continue with the following steps.
3. To submit the job stream without changing any data, type SUB (submit) at the CMD field and then press Enter.

```

$cd.node          Connect:Direct for z/OS - ARS REPORT OPTIONS
CMD ==>
                                hh:mm

AC - ACTIVITY                PS - NON-PDS COPY
SM - SUMMARY                 PO - PDS COPY
EX - EXCEPTION               RJ - RUNJOB
SC - SECURITY                 RT - RUNTASK
                                SB - SUBMIT

REPORT TYPE ==>              EDIT JCL ==> (Y,N)
Connect:Direct SIGNON PARAMETERS
-----
USER ID      ==> $uid
PASSWORD     ==>
NETMAP NAME  ==> $cd.netmap
TRANSPORT    ==> NET
COMMUNICATION ADDRESS ==> (      ,      )
REPORTING RANGE
-----
START DATE   ==>              START TIME ==>
STOP  DATE   ==>              STOP  TIME ==>

```

The following table contains a description of the screen fields.

Field	Description
REPORT TYPE	Specifies the ARS report.
EDIT JCL	Specifies an option to edit JCL.
USER ID	Specifies the userid for signing on to IBM Connect:Direct, if your security environment requires it. (optional)
PASSWORD	Specifies the password needed to access IBM Connect:Direct. (optional)
NETMAP NAME	Specifies the name of the IBM Connect:Direct Network Map.
TRANSPORT	Enables override of signon defaults for DGADBATC transport type.
COMMUNICATION ADDRESS	Specifies port number and IP address with TRANSPORT=TCP.

Field	Description
REPORTING RANGE	<p>Identifies the date, day or time period that you want the requested report to cover. (optional)</p> <p>If all four fields are blank, the contents of the statistics file is used.</p>
START DATE or STOP DATE	<p>Specifies the date or day that the statistics records are selected for the ARS report.</p> <p>You can specify the day (dd), month (mm), and year (yy for 2-digit year and yyyy for 4-digit year). Use periods or backslashes (\) to separate the components of a date value.</p> <p>You can specify the date (dd), month (mm), and year (yy) in one of the following formats: yymmdd or yyyyymmdd; yy/mm/dd or yyyy/mm/dd; yy.mm.dd or yyyy.mm.dd; mmddyy or mmddyyyy; mm/dd/yy or mm/dd/yyyy; mm.dd.yy or mm.dd.yyyy; or the Julian date, yyddd or yyyyddd; yy/ddd or yyyy/ddd; or yy.ddd or yyyy.ddd. If you only specify the date, the time defaults to 00:00.</p> <p>This date must have the same format as specified in the DATEFORM initialization parameter.</p> <p>You can also use day in these fields to indicate day of the week for which the statistics records are searched. Valid names include MONday, TUESday, WEdnesday, THursday, FRiday, SATurday, and SUNday.</p> <p>You can also specify TODAY, which searches for the statistics records today; or TOMORROW, which searches for statistics records the next day; or YESTER, which searches for statistics records for yesterday.</p>
START TIME or STOP TIME	<p>Specifies the time of day in hours (hh), minutes (mm), and seconds (ss) when the statistics records are selected for the ARS report. Specify AM or PM. You can express the time of day using the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, you can express 1:00 hours as 1:00AM, and you can express 1:00 PM as 13:00 hours. If you do not use AM and PM, the 24-hour clock is assumed. You do not need to specify minutes and seconds.</p> <p>You can also specify NOON, which searches for the statistics records at noon, or MIDNIGHT, which searches for the statistics records at midnight.</p> <p>The default for the time is 00:00:00, the beginning of the day.</p> <p>If you do not specify START TIME and STOP TIME but you do specify the START DATE and STOP DATE, the time range default is 00:00 to 24:00.</p>

ARS Connect:Direct Requirements Screen

The ARS Connect:Direct Requirements screen in the following figure identifies specific job stream parameters for the job card and IBM Connect:Direct-related information.

```

$cd.node          Connect:Direct for z/OS - ARS (Job and File Requirements)
CMD ==>                                                    hh:mm
UP TO 3 LINES FOR JOBCARD INFO
==> -----
==> -----
==> -----
STEPLIB DATA SET NAME FOR Connect:Direct PROGRAMS (OPTIONAL):
==> -----
==> -----
Connect:Direct PUBLIB PROCESS LIBRARIES (1 REQUIRED):
==> -----
==> -----
Connect:Direct MESSAGE DATA SET NAME (REQUIRED):
==> -----
UNIT SPECIFICATION FOR Connect:Direct TEMPORARY DATASET:
==> -----

```

The following table contains a description of the screen fields.

Field	Description
JOBCARD INFO	Typical job card information. Use the COND=(0,NE) for best results, meaning the job does not run if an error is found when configuring the job.
STEPLIB DATA SET NAME FOR Connect:Direct PROGRAMS	Name of the library containing the IBM Connect:Direct load modules. You can type two libraries in this section. Optional if the IBM Connect:Direct load modules are available via LINKLIST otherwise a STEPLIB is required.
Connect:Direct PUBLIC PROCESS LIBRARIES	Specifies the name of the library containing IBM Connect:Direct Processes.
Connect:Direct MESSAGE DATA SET NAME	Specifies the name of the message data set containing the IBM Connect:Direct messages.

ARS SAS Requirements Screen and Fields

The ARS SAS Requirements screen in the following figure identifies specific JCL parameters for SAS-related information.

```

$cd.node          Connect:Direct for z/OS - ARS (SAS Files and Output)
CMD ==>                                                    hh:mm
SAS CATALOGED PROCEDURE (REQUIRED):
==> -----
DATASET NAME CONTAINING SAS ROUTINES (REQUIRED):
==> -----
ONE OF THE FOLLOWING MUST BE PROVIDED :
OUTPUT DATA SET NAME :
==> -----
OUTPUT (SYSOUT) CLASS :
==> -

```

The following table contains a description of the screen fields.

Field	Description
SAS CATALOGED PROCEDURE	Identifies the name of the SAS cataloged procedure, which is a collection of JCL statements required to execute SAS for batch processing at your installation.
DATA SET NAME CONTAINING SAS ROUTINES	Specifies where the ARS routines (\$CD.SDGAMAP) that interact with SAS are located.

Field	Description
OUTPUT DATA SET NAME	<p>You must preallocate the sequential data set designated in this field with the following attributes:</p> <p>RECFM=FBA</p> <p>LRECL=240</p> <p>BLKSIZE=3120</p> <p>If this field is left blank, you must specify the OUTPUT (SYSOUT) CLASS field.</p>
OUTPUT (SYSOUT) CLASS	<p>Specifies the SYSOUT class that automatically sends the output to a designated printer queue (optional). If you leave this field blank, you must specify the OUTPUT DATASET NAME field.</p> <p>Note: If you are requesting the IBM Connect:Direct Summary report and routing it to a data set, ensure that the data set is empty. If the data set is not empty, this report is placed after the data already in the data set. The IBM Connect:Direct Summary Report is automatically allocated with DISP=MOD instead of DISP=SHR.</p>

Job Streams Created

After completing the procedure, ARS builds a job stream, as in the following example. If you requested EDIT, ARS displays the job after completing the screens.

You can review the job stream before submitting it. Following is an example of the JCL which is generated using member DGAJARS from \$CD.SDGASENU.

```
//JOBNAME JOB (ACCT), 'NAME', NOTIFY=TSOID,TIME=((1),
// REGION=0M,MSGCLASS=X,CLASS=B
// *
// *
//DGADBATC EXEC PGM=DGADBATC,PARM='YYSLN'
//STEPLIB DD DISP=SHR,DSN=$CD.SDGALINK
//SYSUDUMP DD SYSOUT=*
//DMPUBLIB DD DISP=SHR,DSN=$CD.SDGAPROC
//DMMSGFIL DD DISP=SHR,DSN=$CD.MSG
//TEMPDSN DSN=&CDAPI,DISP=(NEW,PASS),UNIT=PTEMP,
// DCB=(DSORG=PS,RECFM=VBA,LRECL=4100,BLKSIZE=4104),
// SPACE=(4104,(70,13))
//DMPRINT DD SYSOUT=*
//SYSIN DD *
SIGNON USERID=(NAME,,) -
NETMAP=$CD.NETMAP TMPDD=TEMPDSN
SEL STAT WHERE (STARTT = (06/1/2010,12:00AM) -
STOPT = 07/1/2010,11:59PM) ) FILE
SIGNOFF
// *
// *
///SASTEP EXEC $SASPROC,
// OPTIONS='DQUOTE MACRO MACROGEN MERROR MISSING="- "'
//WORK DD UNIT=PTEMP,SPACE=(CYL,(20,10))
//NDMX0001 DD DISP=(OLD,DELETE),DSN=&CDAPI
//DMMSGFIL DD DISP=SHR,DSN=$CD.MSG
//FT20F001 DD DISP=SHR,DSN=$OUTPUT.DATASET.NAME
//SYSIN DD DISP=SHR,DSN=$CD.SDGAMAP(ACTIVITY)
```

Running the Job

When a job runs, it performs the following:

- Signs on to IBM Connect:Direct
- Creates a temporary data set to hold the specified statistics
- Copies statistics to the temporary data set

- Executes the ARS report routine using the IBM Connect:Direct statistics as input data
- Sends the report to the selected output location
- Deletes the temporary data set

Requesting Multiple ARS Reports or Scheduled Processing

You can request ARS reports without using the ARS screens. You can submit multiple ARS reports at one time or submit ARS reports for scheduled processing.

You can request ARS reports without using ARS screens by editing a sample job stream to specify processing requirements and report types. Also, when the output of a multiple report request is routed to a data set, some ARS report routines can require a minor edit change.

Modifying the Sample Job Stream

Modify the sample job stream in \$CD.SDGAJCL(DGAJARS2) in order to customize the job stream with your companies information.

About this task

These steps provide line-by-line instructions for modifying the sample job stream. These modifications are also detailed in the \$CD.SDGAJCL(DGAJARS2) file.

Procedure

1. Modify the job card to uniquely identify your job.
2. Change \$CD.SDGAJCL to the appropriate IBM Connect:Direct load library name.
3. Change \$CD.SDGAJPROC to the appropriate IBM Connect:Direct Process library name.
4. Change all occurrences of \$CDVSAM.MSG to the IBM Connect:Direct message data set name.
5. Change \$UID to your IBM Connect:Direct user ID. Also add the password, if needed.
6. Change \$CDVSAM.NETMAP to your Network Map data set name.
7. Change all occurrences of \$UNITNAME to the valid unit name.
8. Change \$SASPROC to the name of the SAS cataloged procedure used at your installation.
9. Route the output to one of the two options:
 - Route to the SYSOUT class. Look for FT20F001 DD.
 - Change all occurrences of \$OUTPUT.DATASET.NAME to the name of the data set where you route the output. Preallocate this data set as FBA with an LRECL of 240 and BLKSIZE of 3120. Ensure that the data set is empty before running this job. Complete the instructions in [Multiple Reports to an Output Data Set](#).
10. Change \$CD.SDGAJMAP to the name of the data set containing the SAS programs.

Note: If you would like to define the time span that will be reported, modify the SEL Statistics command in the SYSIN stream of the DMBATCH step to include start and stop times. The default is a report on the entire STAT file.

Job Stream Definitions

The following table defines the parameters and SAS options for the sample job stream.

Report Field	Definition
JOBCARD INFO	Specifies the typical job card information. Use the COND=(0,NE) for best results. This means that the second step of the job does not run if an error occurs in the first step.
DGADBATCH	Specifies the program name of the IBM Connect:Direct batch interface.

Report Field	Definition
STEPLIB	Specifies the library containing the IBM Connect:Direct load modules.
DMPUBLIB	Specifies the library containing the IBM Connect:Direct Processes.
DMMSGFIL	Specifies the IBM Connect:Direct message data set that accesses the messages.
TEMPDSN	Specifies the temporary data set containing the extract from the IBM Connect:Direct statistics file that is used as input to SAS. This is always specified as (NEW, PASS) so that when you create the data set, the extracted data is saved in it, and then passed on to the next step.
DMPRINT (Optional)	Specifies where the job output from DGADBATC goes. This is useful if an error occurs in DGADBATC.
SYSPRINT	Specifies where the job execution messages goes.
SYSIN	Contains the IBM Connect:Direct control statements to extract from the statistics file. Represent SYSIN as a sequential data set, PDS member, or instream data set.
SASTEP	Specifies the name of the SAS cataloged procedure used at your installation with the following options: DQUOTE specifies that the system accepts double quotes. MACRO specifies that the SAS macro library is available. MACROGEN specifies that the system can print SAS macros. MERROR specifies that a warning message is produced if a name is prefixed with a percent sign to indicate a SAS macro, but the name is not a valid SAS macro. MISSING specifies that dashes (-) are inserted when a numeric character is missing in the report field.
WORK	Specifies the work area used by SAS for processing.
NDMX0001	Specifies the temporary data set that contains the extract from the IBM Connect:Direct statistics file. Specify the data set parameters (OLD, DELETE) so that the data set is deleted automatically after processing.
DMMSGFIL	Specifies the name of the IBM Connect:Direct message data set.
SASPROGS	Identifies the name of the data set containing the SAS routines to produce the reports. See %INCLUDE parameter in this table.
FT20F001	Specifies the DD statement used by SAS to route output. This DD statement can be either output to a data set or SYSOUT class. If it is to a data set, you must preallocate the data set, it must be sequential, and have the following attributes: RECFM=FBA; LRECL=240; BLKSIZE=3120.
SYSIN	Specifies SYSIN for SAS.

Report Field	Definition
%INCLUDE	<p>Contains a special SAS control statement that enables you to execute SAS programs that are stored separately from their JCL. The first name after SASPROGS identifies the DD name that references the location of the SAS programs. The requested ARS report routine names are placed inside of the parentheses:</p> <p>DGAACTIV specifies the IBM Connect:Direct for z/OS Activity Report</p> <p>DGAAEXEC specifies the IBM Connect:Direct for z/OS PDS Copy Report</p> <p>DGAARUNJ specifies the IBM Connect:Direct for z/OS Run Job Report.</p> <p>DGAARUNT specifies the IBM Connect:Direct for z/OS Run Task Report.</p> <p>DGAASEC specifies the IBM Connect:Direct for z/OS Security Violations Report.</p> <p>DGAASUB specifies the IBM Connect:Direct for z/OS Submit Within a Process Report.</p> <p>DGAASUM specifies the IBM Connect:Direct for z/OS Summary Report.</p>

Sample IBM Connect:Direct Process That Submits Job Stream

The DGAXRPRC member in \$CD.SDGASAMP in the following figure contains a IBM Connect:Direct Process that runs the job stream discussed in the previous section, [Modifying the Sample Job Stream](#).

You can set up the sample Process to run automatically on a specified time interval, by submitting this Process with the IBM Connect:Direct SUBMIT command, specifying the STARTT parameter and RETAIN=YES parameter.

```

RPTPROC          -
* THIS PROCESS WILL SUBMIT FOR EXECUTION THE JCL TO RUN ARS
* CHANGE $SECONDARY.NODE TO THE DESIRED SNODE NAME FOR THE PROCESS
* CHANGE $TSOID TO YOUR TSO ID
* CHANGE $CD.SDGACNTL(DGAJARS) TO THE DATA SET NAME CONTAINING THE JCL
*          TO RUN ARS
PROCESS          -
  SNODE=$SECONDARY.NODE -
  NOTIFY=$TSOID
STEPONE         -
  RUN JOB (      -
    DSN=$CD.SDGACNTL(DGAJARS) -
  )

```

You need to modify the following items if you choose to use this sample Process. Copy DGAXRPRC from \$CD.SDGASAMP to your process library before editing it.

1. Change \$SECONDARY.NODE to the appropriate SNODE.
2. Change \$TSOID to your TSO ID.
3. Change \$CD.SDGACNTL(DGAJARS) to the data set name containing the job that runs ARS.

Multiple Reports to an Output Data Set

Complete this procedure only when you request multiple ARS report types in one job and the reports are sent to an output data set. The IBM Connect:Direct Activity Report and Security Violations Report require a minor edit change in the ARS routine if you include them in a multiple report request.

Before modifying the ARS routines, make a copy of the \$CD.SDGAMAP, which contains all report routines. Use the first copy when you request one report (in batch mode or using the ARS screens). The second copy can contain the modified routines when you make multiple report requests in a job.



CAUTION: \$CD.SDGAMAP contains members ALIASed. If you edit an alias, it becomes a member.

You can submit the ARS report routines using TSO. The routines are in the \$CD.SDGAMAP. The following table lists the types of reports and the corresponding members in the \$CD.SDGAMAP that require modification.

Report Type	Member Name
IBM Connect:Direct Activity Report	DGAACTIV
IBM Connect:Direct Security Violations Report	DGAASEC

Modify the PROC PRINTTO statement listed in the members so that the output from one report appends rather than overlaying output from the previous report in the designated output file. Change the ARS report routine from:

```
PROC PRINTTO NEW UNIT=20
```

to:

```
PROC PRINTTO UNIT=20;
```

ARS Record Layouts

To customize ARS reports, modify the ARS reports or develop new reports. Examples include:

- Change ARS report headings, spacing, field titles, and output format
- Access additional information from the IBM Connect:Direct statistics file to enhance ARS reports or to develop new reports

The ARS routines do not use all of the IBM Connect:Direct fields in the IBM Connect:Direct statistics records. The SAS Informat variables for these records are located in the \$CD.SDGAMAP library. Access the information by the member names listed in the following table.

Note: Use two-character designations for records types when browsing the IBM Connect:Direct statistics file.

Member Name	Contents of Member	Record Types
DGAAAER	Authorization Event Record	IU=INSert USER UU=UPDate USER SU=SELEct USER DU=DELeTe USER
DGAACPTR	Change Process Termination Record	CH=CHange PROCess
DGAACTR	Copy Termination Record	CT=COPIY
DGAADPTR	Delete Process Termination Record	DP=DELeTe PROCess
DGAADTR	Display Statistics Record	SP=SELEct PROCess DT=SELEct TASK FT=FLUSH TASK SS=SELEct STATistics SN=SELEct NETmap
DGAAFPTR	Flush Process Termination Record	FP=FLUSH PROCess
DGAAFMCR	PDS Member Copy Record	MC=PDS member COPIY
DGAAPPSR	Process Submit Statistics Record	PS=SUBmit statement SW=SUBmit command
DGAAPTR	Process Termination Record	PT=PROCess
DGAARJTR	Run Job Termination Record	RJ=RUN JOB
DGAARTTR	Run Task Termination Record	RT=RUN TASK

Member Name	Contents of Member	Record Types
DGAASFR	Signon/Signoff Statistics Record	SI=SIGNON SO=SIGNOFF
DGAASDCR	Start IBM Connect:Direct Command Record	SD=Start IBM Connect:Direct
DGAASTDC	Stop IBM Connect:Direct Statistics Record	ST=STOP IBM Connect:Direct
DGAAWTOS	Write to Operator (WTO) Statistics Record	WT=Write to Operator

Description of an SAS Variable

The SAS Informat variables for these records are located in the \$CD.SDGAMAP library. An example of an SAS Informat variable is @5 SASFIELD PK1. Variable descriptions for this example are listed in the following table.

Variable	Description
@n	Identifies the position of the variable in the IBM Connect:Direct statistics record (relative to 1). Always identified by an @.
SASFIELD	Specifies the SAS name for the corresponding IBM Connect:Direct field name. The SAS name variable must follow the position variable (@n).
PK1	Specifies the type and length of the SAS Informat variable used in the maps. The type variable must follow the name variable (SASFIELD). Type variable used in the maps are: HEX—numeric hexadecimal IB—integer binary PD—packed decimal PIB—positive integer binary PK—packed unsigned \$n—standard character \$CHAR—characters with blanks \$HEX—character hexadecimal \$VARYING—variable-length values

Authorization Event Record

The following table shows each field available in the Authorization Event record. The DGAAAER member of the \$CD.SDGAMAP library contains the SAS maps of the IBM Connect:Direct statistics record.

Field Names	Field Description
AERECLN	Length of this record.
AERTYPE	Record type indicates specific data in statistics record: IU=INSert; UU=UPDate USER; SU=SELEct USER; and DU=DELeTe USER.
AERTIME	Time that event was recorded in hh/mm/ss/tt format.
AEDATE	Date that event was recorded in Julian date format (yyyydddf).
AEPROCNM	Process name.
AEPROCNO	Process number.
AEUNODE	User node of the submitter.
AEUID	Userid of the submitter.
AESTEP	Step name or label.
AESTIME	Time that Process started in hh/mm/ss/tt format.
AESDATE	Date that Process started in Julian date format (yyyydddf).
AESCC	Step completion code.
AEMSGID	Message ID.

Field Names	Field Description
AESDSNL	Length of data set name.
AESDSN	Source data set name.
AESDSTYP	Data set type (LIB, DSN).
AEEVENT	Event code.
AEEVENTQ	Event code qualifier.

Change Process Termination Record

The following table shows each field available in the IBM Connect:Direct Change Process Termination record. The DGAACPTR member of the \$CD.SDGAMAP library contains the SAS maps of the IBM Connect:Direct statistics record.

Field Names	Field Description
CHRECLN	Length of this record.
CHRTYPE	Record type – CH indicates the Change Process Termination Record.
CHTIME	Time that CHange PROCess command completed in hh/mm/ss/tt format.
CHDATE	Date that CHange PROCess command completed in Julian date format (yyyydddf).
CHPROCNM	Process name.
CHPROCNO	Process number.
CHSTEP	Step name or label.
CHUNODE	User node ID of the submitter.
CHUID	Userid of the submitter.
CHSTIME	Time that CHange PROCess command started in hh/mm/ss/tt format.
CHSDATE	Date that CHange PROCess command started in Julian date format (yyyydddf).
CHSCC	Step completion code-displays normal completion code.
CHMSGID	Message ID.
CHRMNID	Node where message routed.
CHRMUID	Userid where message routed.
CHSCHTME	Time that Process was scheduled in hh/mm/ss/tt format.
CHSCHDTE	Date that Process was scheduled in Julian date format (yyyydddf).
CHSCHDAY	Day that Process was scheduled.
CHPRTY	New Process priority in the IBM Connect:Direct Transmission Control Queue (TCQ).
CHRETAIN	Keeps copy in TCQ after execution.
CHPROC	ALL/PROCNAME/PROCNUMBER.
CHNDEST	New destination node.
CHHOLD	Process status in hold queue: Q=quiesce or I=immediate.
CHNPRTY	New priority.
CHREL	Release Processes=R.

Field Names	Field Description
CHRET	Retain Processes=R.
CHNRMNID	New node ID messages to be routed.
CHNRMUID	New userid messages to be routed.
CHNEWTME	New scheduled time in hh/mm/ss/tt format.
CHNEWDTE	New scheduled date in Julian date format (yyyydddf).
CHNPLCLP	New PNODE PLEXCLASS
CHPLCLP	Old PNODE PLEXCLASS value
CHNPLCLS	New SNODE PLEXCLASS value
CHPLCLS	Old SNODE PLEXCLASS value
CHNODNM	NODE name to be changed
CHFRSH	'R' is refresh from NM

Copy Termination Record

The following table shows each field available in the Copy Termination record. The DGAACR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
CTCLASS	Session class.
CTRECLN	Length of this record.
CTRTYPE	Record type – CT indicates the Copy Termination record.
CTTIME	Time that COPY step completed in hh/mm/ss/tt format.
CTDATE	Date that COPY step completed in Julian date format (yyyydddf).
CTSTIME	Time that COPY step started in hh/mm/ss/tt format.
CTSDATE	Date that COPY step started in Julian date format (yyyydddf).
CTSCC	Step completion code. Displays normal completion code.
CTMSGID	Message ID.
CTPROCNM	Process name.
CTPROCNO	Process number.
CTSTEP	Step name or label.
CTUNODE	User node ID of the submitter.
CTUID	Userid of the submitter.
CTPNODE	Name of the node examining its IBM Connect:Direct statistics file.
CTSNODE	Name of the other node.
CTNODE	This node is P(node) or S(node).
CTFROM	Direction of data: Snode to Pnode or Pnode to Snode

Field Names	Field Description
CTTRANS	File translation. Includes: ASCII-TO-EBCDIC; EBCDIC-TO-ASCII; EBCDIC-TO-ASCII DIF; compressed; and compacted.
CTTRANS2	Transmission options. Includes: PDS-to-PDS copy; error originated on other node; sending PDS member to sequential DSN; sending DSN sequential DSN to PDS member; and COPY is restarted.
CTRUSZ	Request/response unit (RU) size.
CTPACCT#	Displacement to Pnode account data length.
CTSACCT#	Displacement to Snode account data length.
CTCERT#	Displacement to CERT Info.
CTCERI#	Displacement to CERI Info
CTFLAGX	Contains CTFXNODE flag
CTINBYTE	Number of bytes read from data set.
CTINRECN	Number of records read from data set.
CTINBLK	Number of blocks read from data set.
CTSBYTES	Number of bytes sent.
CTOBYTE	Number of bytes written to data set.
CTOTRECN	Number of records written to data set.
CTOTBLK	Number of blocks written to data set.
CTRBYTES	Number of bytes received.
CTNOKB	Number of kilobytes sent.
CTNOMEMS	Number of members sent.
CTNOALIS	Number of aliases sent.
CTNOMEMX	Number of members selected but not sent.
CTNOALIX	Number of aliases selected but not sent.
CTNOMEMR	Number of members received.
CTNOALIR	Number of aliases received.
CTCMPTBL	The compaction table name.
CTSDSTYP	Source data set type: ESDS, KSDS, RRDS, PDS, SAM, or LIB.
CTSDISP1	Source data set disposition status: O=OLD and S=SHR.
CTSDISP2	Source data set disposition normal termination: D=DELETE and K=KEEP.
CTSDISP3	Source data set disposition abnormal termination: D=DELETE and K=KEEP.
CTSDSNL	Length of the source data set name.
CTSDSN	Source data set name.
CTTYPE	Type file key, if Type is specified in a Process step.

Field Names	Field Description
CTDDSTYP	Destination data set type.
CTDDISP1	Destination data set disposition status: M=MOD; N=NEW; O=OLD; and S=SHR.
CTDDISP2	Destination data set disposition normal termination: C=CATLG and K=KEEP.
CTDDISP3	Destination data set disposition abnormal termination: C=CATLG; K=KEEP; and D=DELETE.
CTDDSNL	Length of destination data set name.
CTDDSN	Destination data set name.
CTMEMBER	Member name for PS to PO copies.
CTOMSGID	Message ID from other node.
CTRTNCD	Return code from other node.
CTNVTAMS	Number of VTAM sends.
CTNVTAMR	Number of VTAM receives.
CTDSPVLR	Displacement of receiving VOLSER list.
CTDSPVLS	Displacement of sending VOLSER list
CTTBOFS	TCP buffer size used.
CTV2BUFS	V2 buffer size.
CTNEGBF	Negotiated V2 buffer size.
CTECZL	Compression Routine (1=ZLIB, 2=LZW)
CTZLVL	Compression Level
CTZWIN	Compression window size (Send)
CTZMEM	Compression memory level
CTSEC#F1	Secure+ copy term FLAG-1
CTSEC#F2	Secure+ copy term FLAG-2
CTS#CPEA	Merged Cipher-suite number
CTSERVER	SYSPLEX Server name
CTCLASSP	SYSPLEX PNODE Server class
CTCLASSS	SYSPLEX SNODE Server class
CTZFLAG	ZEDC feature FLAG 1
CTUSERSN	User sense from FMH71 CCB
CTCKPI	checkpoint interval
CTRIP	inbound/outbound IP addr
CTRPOR	outbound port
CTZFLAG2	ZEDC FEATURE FLAG 2
CTZFLAG3	ZEDC FEATURE FLAG 3

Field Names	Field Description
CTDBTAB	DBCS table name
CTDBSI	DBCS shift-in character
CTDBSO	DBCS shift-out character
CTDBPAD	DBCS pad character
CTWINR	COMPRESSION WINDOW SIZE (Recv)
CTCPU	CPU time of step via TIMEUSED
CTRIP6HI	IPV6 ADRESS; NETWORK FORM – HIGH WORD
CTRIP6LO	IPV6 ADRESS; NETWORK FORM – LOW WORD
CTTYPE	TYPE file KEY
CTSUBJOB	Submitter's jobname
CTSUBJID	Submitter's jobid
CTECP0S	CPU time spent running on a CP
CTECP1S	CPU time spent running on a zIIP
CTECP2S	zIIP qualified amount of CTECP0S
CTECP0R	CPU time spent running on a CP
CTECP1R	CPU time spent running on a zIIP
CTECP2R	zIIP qualified amount of CTECP0R
CTZFBA2	Reserved
CTZFBA1	zFBA Device Range
CTZLIBVS	ZLIB Version (Send) blanks=1.2.3
CTZLIBVR	ZLIB Version (Recv) blanks=1.2.3
CTSTEPOS	Step offset in TCQ
CTSMFID	SMF ID that created this record
CTCLASS	Process Session Class
CTGPFLAG	CTGPFLAG
CTSDCB	Source DSN DCB Attributes
CTSLRECL	SRC LRECL
CTSBLKSZ	SRC BLKSIZE
CTSRECFM	SEC RECFM
CTSDSORG	SRC DSORG
CTDDCB	Destination DSN DCB Attributes
CTDLRECL	Destination LRECL
CTDBLKSZ	Destination BLKSIZE
CTDRECFM	Destination RECFM
CTDDSORG	Destination DSORG

Field Names	Field Description
CTFASPPL	fasp.policy
CTFASPBW	fasp.bandwidth (bits, 1K= 1000 bits)
CTFASPFT	fasp.filesize.threshold (bytes/1024bytes)
CTFLAG17	CTFLAG17
CTFLAG18	CTFLAG18
CTXTIME	Process Stop Time
CTXDATE	Process Stop Date
CTPNRLS	Pnode C:D Version
CTSNRLS	Snode C:D Version

Delete Process Termination Record

The following table shows fields available in the Delete Process Termination record. The DGAADPTR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
DPRECLN	Length of this record.
DPRTYPE	Record type—DP indicates the Delete Process Termination record.
DPTIME	Time that DELeTe PROCess command completed in hh/mm/ss/tt format.
DPDATE	Date that DELeTe PROCess command completed in Julian date format (yyyymmdd).
DPPROCNM	Process name.
DPPROCNO	Process number.
DPSTEP	Step name or label.
DPUNODE	User node iD of the submitter.
DPUID	Userid of the submitter.
DPSTIME	Time that DELeTe PROCess command started in hh/mm/ss/tt format.
DPSPDATE	Date that DELeTe PROCess command started in Julian format (yyyymmdd).
DPSCC	Step completion code. Displays normal completion code.
DPMSGID	Message ID.
DPDPRNM	Name of the deleted Process.
DPDPRNOP	Number of the deleted Process.
DPRMNID	Route message node of deleted Process.
DPRMUID	Route message userid of deleted Process.

Display Statistics Record

The following table contains each field available in the Display Statistics record. The DGAADTR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Name	Field Description
DTRECLN	Length of this record.
DTRTYPE	Record type indicates specific data in statistics record: SP=SElect PROCess; DT=SElect TASK; FT=FLUSH TASK; SS=SElect STATistics; and SN=SElect NETMAP.
DTTIME	Time that command completed in hh/mm/ss/tt format.
DTDATE	Date that command completed in Julian date format (yyyydddf).
DTPROCNM	Process name.
DTPROCNO	Process number.
DTSTEP	Step name or label.
DTUNODE	User node ID of the submitter.
DTUID	Userid of the submitter.
DTSTIME	Time that command started in hh/mm/ss/tt format.
DTSDATE	Date that command started in Julian date format (yyyydddf).
DTSCC	Step completion code. Displays normal completion code.
DTMSGID	Message ID.
DTDFLNM	Name of the displayed data set.
DTNVTAMS	Number of VTAM sends.
DTNVTAMR	Number of VTAM receives.
DTGETS	Number of GETS.
DTPUTS	Number of PUTS.

Flush/Suspend Process Termination Record

The following table shows each field in the Flush Process/Suspend Process Termination record. The DGAAFPTR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
FPRECLN	Length of this record.
FPRTYPE	Record type—FP indicates the Flush Process Termination record.
FPTIME	Time that FLUSH PROCess command completed in hh/mm/ss/tt format.
FPDATE	Date that FLUSH PROCess command completed in Julian date format (yyyydddf).
FPPROCNM	Process name.
FPPROCNO	Process number.
FPSTEP	Step name or label.
FPUNODE	User node ID of the submitter.
FPUID	Userid of the submitter.
FPSTIME	Time that FLUSH PROCess command started in hh/mm/ss/tt format.
FPSDATE	Date that FLUSH PROCess command started in Julian date format (yyyydddf).
FPSCC	Step completion code. Displays normal completion code.

Field Names	Field Description
FPMSGID	Message ID.
FPDPRNM	Name of the flushed Process.
FPDPRNOP	Number of the flushed Process.
FPRMNID	Route message node of flushed Process.
FPRMUID	Route message userid of flushed Process.

PDS Member Copy Record

The following table shows each field available in the PDS Member Copy record. The DGAAFMCR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
MCRECLN	Length of this record.
MCRTYPE	Record type—MC indicates the PDS Member Copy record.
MCTIME	Time that COPY step completed in hh/mm/ss/tt format.
MCDATE	Date that COPY step completed in Julian date format (yyyydddf).
MCSTIME	Time that member processing started in hh/mm/ss/tt format.
MCSDATE	Date that member processing started in Julian date format (yyyydddf).
MCSCC	Step completion code. Displays normal completion code.
MCMSGID	Message ID.
MCPROCNM	Process name.
MCPROCNO	Process number.
MCSTEP	Step name or label.
MCUNODE	User node ID of the submitter.
MCUID	Userid of the submitter.
MCPNODE	The Pnode name for this Process.
MCSNODE	The Snode name for this Process.
MCNODE	This node is P(node) or S(node).
MCFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
MCTNAME	Name of member on destination PDS.
MCFNAME	Name of member on source PDS (if different).
MCANAME	Name of member for which node is an alias.
MCNRECS	The number of records.
MCNBLKS	The number of blocks.
MCSEQ	Sequence # of Date/Time
MCLNBYTS	Bytes of data moved
MCSERVER	Connect:Direct SYSPLEX SERVER NAME
MCFLAG1	Contains MCFXNODE flag

Field Names	Field Description
MCFLAG2	unused

Process Submit Statistics Record

The following table describes fields in the Process Submit Statistics record. The DGAAPPSR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
PSSRECLN	Length of this record.
PSSRTYPE	Record type indicates specific data in statistics record: PS=SUBmit statement and SW=SUBmit command within a Process.
PSSCTIME	Time that Process completed in hh/mm/ss/tt format.
PSSCDATE	Date that Process completed in Julian date format (yyyydddf).
PSSSTIME	Time that Process started in hh/mm/ss/tt format.
PSSSDATE	Date that Process started in Julian date format (yyyydddf).
PSSSCC	Step completion code. Displays normal completion code.
PSSMSGID	Message ID.
PSSPRCNM	Process name.
PSSPRCNO	Process number.
PSSSTEP	Step name or label.
PSSUNODE	Submitter's symbolic node name.
PSSUID	Userid of the submitter.
PSSPNODE	Pnode name for this Process.
PSSSNODE	Snode name for this Process.
PSSNODE	This node is P(node) or S(node).
PSSFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
PSSROSIZ	The number of bytes in the Process.
PSSRMNID	Route message userid of Process.
PSSFUNCD	The function code for the SUBMIT.
PSSSYNTAX	The syntax error in SUBmit command.
PSSPARSE	The parse error in SUBmit command.
PSSPACT#	Displacement to Pnode account data.
PSSSACT#	Displacement to Snode account data.
PSSDSN#	Displacement to data set name.
PSSSUBJOB	Submitter's jobname
PSSSUBJID	Submitter's jobid

Process Termination Record

The following table describes fields in the Process Termination record. The DGAAPTR member of the \$CD.SDGAMAP library contains the SAS maps of the IBM Connect:Direct statistics record.

Field Names	Field Description
PTRECLN	Length of this record.
PTRTYPE	Record type—PT indicates the Process Termination record.
PTTIME	Time that Process completed in hh/mm/ss/tt format.
PTDATE	Date that Process completed in Julian date format (yyyydddf).
PTSTIME	Time that Process started in hh/mm/ss/tt format.
PTSDATE	Date that Process started in Julian date format (yyyydddf)
PTSCC	Step completion code. Displays normal completion code.
PTMSGID	Message ID.
PTPROCNM	Process name.
PTPROCNO	Process number.
PTSTEP	Step name or label.
PTUNODE	Submitter's symbolic node name.
PTUID	Userid of the submitter.
PTPNODE	Pnode name for this Process.
PTSNODE	Snode name for this Process.
PTNODE	This node is P(node) or S(node).
PTFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
PTUNPRNO	Process number unique to submitter's node.
PTRMNID	Route message userid of Process.
PTRMUID	Userid where message is routed.
PTSUBTME	Time that Process was submitted in hh/mm/ss/tt format.
PTSUBDTE	Date that Process submitted in Julian date format (yyyydddf).
PTSCHTME	Time that Process was scheduled in hh/mm/ss/tt format.
PTSCHDTE	Date that Process was scheduled in Julian date format (yyyydddf).
PTSCHDAY	Day Process was scheduled.
PTSTMTN	Number of statements.
PTPRTY	Process selection priority.
PTSUBJOB	Submitter's jobname
PTSUBJID	Submitter's jobid

Run Job Termination Record

The following table describes fields in the Run Job Termination record. The DGAARJTR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
RJCLASS	Session class.
RJRECLN	Length of this record.
RJRTYPE	Record type–RJ indicates the Run Job Termination record.
RJTIME	Time that RUN JOB step completed in hh/mm/ss/tt format.
RJDATE	Date that RUN JOB step completed in Julian date format (yyyydddf).
RJSTIME	Time that RUN JOB step started in hh/mm/ss/tt format.
RJSDATE	Date that RUN JOB step started in Julian date format (yyyydddf).
RJSCC	Step completion code. Displays normal completion code.
RJMSGID	Message ID.
RJPROCNM	Process name.
RJPROCNO	Process number.
RJSTEP	Step name or label.
RJUNODE	Submitter's symbolic node name.
RJUID	Userid of the submitter.
RJPNODE	Pnode name for this Process.
RJSNODE	Snode name for this Process.
RJNODE	This node is P(node) or S(node).
RJFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
RJJOBNM	Job name.
RJJOBNO	Job number.
RJSYRCD	System return code.
RJDDSN	Displacement to data set name.
RJPACCT#	Displacement to Pnode account data.
RJSACCT#	Displacement to Snode account data.
RJSUBJOB	Submitter's jobname
RJSUBJID	Submitter's jobid

Run Task Termination Record

The following table shows each field in the Run Task Termination record. The DGAARTTR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
RTRECLN	Length of this record.
RTRTYPE	Record type–RT indicates the Run Task Termination record.
RTTIME	Time that RUN TASK step completed in hh/mm/ss/tt format.
RTDATE	Date that RUN TASK step completed in Julian date format (yyyydddf).
RTSTIME	Time that RUN TASK step started in hh/mm/ss/tt format.

Field Names	Field Description
RTSDATE	Date that RUN TASK step started in Julian date format (yyyydddf).
RTSCC	Step completion code. Displays normal completion code.
RTMSGID	Message ID.
RTPROCNM	Process name.
RTPROCNO	Process number.
RTSTEP	Step name or label.
RTUNODE	Submitter's symbolic node name.
RTUID	Userid of the submitter.
RTPNODE	Pnode name for this Process.
RTSNODE	Snode name for this Process.
RTNODE	This node is P(node) or S(node).
RTFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
RTMODNM	Name of the RUN TASK program.
RTSYSRTN	System return code.
RTPARMLN	Length of the parameter list.
RTPACCT#	Displacement to Pnode account data.
RTSACCT#	Displacement to Snode account data.
RTSUBJOB	Submitter's jobname
RTSUBJID	Submitter's jobid
RTCLASS	Session class

Signon/Signoff Statistics Record

The following table shows each field in the Signon/Signoff Statistics record. The DGAASFR member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
SIRECLN	Length of this record.
SIRTYPE	Record type indicates specific data in the statistics record: SI=SIGNON command and SO=SIGNOFF command.
SITIME	Time that SIGNON/SIGNOFF command completed in hh/mm/ss/tt format.
SIDATE	Date that SIGNON/SIGNOFF command completed in Julian date format (yyyydddf).
SIUNODE	User symbolic node ID.
SIUID	Userid of the submitter.
SISCC	Step completion code. Displays normal completion code.
SIMSGID	Message ID.

Start IBM Connect:Direct Command Record

The following table shows each field available in the Start IBM Connect:Direct Command record. The DGAASDCR member of the \$CD.SDGAMAP library contains the SAS maps of the IBM Connect:Direct statistics record.

Field Names	Field Description
SDRECLN	Length of this record.
SDRTYPE	Record type—SD indicates the Start IBM Connect:Direct Command record.
SDTIME	Time that Start IBM Connect:Direct command completed in hh/mm/ss/tt format.
SDDATE	Date that Start IBM Connect:Direct command completed in Julian date format (yyyydddf).
SDPROCNM	Process name.
SDPROCNO	Process number.
SDSTEP	Step name or label.
SDUNODE	Submitter's symbolic node name.
SDUID	Userid of the submitter.
SDSTIME	Time that Start IBM Connect:Direct command was issued in hh/mm/ss/tt format.
SDSDATE	Date the Start IBM Connect:Direct command was issued in Julian date format (yyyydddf).
SDSCC	Step completion code. Displays normal completion code.
SDMSGID	Message ID.

Stop IBM Connect:Direct Statistics Record

The following table shows each field available in the Stop Statistics record. The DGAASTDC member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
TDRECLN	Length of this record.
TDRTYPE	Record type—ST indicates the Stop IBM Connect:Direct Statistics record.
TDTIME	Time that STOP IBM Connect:Direct command completed in hh/mm/ss/tt format.
TDDATE	Date that STOP IBM Connect:Direct command completed in Julian date format (yyyydddf).
TDPROCNM	Process name.
TDPROCNO	Process number.
TDSTEP	Step name or label.
TDUNODE	User node of the submitter.
TDUID	Userid of the submitter.
TDSTIME	Time that STOP IBM Connect:Direct command was issued in hh/mm/ss/tt format.
TDSDATE	Date that STOP IBM Connect:Direct command was issued in Julian date format (yyyydddf).
TDSCC	Step completion code. Displays normal completion code.

Field Names	Field Description
TDSTOP	IBM Connect:Direct shutdown types: W=Step shutdown; X=Immediate shutdown; Y=Quiescent shutdown; and Z=Forced shutdown by an abend.
TDMSGID	Message ID.

Write to Operator (WTO) Statistics Record

The following table shows each field available in the WTO Statistics record. The DGAAWTOS member of the \$CD.SDGAMAP library contains the SAS maps of the statistics record.

Field Names	Field Description
WTRECLN	Length of this record.
WTRTYPE	Record type—WT indicates the WTO Statistics record.
WTCTIME	Time that Process completed in hh/mm/ss/tt format.
WTCDATE	Date that Process completed in Julian date format (yyyydddf).
WTSTIME	Time that function started in hh/mm/ss/tt format.
WTSDATE	Date that function started in Julian date format (yyyydddf).
WTSCC	Step completion code. Displays normal completion code.
WTMSGID	Message ID.
WTPROCNM	Process name.
WTPROCNO	Process number.
WTSTEP	Step name or label.
WTUNODE	Submitter's symbolic node name.
WTUID	Userid of the submitter.
WTPNODE	The Pnode name for this Process.
WTSNODE	The Snode name for this Process.
WTNODE	This node is P(node) or S(node).
WTFROM	Direction of data: P(node) to Snode or S(node) to Pnode.
WTTASKNO	Task number that issued the WTO.

The Operator Interface

The Operator Interface enables you to issue Connect:Direct for z/OS commands from a z/OS console using a MODIFY command.

Access the Operator Interface after IBM Connect:Direct initializes if you specify the initialization parameters MCS.CLIST and MCS.SIGNON.

Following is the syntax for the MCS.CLIST and MCS.SIGNON initialization parameters:

```
MCS.CLIST=console operator's CLIST library (dsn)
MCS.SIGNON=(SIGNON USERID=(userid,password) -
CASE=YES | NO -
NETMAP=network map file name)
```

An automatic signon is issued after the first IBM Connect:Direct CLIST is invoked using the information in MCS.SIGNON. This session is active until you submit a CLIST that contains a IBM Connect:Direct SIGNOFF command.

Note: The NETMAP used by the MCS.SIGNON must be the same as the NETDSN in the INITPARMs for this DTF. If a different NETMAP is used, LOCAL.NODE information from that NETMAP will be used for a SIGNON, and the commands to this DTF will be executed on the other DTF if it is running.

You can create easy to remember customized IBM Connect:Direct commands with the Operator Interface through a CLIST-type facility. The software supports symbolic substitution and CLIST-type parameters allowing you to alter IBM Connect:Direct commands without changing the CLIST.

In a IBM Connect:Direct/Plex, if you issue a console operator command to a IBM Connect:Direct/Server, the console interface actually signs on to the IBM Connect:Direct/Manager. As a result, any commands issued to IBM Connect:Direct/Server are actually issued to the IBM Connect:Direct/Manager, which is the only IBM Connect:Direct/Plex member that accepts operator commands.

Concatenation of Operator CLIST

Currently, the OPLIST can only be defined in the initialization parameter, MCS . CLIST. This does not allow the user to concatenate their own OPLIST.

The current syntax for the initialization parameter (initparm) is

```
MCS.CLIST = oplist_dsn
```

The MCS . CLIST and MCS . SIGNON syntax remains unchanged. However, if the startup JCL contains the DD statement DGAOPLS, IBM Connect:Direct builds and saves the dsn list from the JCL instead of using the MCS . CLIST parameter. The JCL statement provides an override of the initialization parameter.

```
//DGAOPLS DD DSN=oplist_dsn_1  
//DD DSN=oplist_dsn_2  
//DD DSN=oplist_dsn_n
```

Note: If the IBM Connect:Direct Server initialization finds the DGAOPLS DD allocated, the MCS . CLIST initialization parameter is ignored. However, the MCS . SIGNON initialization parameter is still required.

Sample Connect:Direct for z/OS CLISTs

The IBM Connect:Direct sample CLIST library, SDGAOPLS, contains the sample CLISTs included in the following table. Use these CLISTs to build customized IBM Connect:Direct commands with symbolic parameters that allow you to customize the CLIST at submission time. Sample operator commands for each CLIST are listed in the comment section.

CLIST Name	Command	Description
DGAOCEV	SECURE CERTCK	Execute the Secure Certck. Performs the Certificate Expiration validation.
DGAOCRF	SECURE REFRESH	Execute the Secure refresh. Refreshes the SecurePlus SSL/TLS environment.
DGAOCMD	CMD	Provides a general format for issuing a IBM Connect:Direct command.
DGAODP	DELETE PROCESS	Shows six ways to delete a Process using this CLIST.
DGAODUMP	STOP CD (Force)	Stops IBM Connect:Direct using the FORCE parameter and produces abnormal ending (abend) 4095 and a dump.

CLIST Name	Command	Description
DGAOIPRM	DISPLAY INITPARMS	Displays initialization parameters to the operator console. In a PLEX environment, it works as follows: <ul style="list-style-type: none"> • Displays the global initialization parameters • Defaults to the manager's local initialization parameters • Allows the name of a server to display the server's global and local initialization parameters
DGAOMSG	SELECT MSG	Shows a detail of the IBM Connect:Direct message requested.
DGAONM	SELECT NETMAP	Displays entire NETMAP or it allows for one parameter – a node name.
DGAORLSE	CHANGE PROCESS	Shows three ways to release a Process.
DGAOSOFF	SIGNOFF	Signs the operator off IBM Connect:Direct.
DGAOSON	SIGNON	Connects the operator to IBM Connect:Direct.
DGAOSP	SELECT PROCESS	Shows six ways to display Process status information.
DGAOSS	SELECT STATISTICS	Shows seven ways to select statistics for display.
DGAOSTAT	SELECT STATISTICS	Shows three ways to select statistics for display covering a specific time period.
DGAOSTOP	STOP CD	Stops IBM Connect:Direct using the STOP IMMEDIATE command. To perform an immediate shutdown in a PLEX environment from the operator console, issue the following command: F CD,STOP CDPLEX. To stop a specific server, use this command: F CD,STOP WHERE(SERVER=name)
DGAOSUB	SUBMIT PROCESS	Shows two ways to submit a Process.
DGAOSUSP	SUSPEND PROCESS	Shows four ways to suspend a Process.
DGAOSWAP	SWAP NODE	Swaps operator to another node in a multiple session environment.
DGAOTS	SELECT TASK	Shows active tasks.
DGAOVP	VIEW PROCESS	Displays a specific Process.

Rules for Setting Up Connect:Direct for z/OS CLISTs

The following rules apply when setting up an operator CLIST:

- All operator CLISTs must have a PROC record as the first record in the CLIST. The PROC record defines the parameters and keywords that are passed to the CLIST.
- You can stack multiple IBM Connect:Direct commands in one CLIST, but you only need one PROC statement.
- A number between 0 and 24 (inclusive) must follow the PROC identifier. This number indicates the number of positional parameters used by the CLIST.
- Parameter names (one to eight characters each) that correspond to each positional parameter follow the number. If no positional parameters exist, specify PROC 0.
- If you define more than one positional parameter on the PROC statement, you do not need to specify trailing positional parameters in the command unless you also specify a keyword parameter.
- Use commas to indicate null values.

- A positional parameter is terminated by the first blank encountered.
- You can specify optional keyword parameters after the positional parameters. Enclose default values in parentheses after each keyword name.
- Use a hyphen (-) to indicate that a IBM Connect:Direct command continues on the next line. An example follows.

```
PROC 1 PNUM
SELECT PROCESS WHERE (
                        PNUM=&PNUM)      -
```

- You can use comments in the CLIST only if you include an asterisk (*) in the first column.
- Sequence numbers are not allowed.

Connect:Direct for z/OS Commands

Submit the IBM Connect:Direct commands to the Operator Interface using the MODIFY command interface. This operator command lets you submit a CLIST containing IBM Connect:Direct commands, and modify the IBM Connect:Direct commands by substituting symbolic parameters with real values.

The command format is:

```
F jobname,clist [options]
```

Following is a description of the parameters:

Parameter	Description
F	The short form of the Administrator MODIFY command.
jobname	The name on the job statement of the job stream that brings up the DTF. Caution: If jobname is a IBM Connect:Direct/Server, the command is redirected to the IBM Connect:Direct/Manager.
clist	The name of the CLIST containing the IBM Connect:Direct command.
options	Optional positional and keyword parameters to be passed to the CLIST.

CLIST Examples

The following are CLIST examples and operator commands that submit the CLIST examples. As some examples show, you can use symbolic parameters to modify the IBM Connect:Direct command at the time of submission. For example purposes, CDDTF is the jobname.

CLIST with Command and No Parameters

The following CLIST SELX command contains one command and no parameters:

```
PROC 0 SEL PROC WHERE (QUEUE=EXEC)
```

To execute SELX, type the following operator command.

```
F CDDTF,SELX
```

When you issue the operator command, a Select Process command for Processes on the executing queue initiates. As a result, a list of these Processes is displayed.

CLIST with Command and One Parameter

In this example, the CLIST SELQ contains one command and one positional parameter (SUBD).

```
PROC 1 SUBD SEL PROC WHERE(QUEUE=&SUBD)
```

The following options present two ways to submit SELQ:

- When you issue the following command, TIMER (timer queue) is substituted for &SUBD. This command displays all Processes on the timer queue.

```
F CDDTF,SELQ TIMER
```

- When you issue the following command, ALL is substituted for &SUBD in the CLIST SELQ. This command displays all Processes on all queues.

```
F CDDTF,SELQ ALL
```

CLIST with Command and Multiple Parameters

In this example, the CLIST SUB contains one command, two positional parameters (PROC and PARMS), and one keyword parameter (SNODE).

```
PROC 2 PROC PARMS SNODE(OTHER.NODE) SUB PROC=&PROC SNODE=&SNODE &PARMS
```

The following options present four ways to execute SUB:

- When you issue the following command, PAYROLL is substituted for &PROC. This substitution causes the Process PAYROLL to be submitted.

```
F CDDTF,SUB PAYROLL
```

- Because no SNODE is specified, the SNODE defaults to OTHER.NODE.
- When you issue the following command, CHECKS is substituted for &PROC and HOLD=YES is substituted for &PARMS. These cause the Process CHECKS to be submitted as a held Process.

```
F CDDTF,SUB CHECKS HOLD=YES
```

Because no SNODE is specified, the SNODE defaults to OTHER.NODE.

- When you issue this command, ORDERS is substituted for &PROC and CD.CHICAGO is substituted for &SNODE. These cause the Process ORDERS to be submitted to run in session with the secondary node called CD.CHICAGO.

```
F CDDTF,SUB ORDERS,,SNODE=(CD.CHICAGO)
```

Because PROC and PARMS are positional parameters, commas are required.

- When you issue the following command, CONFIRM is substituted for &PROC and HOLD=NO,STARTT=(FRIDAY,NOON) is substituted for &PARMS. These submit the Process CONFIRM so that it runs at noon on Friday.

```
F CDDTF,SUB CONFIRM HOLD=NO,STARTT=(FRIDAY,NOON)
```

The SNODE defaults to OTHER.NODE because no SNODE is specified.

CLIST with Command and %IF %ELSE %EIF

In this example, the CLIST NM contains 1 command and 1 positional parameter.

```
PROC 1 NODE
  %IF &NODE = ,
  SEL NETMAP WHERE (NODE=*)
  %ELSE
  SEL NETMAP WHERE (NODE=&NODE)
  %EIF
```

If you enter no operands it displays the entire netmap. If you enter a node it displays that node only.

Operation Messages

IBM Connect:Direct operation diagnostic messages are formatted into multiple lines so that errors and exception conditions are easy to read. Operator messages are sent to:

- The route code given in the IBM Connect:Direct initialization parameters
- The ddname NDMLOG, if it is allocated

The COPY termination message (SVTM052I) is displayed in four lines, as follows:

- The first line shows the step label, the operation (COPY), the Process name and number, the associated node name, and the session class.
- The second line shows the FROM file name.
- The third line shows the TO file name.
- The fourth line shows the completion code and message ID. If the COPY does not complete successfully, the first four positions of the last line contain the “####” flag.

When session errors occur, the error message is formatted into two lines, as follows:

- The first line gives as much information as possible in the same format as the COPY termination message (Process name and number, and associated node name).
- The second line gives the text of the error message. The first four positions of the second line contain the “****” flag.

The following example shows two IBM Connect:Direct operation error messages following a COPY termination message.

```
SVTM052I STEP1 COPY PROCESS1( 126)PNODE=CD.NODE.A (002)
FROM SYSA.SAM.DATA.SET
TO SYSB.TEST.SAM.DATA.SET
#### COMPLETED 00000010/SCPA010I
SVTM045I PROCESS1( 126)PNODE=CD.NODE.A
**** RPLERRCK:CD/CD SESSION FAILURE
SVTM050I PROCESS1( 126)PNODE=CD.NODE.A
**** PROCESS INTERRUPTED:RECOVERY INITIATED
```

Stopping IBM Connect:Direct

To stop Connect:Direct for z/OS, use the following command, which executes the STOP CLIST using the Operator Interface.

```
F jobname,STOP
```

Note: The default value for the STOP CLIST is Immediate or I.

Following is a description of the parameters:

Parameter	Description
F	The short form of the Administrator MODIFY command.
jobname	The name on the job statement of the job stream that brings up the DTF. Caution: If jobname is a IBM Connect:Direct/Server, the command is redirected to the IBM Connect:Direct/Manager.
clist	The name of the CLIST containing the IBM Connect:Direct command.
options	Optional positional and keyword parameters to be passed to the CLIST.

Stopping a IBM Connect:Direct/Plex Manager and All Servers

To stop a IBM Connect:Direct/Plex from the console, use the following command which executes the STOP CLIST using the Operator Interface:

```
F jobname,STOP [Q|S|I|R|F,] CDPLEX [,RECOVER]
```

Stopping an Individual IBM Connect:Direct/Plex Server

To stop a IBM Connect:Direct/Plex Server from the console, use the following command which executes the STOP CLIST using the Operator Interface:

```
F jobname,STOP [Q|S|I|R|F,] WHERE(SERVER=name) [,RECOVER]
```

About Tape Mount Messages

Connect:Direct for z/OS supports the transmission of tape files. IBM Connect:Direct implements separate and distinct messages to control tape management, providing the greatest flexibility to operators in managing tape transfers.

Tape Pre-mount Message

This optional message is issued prior to allocating a tape device. It tells you that IBM Connect:Direct requires a specific volume on a specific device type. With this message, you can:

- Control the number of tape units available to Connect:Direct for tape transfers.
- Locate the volumes before the transfer occurs.

You are required to respond to this message when the required resources are available.

Tape Mount Messages

These messages are also optional. The first message (WTO) is in the format of the z/OS mount message. The second message is issued as a WTOR to inform the operator to mount a specific volume on a specific device. The third message is issued only if TAPE.PREMount=LIST. Message SVST00C is displayed listing all volume-serial numbers of the requested file. If you are unable to satisfy the specific request, you can reply CANCEL to cancel the request. This message is also used to drive the visual display devices attached to 3480-type devices.

Note: If you use tape silos in your environment, search for Tapemount Exit. To prevent the IBM Connect:Direct environment from locking up due to an outstanding tape mount request, the IBM Connect:Direct tapemount exit provides an interface to StorageTek Tape Silo Software to query the silo to determine that all VOLS required for a tape file are present.

Setting Up Connect:Direct for z/OS Tape Pre-mount Messages

Procedure

- To issue the IBM Connect:Direct tape pre-mount messages, specify TAPE.PREMOUNT=YES|LIST in the IBM Connect:Direct initialization parameter file.

Setting Up Connect:Direct for z/OS Tape Mount Messages

- Issue the IBM Connect:Direct tape mount messages in place of the z/OS mount message. The IBM Connect:Direct mount messages consist of a z/OS-format mount message followed by a WTOR issued to the descriptor and route codes specified by the DESC.TAPE and ROUTCDE.TAPE.
- You can suppress the IBM Connect:Direct message by specifying ROUTCDE.TAPE=(0) in the initialization parameters. The defaults are DESC.TAPE=(2) ROUTCDE.TAPE=(5,11). If you suppress IBM Connect:Direct tape mount messages, the standard z/OS mount messages are issued with related serialization during mount processing.
- If you specify ROUTCDE.TAPE=(0), causing IBM Connect:Direct not to issue the mount messages, then the normal z/OS mount message is issued when the dataset is opened. The z/OS mount processing holds an ENQ on the SYSZTIOT resource. This ENQ causes all other IBM Connect:Direct Processes to hang at allocation, open, and deallocation until the tape is mounted.
- If you use the IBM Connect:Direct tape mount message, the tape already is on the tape drive at open time and z/OS mount processing does not hold the SYSZTIOT resource.

Connect:Direct for z/OS Tape Pre-mount Messages

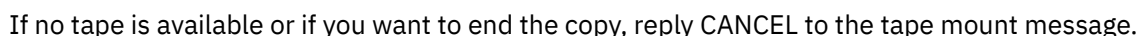
Following is the format of a IBM Connect:Direct tape pre-mount message.

```
SVST000B - C:D REQUIRES VOL=SCRCH TYPE=TAPE 6250/1600BPI  
REPLY "GO" WHEN READY
```

To continue processing, reply GO to the message. While this message is outstanding, no unit is allocated. However, the session for this transfer request remains active.

Connect:Direct for z/OS Tape Mount Messages

Following is the format of a IBM Connect:Direct tape mount message issued during a specific tape request.



After the GO reply to a pre-mount message, and following the pre-mount processing, IBM Connect:Direct allocates a tape device using Defer mounting. If an allocation error occurs, and the allocation error code is in the ALLOC.CODES parameter of the IBM Connect:Direct initialization parameters, IBM Connect:Direct retries the operation.

IEC501A M, cuu, volser, sl, , jobn, dsn, pnum, pnam, step

A COPY step within a Process requiring a tape to be mounted causes the tape mount message to display. You can terminate the COPY step with a CANCEL reply. Any other reply causes IBM Connect:Direct to reissue the mount message.

For standard label tapes, Connect:Direct for z/OS verifies the volume serial number for a specific volume request. If you mount the wrong volume, the following message is issued and the volume is dismounted.

Mount the correct volume, or reply CANCEL to the reissued SVST000A mount message.

Handling Multivolume Files

The Connect:Direct for z/OS mount message is issued only for the first volume of a file, when a multivolume file is requested. Subsequent mount requests causes a SYSZTIOT enqueue during end-of-volume processing.

To keep IBM Connect:Direct from long waits while a second or subsequent volume is mounted, use UNIT=(unit,P) or UNIT=(unit,n) for the UNIT keyword of the COPY statement.

Connect:Direct for z/OS Messages on 3480 Display

The 3480 tape drive has an 8-character display. IBM Connect:Direct issues messages to this display concerning tape processing in the following manner:

- When a tape is to be dismounted, the first character position of the display contains a D. Positions 2–7 consist of the tape's volume serial number. Position eight contains an N if the tape is nonlabeled or an S if the tape has standard labels.
- When a tape is to be mounted, the first character position of the display contains an M. Positions 2–7 consist of the tape's volume serial number. Position eight contains an N if the tape is nonlabeled or an S if the tape has standard labels.
- When a tape is loaded, the first character position of the display is blank. Positions 2–7 consist of the tape's volume serial number. Position eight contains an N if the tape is nonlabeled or an S if the tape has standard labels.

Event Services Support

IBM Connect:Direct Event Services Support (ESS) implements an asynchronous, event-generation facility in IBM Connect:Direct. ESS is designed for use by external management and automated operations applications that require real-time notification of IBM Connect:Direct activities. ESS supports a publish or subscribe protocol with optional guaranteed event delivery and automated recovery.

Customer applications register interest in receiving event data, and specify the types of event data the application receives through the new event services commands. ESS supports a user exit point and an optional API for interface communication with user-written CICS automation applications.

ESS Concepts and Components

ESS provides real-time delivery of IBM Connect:Direct event records to interested application interfaces. ESS is based on the existing IBM Connect:Direct statistics facility and is designed to:

- Minimize the server-performance overhead for ESS processing
- Prevent adverse impact on IBM Connect:Direct server reliability

ESS is a low-level IBM Connect:Direct server technology that enables you to layer multiple APIs. An end-user application interfaces with the API rather than the low-level ESS core technology. This design enables future extension of ESS to support multiple application interface protocols and publishing of the same event data to several event consumer applications.

System Interfaces

ESS supports the event services exit and the CICS API interface.

An application can access event data through an ESS exit. A sample event exit (DGAXEV01) is in the sample library, and writes the event data to a predefined data set.

A CICS API initiates event processing by signing on to IBM Connect:Direct through the current CICS API and issuing ESS commands. The CICS API provided with IBM Connect:Direct writes event data to a CICS Transient Data Queue (TDQ). A sample program, DGAQ249, shows how to read an event record from the TDQ.

System Advantages

Building on the existing statistics facility has the following advantages:

- Event Definition

Because ESS event records are IBM Connect:Direct statistics records, ESS does not restrict events to abnormal activities. This design enables ESS-enabled applications to initiate procedures based on either successful or unsuccessful Process statements.

- Historical Data Recovery

The IBM Connect:Direct statistics file functions as the historical repository of ESS event data. ESS supports optional automated recovery of past event data if an event consumer application fails. Upon restart, ESS requests old event records from the statistics file.

The API automatically manages the restart point for an ESS-enabled application. This API tracks the last successfully delivered event record to a designated subscriber. Upon restart, the API resynchronizes event processing by issuing a SElect STATistics command to retrieve the historical event data and deliver it to the restarted event consumer.

- Event Record Types

The event record types give external applications greater visibility into IBM Connect:Direct activities and improve the reliability and responsiveness of interface applications in a production environment. ESS supports several IBM Connect:Direct event record types, including:

- Process Initialization (PI) occurs immediately prior to the execution of the first Process statement in a Process.
- Step Start (CI, JI, TI) events occur immediately prior to the execution of a COPY, RUN JOB, or RUN TASK statement.
- COPY Step I/O Start (CE) occurs just before the first COPY step I/O in a Process.
- Queue Change (QW, QH, QE, QT) events occur whenever a Process moves from one logical queue to another in the TCQ. The sample initialization parameters provided in the base IBM Connect:Direct install the \$CD.SDGAPARM data set and specify STAT.EXCLUDE=(QE, QH, QT, QW). This specification prevents IBM Connect:Direct from producing the queue change records. If you want queue change records, you must set this parameter appropriately.

ESS also supports basic event filtering. When an event consumer application registers interest in receiving events, the application can restrict the types of event data that it wants to receive. These options support the same filtering syntax as the SElect STATistics command. Following are two possible selections:

- A fault management application can request notification of COPY requests that did not complete normally, QH (Queue Hold) events or both. When such an event occurs, the fault management application can then poll IBM Connect:Direct through SElect STATistics or SElect PROCess to investigate the problem further.
- An application registers interest in all events for a small group of mission-critical Processes.

Event Services Support Architecture

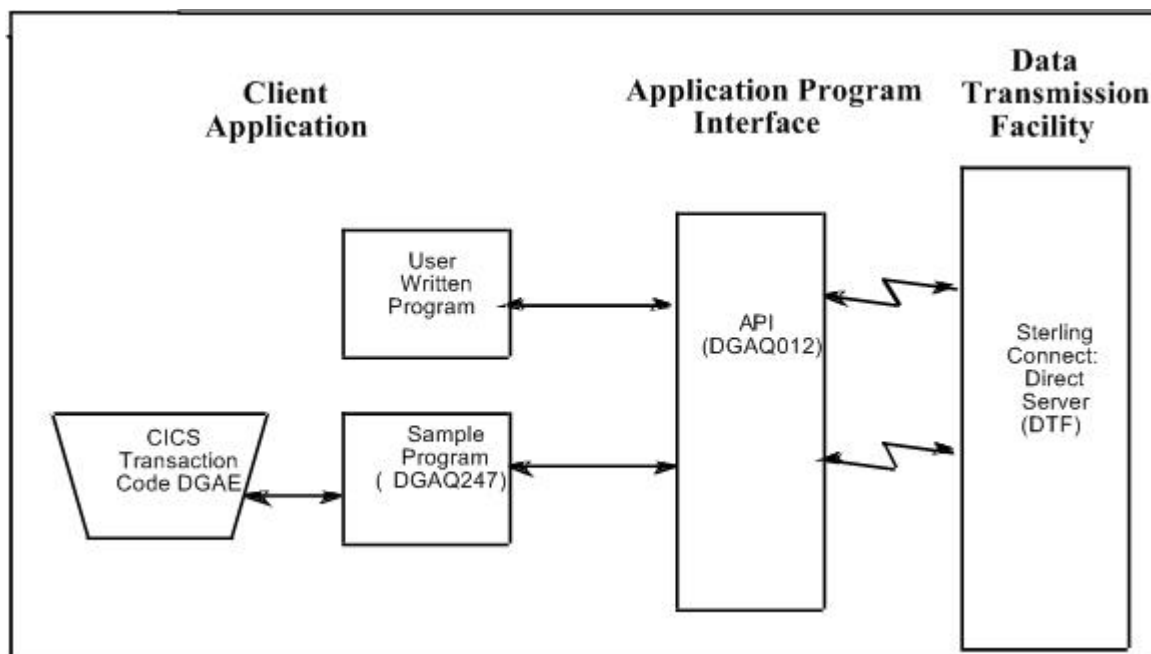
ESS uses three components to provide real-time information about IBM Connect:Direct activities:

- Client application
- API
- IBM Connect:Direct server

ESS communicates with client application interfaces through the CICS API or an ESS user exit.

The components interact to accomplish the following tasks:

- Define which events are reported to the client application
- Turn event services requests on and off
- Register event data



ESS Data Flow Using CICS API

Define the event services environment by specifying the types of event data your application receives. These definitions are the event data requests that you issue through the DGAQ247 program or a user-written program.

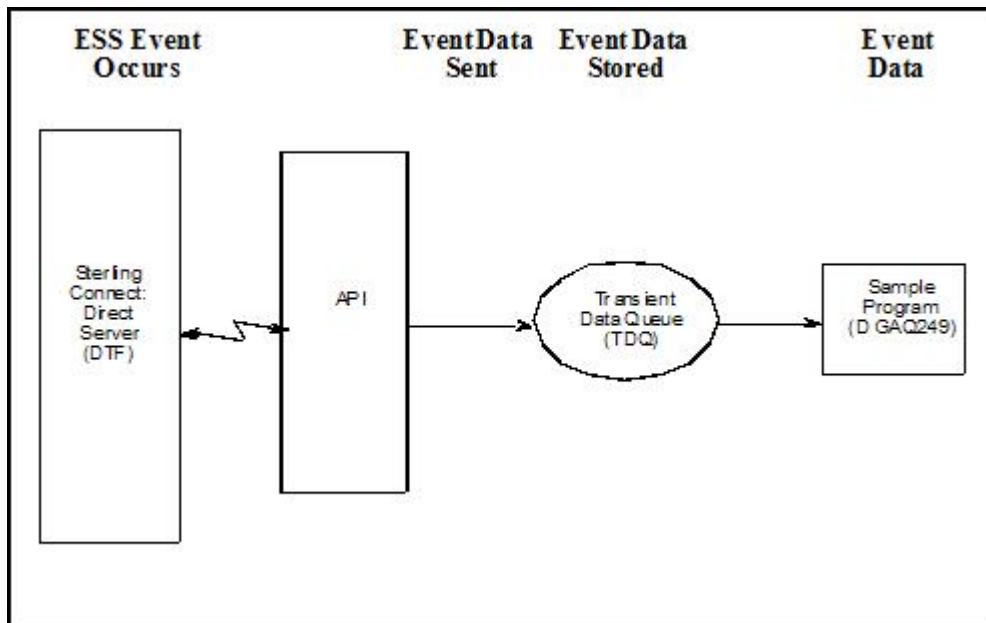
After you define the event data requests for the client application and activate the request by issuing the EVENT SERVICES START command, ESS processes the event data as follows:

- Upon receiving the EVENT SERVICES START ID parameter, the API (DGAQ012) passes the command to the DTF through a VTAM session.

When a requested event occurs, the DTF sends the event data to the API for distribution to the Transient Data Queue (TDQ).

- The event data is then available to the client application through the DGAQ249 Sample Program. The DGAQ249 program provides a way to read event data from the TDQ and processes the data for use by your application.

The following diagram illustrates the flow of event data from the DTF to the client application:



Deciding What Event Data to Collect

Deciding what event data to collect requires that you understand the types of data the ESS delivers and how you can use this data in your event consumer application. Although it is technically possible to request all event data for all Processes in the system, it is usually unnecessary and impractical to do so. Typically, an event consumer application is interested in a subset of event data.

Note: Statistics records excluded through the API STAT command or a DTF statistics exit, or by coding the DTF STAT.EXCLUDE initialization parameter, are not available to ESS.

Following are commonly requested event data:

- All event data for a Process or group of Processes

Requesting all event data provides the most detailed trace of Process execution. This information is useful when testing an event consumer application. However, it is unnecessary after an application goes into production.

- Major event data for a Process or group of Processes

This request is the most common subset specification. For normal production work, this request gives a sufficient trace of Process execution, and minimizes overhead for ESS processing and event records by your application.

- Subset of event data for a particular Process, set of Processes, or all Processes

For example, an application executes a procedure when a IBM Connect:Direct Process terminates. In this case, it may be sufficient to register interest in PT (Process Termination) records only. As a second example, a fault reporting application needs CT (Copy Termination) statements that did not complete normally (signalling the failure of a IBM Connect:Direct file transfer request).

ESS Event Record Examples

The following examples show the ESS event records from a single Copy Process. The examples show the records produced from Process submittal until normal Process termination. The examples assume that no IBM Connect:Direct commands affecting the Process [including the SElect PROCess command] are issued between Process submittal and termination.

ESS Event Records Example 1 (No Retries)

In this example, the Process does not undergo a Process retry for any reason. The sequence of ESS event records produced by this Process are listed in the following table:

Record Type	Record Description
PS	Process submit
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step statement start
CE	COPY step I/O start
CT	COPY termination
NL	Last statement (null step 1)
PT	Process termination

ESS Event Records Example 2 (One Retry)

This example shows the sequence of ESS records that are produced if the same Process undergoes a single Process retry. IBM Connect:Direct commands are deliberately issued to suspend the Process and then release it for retry. As a result, the following record sequence includes the various operator commands required. The events are in sequential order.

Record Type	Record Description
PS	Process submit
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step statement start
CE	COPY step I/O start
FS	Suspend Process command issued
CT	COPY termination (as a result of requested SUSPEND)
QH	Process moved to HOLD queue
CH	Change Process command issued with RELEASE=Yes
QE	Process moved to EXEC queue
PI	Process initiation
CI	COPY step start
CE	COPY I/O start
CT	COPY termination
NL	Last statement in Process end
PT	Process termination

Using ESS with the CICS API

The CICS API writes event data to the Transient Data Queue (TDQ). Your CICS application must include a program to read these event records from the TDQ and send them for processing..

About this task

To enable a CICS application for ESS:



Warning: Automated Connect:Direct CICS tasks that are repeated and issue a SIGNON without a SIGNOFF can consume memory and render the interface inactive.

Procedure

1. Decide what event data the application requires.
2. Verify that the Connect:Direct CICS API software is installed and working. Refer to the Connect:Direct installation documentation for information.
3. Create an ESS registration program. Include this program in the startup portion of your application. It must be called whenever your application wants to initiate event record collection. This program must perform the following:
 - Sign on to the Connect:Direct CICS API.
 - Issue the EVENT SERVICES CREATE command to register interest in receiving event data and to specify event record filter criteria.
 - Specify the ID parameter to identify a subscriber for your application. Specify the WHERE and ORWHERE parameters for filtering criteria.
 - Issue the EVENT SERVICES START command and specify the ID parameter to initiate event reception.
4. Create a customer interface program to read event records from the CICS TDQ and send them for processing. The sample interface program (DGAQ249) demonstrates how to read an event record from the TDQ. The application must issue an EXEC CICS READQ TD QUEUE instruction.
5. Modify the application to process the ESS event records.
6. Embed a call to the EVENT SERVICES STOP ID=subscriber_name parameter in the location to stop collecting event data.

Using ESS with the ESS User Exit

You can execute Event Services commands through any existing IBM Connect:Direct API such as CICS, ISPF, and the batch interface. However, not all IBM Connect:Direct APIs can process synchronous event data received from an EVENT SERVICES START command. In some cases, it is preferable not to have the IBM Connect:Direct API process synchronous event data. The event services exit provides an alternative. Rather than passing synchronous event data back to the API that issued the EVENT SERVICES START (ESS) command, an exit processes the event data.

The ESS exit acts as an extension to the IBM Connect:Direct server. When a synchronous event occurs, IBM Connect:Direct calls the event services exit. An event exit control block is passed for each event and contains the following:

- Address of the event record
- Flags indicating first or last call
- Fields for the user to indicate return code and message

DGA\$EVCB, found in \$CD.SDGAMAC, maps the event exit control block. The sample exit, DGAXEV01, is in \$CD.SDGASAMP.

The event services exit is called as a subtask from IBM Connect:Direct so that the user code waiting for system services does not directly effect IBM Connect:Direct processing. The sample exit (DGAXEV01) writes each event record to a predefined data set. You must modify DGAXEV01 to specify the name of

your event exit data set, and you must define the data set to accommodate records up to 2048 bytes in length.

Using the ESS Exit

About this task

Each record passed to the exit consists of a record header and an event record. DGA\$NHDR, in \$CD.SDGAMAC, maps the record header. Other members in \$CD.SDGAMAC map the event records. See [Event Services Record](#), to determine which member maps a particular event record.

The exit must reside in a load library accessible to the IBM Connect:Direct server (DTF). This exit accommodates systems where I/O calls must be done below the line.

To use the ESS user exit:

Procedure

1. Create an ESS exit program that can process the requested event records.
2. Decide what event data your application requires.
3. Verify that your software is correctly installed and working.
4. Create an ESS registration program. Include this program in the startup portion of your application. It is called whenever your application initiates event record collection. This program must accomplish the following tasks:
 - Signon to IBM Connect:Direct through the CICS API, the ISPF IUI, or the batch interface.
 - Issue the EVENT SERVICES CREATE command to register interest in receiving event data and specify event record filter criteria.
 - Specify the EXIT parameter to identify the name of your exit program, and the ID parameter to identify a subscriber name for your application. Specify the WHERE and ORWHERE parameters for your filtering criteria. Refer to EVENT SERVICES CREATE Command Format for more information about event filter criteria.
 - Issue the EVENT SERVICES START command and specify the ID parameter to initiate event reception.

Embed a call to the EVENT SERVICES STOP ID=subscriber_name parameter in your application where you want to stop collecting event data. If the user application does not process records fast enough, the internal queue, which holds asynchronous events, becomes full. When the queue is full, event processing for this request ends, and Connect:Direct writes a WTOR message. To customize the internal queue size, specify the MAXQCNT parameter.

Issuing Event Services Commands

EVENT SERVICES CREATE Command Format

The EVENT SERVICES CREATE command enables you to create a new request for event data and define which data is sent to your application. The EVENT SERVICES CREATE command uses the following format and parameters:

Command	Parameters
EVENT [SERVICES] CREATE	ID=event-request-name
	WHERE (CCODE PNAME PNUMber STARTT STOPT USER SNODE TYPE FNAME CASE LASTSEQ MAXQCNT EXIT)
	ORWHERE =(CCODE PNAME PNUMber USER SNODE TYPE FNAME)

Required Parameters

The EVENT SERVICES CREATE command has two required parameters.

ID=event-request-name

specifies a logical name for the event services request. This name is a text string of 1–16 characters. You can use any printable characters. You cannot use embedded blanks.

WHERE (CCODE = (condition, completion code) | PNAME = name | (list)

PNUMBER = number | (list)

STARTT = ([date | day] [,hh:mm:ssXM])

STOPT = ([date | day] [,hh:mm:ssXM])

USER = name | (list)

SNODE = name | (list)

TYPE = id | (list)

FNAME = dsname | (list)

CASE = YES | NO

LASTSEQ=n

MXQCNT=n

EXIT=exitname)

specifies selection criteria for event records.

You must specify at least one WHERE () subparameter, such as CCODE, PNAME, PNUMBER, or TYPE.

CCODE = (condition, completion code)

specifies selection by completion code.

condition specifies a relationship to the completion code given in the subsequent positional parameter. The options for specifying condition are:

GT greater than LT less than EQ equal to NE not equal to GE greater than or equal to LE less than or equal to

completion specifies a completion code value ranging from 1 to 2,147,483,647 so the RUN TASK can pass all values.

For example, if you specify CCODE = (GT,0), you see event records in which the step completion code is greater than zero, as long as the records also meet other specified criteria.

PNAME = name | (list)

specifies selection by Process name. Specify a list of Processes by enclosing them in parentheses. You can use a wild card character (*) at the end of the name.

For example, if you specify PNAME=TEST*, then all records with TEST in the first four characters of the Process name field are selected. Records having TEST, TEST123, and TESTX all satisfy the selection criterion.

name specifies the name of the Process to select.

(list) specifies a list of Process names to select. Enclose the list in parentheses. Separate Processes in the list with commas.

PNUMBER = number | (list)

specifies selection by Process number. To request a list of Processes, enclose them in parentheses. The range is 1–99999.

number specifies the number of the Process to select.

STARTT = ([date | day] [,hh:mm:ssXM])

specifies selection by start date and time. Specify STARTT as a date and time prior to the current time. If you set this parameter, the event records retrieved from the statistics file begin with the date and time you specified and continue until the current time is reached. At that point, event records are processed as they occur.

Note: If you specify STARTT and LASTSEQ with the EVENT SERVICES START command, those values replace the values specified for these parameters in the EVENT SERVICES CREATE command. When you use the time specification in conjunction with the LASTSEQ parameter, you must include hours, minutes, seconds, and hundredths of seconds in the format hh:mm:ss.th.

date specifies the starting date from which event records are retrieved from the statistics file. You can specify the date in either Gregorian or Julian format.

If you use a Gregorian date format, set the DATEFORM initialization parameter to the appropriate date format. Otherwise, the date format defaults to the platform date format. You do not need to set the DATEFORM parameter for Julian date format.

The following table shows the acceptable date formats:

Format	DATEFORM Parameter	Date Format
Gregorian	DATEFORM=MDY	mmddy or mmddy mm/dd/yy or mm/dd/yyyy mm.dd.yy or mm.dd.yyyy
Gregorian	DATEFORM=DMY	ddmmy or ddmmy dd/mm/yy or dd/mm/yyyy dd.mm.yy or dd.mm.yyyy
Gregorian	DATEFORM=YMD	yymmdd or yyyymmdd yy/mm/dd or yyyy/mm/dd yy.mm.dd or yyyy.mm.dd
Gregorian	DATEFORM=YDM	yyddmm or yyyyddmm yy/dd/mm or yyyy/dd/mm yy.dd.mm or yyyy.dd.mm
Julian	N/A	yyddd or yyyyddd yy/ddd or yyyy/ddd yy.ddd or yyyy.ddd

Note: If RETAIN=Y, you cannot specify a date in the STARTT parameter of the initialization file.

day specifies the day of the week to select. Valid names include MONday, TUESday, WEDnesday, THURsday, FRIday, SATurday, and SUNday. You can also specify YESTER to retrieve event records created since yesterday, or TODAY to retrieve event records created today.

hh:mm:ssXM indicates the start time of day in hours (hh), minutes (mm), and seconds (ss) selected. XM indicates AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, valid times are 00:00-12:00 and you must indicate AM or PM. For example, 01:00 hours on the 24 hour clock is expressed as 1:00AM on the 12 hour clock. If you do not specify AM or PM, IBM Connect:Direct assumes the 24-hour clock. You do not need to specify minutes and seconds. You can also specify

NOON to retrieve event records starting at noon, or MIDNIGHT to retrieve event records starting at midnight. The default for the time is 00:00:00, the beginning of the day.

If you do not specify the STARTT parameter, event processing begins with the current time.

STOPT = ([date | day] [,hh:mm:ssXM])

specifies when interest in event data ends.

date specifies the stop date when event processing ends. You can specify the date in either Gregorian or Julian format.

Refer to the STARTT subparameter for a discussion of Gregorian and Julian date formats.

If you specify only the date, the time defaults to 24:00:00.

day specifies the day of the week to select. Valid values include MONday, TUESday, WEDnesday, THURsday, FRIday, SATurday, and SUNday. You can also specify TODAY.

hh:mm:ssXM indicates the stop time of day in hours (hh), minutes (mm), and seconds (ss) to select. XM indicates AM or PM. You can use the 24-hour clock or the 12-hour clock. If you use the 24-hour clock, valid times are 00:00-24:00. If you use the 12-hour clock, valid times are 00:00-12:00 and you must indicate AM or PM. For example, 01:00 hours on the 24 hour clock is expressed as 1:00AM on the 12 hour clock. If you do not specify AM or PM, IBM Connect:Direct assumes the 24-hour clock.

You do not need to specify minutes and seconds. You can also specify NOON to retrieve event records starting at noon, or MIDNIGHT to retrieve event records starting at midnight. The default time is 24:00:00, the end of the day.

If you do not specify the STOPT parameter, you must issue an EVENT SERVICES STOP command to stop event processing.

USER = name | (list)

limits the selected event records to those written for userids with the specified name. You can specify a list of names by enclosing them in parentheses. You can use wild card characters. For example, if you specify USER = SYS\$, then records with SYS\$ in the first four characters of the userid field are selected. Records having SYS\$BOB, SYS\$ADM, and SYS\$0001 all satisfy this selection criterion. Userid names can be up to 64 characters in length and can contain lowercase characters.

name specifies the userid to select.

(list) specifies a list of userids to select. Enclose the list in parentheses. Separate userids in the list with commas.

SNODE = name | (list)

limits the selected event records to those written for Processes where the specified node name acted as SNODE. You can specify a list of names by enclosing them in parentheses. You can use wild card characters. For example, if you specify SNODE=DALLAS*, then all records with DALLAS in the first six characters of the SNODE field are selected. Records having DALLAS.PROD, DALLAS.TEST, and DALLAS all satisfy this selection criterion. SNODE names can contain lowercase characters.

name specifies the SNODE to select.

(list) specifies a list of SNODES to select. Enclose the list in parentheses. Separate SNODES in the list with commas.

TYPE = id | (list)

specifies the event record types to select. Every event record that IBM Connect:Direct generates has an associated record-type identifier. Each identifier is two characters long and indicates the event or function that generated the record. The identifier also indicates the record format and contents.

id specifies the event record type.

(list) specifies a list of event record types to select. Enclose the list in parentheses. Separate record types in the list with commas.

FNAME=dsname | (list)

limits the selected event records to those that contain the specified filename. The FNAME subparameter is valid for the following record types: Copy Termination (CT), Run Job (RJ), Start IBM Connect:Direct (SD), and Submit within Process (SW).

name specifies the filename to select.

(list) specifies a list of filenames to select. Enclose the list in parentheses. Separate filenames in the list with commas.

The meaning of the filename within these records is unique for each record type. For example, the Run Job record contains the filename of the submitted JCL. Filenames can be up to 254 characters in length and can contain lowercase characters. Filenames must conform to your platform's naming conventions.

CASE=YES | NO

specifies whether lowercase or mixed-case data is permitted for the USER, SNODE, and FNAME subparameters. The CASE subparameter overrides the global CASE option defined at signon for the SELECT STATISTICS command.

YES changes the data in USER, SNODE, and FNAME to uppercase regardless of the actual data specified.

NO preserves the actual case entered for the USER, SNODE, and FNAME subparameters.

The CASE defaults to the setting defined in the session defaults, if nothing is specified.

LASTSEQ=n

specifies the last sequence number (for Restart capability) associated with the date/time stamp. Use it with the STARTT parameter. This information is provided by CICS when you issue a CREATE EVENT SERVICES REQUEST command for an event services request that finished abnormally.

MAXQCNT=n

specifies the maximum number of elements that can reside in the internal event queue for this request. All events that match the WHERE criteria for this request are placed on an internal queue. If events occur faster than the API can process them, the queue can become full. If the queue becomes full, IBM Connect:Direct terminates event services requests. You can modify the queue size using MAXQCNT. Each queue element takes approximately 2K in storage. The default is 100.

EXIT=exitname

specifies the name of an EVENT SERVICES OPTION exit. The sample exit DGAXEV01 is provided in the sample library. Event data is processed by an exit rather than the calling API, if EXIT is specified.

Optional Parameters

The following parameters are optional for the EVENT SERVICES CREATE command.

ORWHERE (CCODE = (condition, completion code) | PNAME = name | (list)

PNUM = number | (list)

USER = name | (list)

SNODE = name | (list)

TYPE = id | (list)

FNAME = dsname | (list)

specifies which statistics records you want to examine.

The subparameters, such as CCODE, PNAME, PNUMBER, and TYPE, are optional, but you must specify at least one.

The parameters for the ORWHERE keyword are defined in the WHERE keyword section.

Sample Command

The following is an example of the EVENT SERVICES CREATE command.

```
EVENT CREATE ID=PROCESS_TERM WHERE (TYPE=PT)
```

EVENT SERVICES START Command Format

The EVENT SERVICES START command enables you to start a specific notification request for event data. This command provides parameters where you can define what data (date, time, and sequence) is sent to your application.

The EVENT SERVICES START command uses the following format and parameters.

Command	Parameters
EVENT [SERVICES] START	ID=event-request-name
	WHERE=(STARTT LASTSEQ KEEPRREC (for CICS only)TRACE) (CICS only)

Required Parameter

The EVENT SERVICES START command has one required parameter.

ID=event-request-name

specifies the logical name of the event services request to start. Specify this name in a CREATE EVENT SERVICES command.

Optional Parameters

The following parameter is optional for the EVENT SERVICES START command.

WHERE(STARTT | LASTSEQ | KEEPRREC | TRACE)

sets restriction on event records meeting the following selection criteria.

STARTT=([date|day][,hh:mm:ssXM])

specifies selection by designated starting date and time. STARTT must be a date and time prior to the current time. If you specify this parameter, the event records retrieved from the statistics file begins with the date and time you specified, and continues until the current time is reached. At that point, event records are processed as they occur.

For a complete explanation of the subparameters, refer to [EVENT SERVICES CREATE Command Format](#).

LASTSEQ=n

specifies the last sequence number (for Restart capability) associated with the date/time stamp. Use this parameter with the STARTT parameter. This information is provided by CICS when you issue a CREATE EVENT SERVICES REQUEST command for an event services request that finished abnormally.

KEEPRREC (CICS-only)

specifies that the restart record created by IBM Connect:Direct is kept, regardless of whether the event request completed abnormally or not. If KEEPRREC is not specified, the restart record created by IBM Connect:Direct is deleted if the event request completes normally.

TRACE (CICS-only)

specifies that every event record placed on the CICS TDQ is also written to the NDMTRACE temporary storage (TS) queue. You can use the NDMTRACE temporary storage queue to verify that EVENT records are flowing to the transient data queue.

Sample Command

Following is an example of the EVENT SERVICES START command.

```
EVENT START ID=PROCESS_TERM
```

EVENT SERVICES STOP Command Format

The EVENT SERVICES STOP command stops event notification for a specific event services request.

The EVENT SERVICES STOP command stops event notification and deletes the event request from the system.

The EVENT SERVICES STOP command uses the following format and parameters.

Command	Parameters
EVENT [SERVICES] STOP	ID=event-request-name
	IMMEDIATE

Required Parameter

The EVENT SERVICES STOP command has one required parameter.

ID=event-request-name

specifies the logical name of the event services request to start. This name must match a name specified in a CREATE EVENT SERVICES command.

Optional Parameter

The EVENT SERVICES STOP command has one optional parameter.

IMMEDIATE

indicates that no more event records can be sent. If you do not specify IMMEDIATE, all records on the internal EVENT queue are sent before the EVENT request is terminated.

EVENT SERVICES DISPLAY Command Format

The EVENT SERVICES DISPLAY command enables you to display event data for a specific event services request. The EVENT SERVICES DISPLAY command uses the following format and parameter.

Command	Parameter
EVENT [SERVICES] DISPLAY	ID=event-request-name

Required Parameters

The EVENT SERVICES DISPLAY command has no required parameters.

Optional Parameter

The EVENT SERVICES DISPLAY command has one optional parameter.

ID=event-request-name

specifies the logical name of the event services request to display. This name must match a name specified in a CREATE EVENT SERVICES command. If you do not specify this parameter, all event services requests are displayed.

CICS ESS Customization

The IBM Connect:Direct CICS interface enables you to use IBM Connect:Direct through the Customer Information Control System (CICS) from local and remote sites. The system includes a set of nested menus, prompts for required information, online Help facilities, and monitoring features for current status.

In addition to the IBM Connect:Direct IUI, a facility is provided that enables you to issue standard IBM Connect:Direct commands from a CICS application program. You can use this interface for both terminal and non-terminal tasks. Following are two typical uses for this API:

- For terminal tasks, the API enables an installer to provide their own user interface to all or part of IBM Connect:Direct or work with IBM Connect:Direct from application programs using the DGAN transaction.
- For non-terminal tasks, the API enables an installer to write background transactions which programmatically issue IBM Connect:Direct commands. The most typical sequence is SIGNON - SUBMIT - SIGNOFF. In this case, the SIGNON generates a SIGNON TYPE=CICS, and the SIGNOFF results in a cleanup operation of the signon table entry, temporary storage, and so forth.

To avoid having to sign on before each command and sign off after each command, the user application can pass a logical task number through the Q012TASK field. If Q012TASK is used, a user program can sign on once and issue multiple commands in pseudo-conversational mode. A signoff is only needed when the user application is terminating.

An example of the use of this facility is provided in member DGAQ247 of the \$CD.SDGASAMP. This program enables you to type IBM Connect:Direct commands on a screen and view the resulting return code, message number, message text, and Process number assigned to your IBM Connect:Direct Process. In addition, the CICS application displays the name of the CICS Temporary Storage (TS) queue where the results of your command are stored, the count of items in the queue, and the maximum record length in the queue.

You must install both IBM Connect:Direct and the IBM Connect:Direct CICS interface, and they must be operating for the IBM Connect:Direct CICS API to function.

The following elements are required to use DGAQ247:

Component	Description
DGAQ247	The program source, written in assembly language
DGAQM98	The BMS map used by DGAQ247
DGAQBMST	A parsing macro used by DGAQ247
DGAQCA12	A command-level COMMAREA passed to the CICS Interface

Following is a sample IBM Connect:Direct API DRIVER screen. It shows data after the ESS registration program has signed on to the IBM Connect:Direct CICS API driver and issued an EVENT SERVICES CREATE command.

```

CONNECT:Direct API DRIVER
COMMAND      EVENT SERVICES CREATE ID=MES00031 WHERE(MAXQCNT=998 LASTSEQ=1
              STARTT=(02.14.2010,04:14:12.45) STOPT=(12.31.2041,24:00:00.00)
              TYPE=(QE,QW,QH,QT,PI,CE,CT,PT,CI,RJ,RT,JI,TI,SW,DP) )
COMMAND RC    0000
COMMAND MSG ID SOPM000I
COMMAND MSG    An EVENT REQUEST HAS COMPLETED NORMALLY
PROCESS NUMBER
TD EXIT COUNT
TS QUEUE NAME
TS MAX LRECL
EVENT RESTART DATA
WHERE(STARTT=(000000,00:00:00.00),LASTSEQ=000)
PF keys:  3 Exit   5 Signon   6 Command   7 Signoff

```

The following table describes the Driver fields:

Field	Description
COMMAND	This 3-line field contains your API command.
COMMAND RC	This 4-character field contains the return code from your API Process.
COMMAND MSG ID	This 8-character field contains the identification number of the message associated with your API Process.
COMMAND MSG	This 64-character field contains the text of the message.
PROCESS NUMBER	This 6-character field contains the Process number assigned by the system to your API Process.
TD EXIT COUNT	This 6-character field contains the number of bytes indicating how much data is written by the exit module for your API Process.
TS QUEUE NAME	This 8-character field contains the name of the TS queue used during your API request.
TS MAX LRECL	This 4-character field contains the maximum logical record length in bytes of the TS queue.
EVENT RESTART DATA	This 62-character field contains the DATE/TIME/SEQ of the last event record successfully received. Data is only displayed if an EVENT SERVICES CREATE command is issued and a previous event services request had ended abnormally.

CICS API Option

To use the sample program, you must first change DGAQ247 to issue a valid SIGNON command for your environment (the SIGNON command is defined near the end of the source module). Include a valid USERID, PASSWORD, and NODE. Before executing this program as a transaction, you must be signed on to CICS. Also observe the following items.

- Assemble DGAQM98 and then DGAQ247. Sample JCL is in the \$CD.SDGASAMP. Assemble DGAQM98 and DGAQ247 as follows:
 - Use member DGAXASMB to assemble DGAQM98
 - Use member DGAXASMC to assemble DGAQ247
- Use the IBM Connect:Direct Administrative (DGAA) transaction to verify that the CICS Interface and the appropriate node are both active.
- Use transaction DGAN to display DGAQM98.

The program checks for the presence of a communications area in the Exec Interface Block (EIB). If none is present, or if you press the Clear key, the DGAQM98 map is sent and a RETURN TRANSID is performed to invoke the transaction again when you press Enter.

- If you pressed the PF3 or PF15 key, the program terminates.

- The DGAQM98 map is received. If you pressed PF5 or PF17, a IBM Connect:Direct SIGNON request is generated and the results of the command are presented in the map.
- Following a successful signon (that is, the return code on DGAQM98 after the SIGNON is zero), you can type a valid IBM Connect:Direct command on the line provided in the DGAQM98 map and press PF6 or PF18 to send the command to the IBM Connect:Direct DTF. Command results are displayed when they are returned from the DTF.
- To signoff from the DTF, press PF7 or PF19.

When you type a command through the DGAQM98 screen, its length is determined and the address of the length and command are placed in the DGAQC12 communications area at label Q012CMDA. Program DGAQ012 is then invoked through an EXEC CICS LINK command. When control returns to DGAQ247, the DGAQC12 communications area contain the results of the command.

Note: The DGAQ247 does not display the results of the issued command stored in CICS temporary storage. Retrieve these records programmatically or view them using the CICS CEBR transaction.

Use the techniques in the DGAQ247 sample program to issue any valid IBM Connect:Direct command. Results of commands such as SELECT PROCESS and SELECT STATISTICS are written to CICS temporary storage; other commands may produce no output.

Accessing the API Module (DGAQ012)

Procedure

- Access the API by linking to program DGAQ012 within your CICS program/transaction as follows.

```
EXEC CICS LINK PROGRAM('DGAQ012') COMMAREA(Q012COMM) +
      LENGTH(Q012CMLH)
```

The COMMAREA Q012COMM is defined by macro DGAQCA12 in \$CD.SDGASAMP and is also provided in \$CD.SDGASAMP. A COBOL version of this record layout is provided as member DGAQAPIC in the sample library. A brief description of each of the fields follows:

Field	Assembler Directive	Format	Description
Q012CMDA	DS	XL4	Full word containing the address of the command to be issued to IBM Connect:Direct. The command must be in the following format: CMDLEN DS H Length of CMDTEXT that IBM Connect:Direct recognizes, not including length of CMDLEN. CMDTEXT DS Clx Command text
Q012RETC	DS	XL4	Return code received after the command is invoked. 0: request completed successfully 1: C:D-CICS interface level error 2: C:D-CICS node level error 3: C:D-CICS signon level error 5: invalid COMMAREA or command passed to CICS API 8 and above: unsuccessful request

Field	Assembler Directive	Format	Description
Q012TDCT	DS	XL4	TDEXIT ITEM COUNT. The field contains the number of records received from the IBM Connect:Direct DTF.

Q012CMDA is the only required parameter you must set before the EXEC CICS LINK to DGAQ012. For optimum performance, follow the example of DGAQ247 and initialize all fields before linking to DGAQ012.

Any output generated by issuing the command is returned in a temporary storage queue. Your terminal ID is displayed as the last four characters in the unique TS queue name.

LASTSEQ Parameter

All event records contain a unique timestamp that consists of date, time, and sequence number. Each EVENT SERVICES request is associated with an ID. The IBM Connect:Direct CICS API creates a restart record for an ID when an EVENT SERVICES CREATE command is processed.

After an event record is successfully delivered to the Transient Data Queue, its timestamp is saved in the restart file. If event services processing ends abnormally, the restart record is retained. The restart record is also retained when you specify KEEPREC (keep restart record) in the EVENT START statement.

When an EVENT SERVICES CREATE command is processed, the restart file is searched for a record matching the ID of the CREATE. If a match is found, the date, time, and sequence number of the last successfully delivered event record is returned. When you use the time specification in conjunction with the LASTSEQ parameter, you must include hours, minutes, seconds, and hundredths of seconds in the format hh:mm:ss.th.

Field	Assembler Directive	Format	Description
Q012TDMX	DS	XL2	TDEXIT MAX ITEM SIZE - the maximum record size received from the IBM Connect:Direct DTF.
Q012PROC	DS	CL6	Process number of the latest submitted Process.
Q012TSKY	DS	CL8	Temporary storage ID - the name of the CICS TS queue that contains the command output.
Q012MSID	DS	CL8	Message ID returned from the Connect:Direct for z/OS DTF.
Q012MSTX	DS	CL64	Message Text returned from the Connect:Direct for z/OS DTF.
Q012DTE	DS	PL4	Date of the last event acknowledgment.
Q012TME	DS	XL4	Time of the last event acknowledgment.
Q012SEQ	DS	XL2	Sequence number of the last event acknowledgment.

Note: DGAQ012 requires that it run in CICS key. For the transaction that executes a program linking DGAQ012, set the following parameters in the CEDA definitions (or in the RDO for that program):

- TaskDataLoc:ANY
- TaskDataKey:CICS

If missed event data is required, use this information as the STARTT and LASTSEQ values as in the following example.

Event records beginning after this timestamp are processed.

Event Record from the Transient Data Queue

The sample program DGAQ249 reads an ESS event record from the Transient Data queue. The sample program DGAQ249 is in \$CD.SDGASAMP. Use JCL DGAXASMC in \$CD.SDGASAMP to assemble DGAQ249.

Each record passed to the user application consists of a record header and an event record. DGA\$NHDR, which is in the IBM Connect:Direct sample library, maps the record header. Members in the sample library also map the event records. Search for Event Services Record Descriptions to determine which member maps a particular event record.

Event Services Record Descriptions

ESS enables you to create ESS-enabled applications. Following is reference information about event record types and attributes that help you create applications. For a list of statistics record types, search for Using IBM Connect:Direct Exits.

Assembler DSECT members are in the sample library. Access the information by the member names listed in the following table. Use the two-character designations for record types when you browse the IBM Connect:Direct statistics file.

Event Record Type Attributes

ESS has several statistics record types. This section summarizes the attributes of these record types.

Most record types have the following common attributes:

- Process name
- Process number
- Message ID
- Submitter's symbolic node name
- PNODE name for this Process
- SNODE name for this Process
- THISNODE (This node is P[node] or S[node].)

The following paragraphs detail the attributes of specific record types. For ease of reference, the information is grouped according to event type or enhancement type, and they are presented alphabetically.

Event Record Types

Record type attributes for event records are described in the following paragraphs. The information is organized alphabetically by record type identifier.

CI (COPY Step Initiation)

This record is created immediately prior to the execution of any tasks associated with a COPY statement. The intent of this record is to capture the timing of such work as data set allocation prior to any I/O occurring. In addition to the common attributes, CI has the following individual attributes:

- COPY step start date/time
- Step name or label
- CI record retry count
- Transfer direction
- Source file name
- Destination file name
- Member name (when applicable)

CE (COPY I/O Start)

This record is created immediately prior to the first block of data transferred in a COPY statement. In addition to the common attributes, CE has the following individual attributes:

- COPY I/O start date/time
- Step name or label
- CE record retry count

EI (Event Request Initiation)

This record is created upon completion of a EVENT SERVICES START command.

Note: The common attributes for Process-related records do not apply to this record.

The following attributes apply to this event:

- Date/Time the command completed
- User ID of the user who issued the command
- Event services request ID
- Completion code for the command
- Command parameter string

EL (Event Session Lost)

This record is created when the IBM Connect:Direct DTF is lost.

ET (Event Request Stop)

This record is created when an EVENT SERVICES STOP command completes.

Note: The common attributes for Process-related records do not apply to this record.

The following attributes apply to this event:

- Date/Time the command completed
- User ID of user who issued the command
- Event services request ID
- Completion code for the command

JI (RUN JOB Start)

This record is created immediately before the job specified by the RUN TASK statement is submitted. In addition to common attributes, JI has the following individual attributes:

- RUN JOB start date/time
- Step name or label
- Job name
- JI record retry count

QE (Process moved to EXEC queue)

This record is created when a Process is moved to the execution queue. In addition to the common attributes, QE has the following individual attributes:

- Process execution start date/time
- QE record retry count

PI (Process Initiation)

This record is created immediately prior to when the first Process step is executed. In addition to the common attributes, PI has the following individual attributes:

- Queue change date/time
- Execution queue status value
- PI record retry count

QH (Process moved to the HOLD queue)

This record is created when a Process is moved to the Hold queue, through a problem during Process execution or submission, a SUSPEND PROCESS command issued or a session failure. The reason is identified by the Hold queue status value. In addition to the common attributes, QH has the following individual attributes:

- Queue change date/time
- Hold queue status value
- QH record retry count

QT (Process moved to the TIMER queue)

This record is created whenever a Process is moved to the Timer queue, normally through a Process retry. In addition to the common attributes, QT has the following individual attributes:

- Queue change date/time
- Timer queue status value
- QT record retry count

QW (Process moved to the WAIT queue)

This record is created whenever a Process is moved to the Wait queue from the Hold or Timer queues. In addition to the common attributes, QW has the following individual attributes:

- Queue change date/time
- Wait queue status value
- QW record retry count

TI (RUN TASK Initiation)

This record is created immediately prior to the initiation of the task specified by the RUN TASK statement. In addition to the common attributes, TI has the following individual attributes:

- RUN TASK start date/time
- Step name or label
- Program name
- TI record retry count
- Date/time
- Process name and number
- Event name
- Event trigger
- Event type

Event Record Enhancements

An enhancement to existing statistics records is described in the following paragraph.

CT (COPY Termination) Record Enhancements

The existing CT record is enhanced to include the following information:

- Total number of retries for the COPY step
- Total amount of data moved for all attempts of the COPY
- Type keywords specified in a COPY step to retrieve data set defaults for allocating the destination file

Spool Transfer Facility

IBM Connect:Direct Spool Transfer is an interface that enables you to transfer Job Entry Subsystem (JES) spool files in the following ways:

- Copy from JES Spool Files

The VPS/CDI option and the VPSSCDI program manage input from JES spool files. Using this feature allows IBM Connect:Direct to transfer spool from one system to another.

- Copy to JES Print Queues

IBM Connect:Direct dynamically allocates a print file and writes the input file directly to the JES Spool.

- Copy to JES Reader Queues

IBM Connect:Direct dynamically allocates an internal reader and writes the input file directly to the JES Reader.

For outbound transfers, this feature requires the following additional products from Levi, Ray & Schoup (LRS):

- VTAM Printer Support (VPS)
- VPS IBM Connect:Direct Interface (VPS/CDI) Option

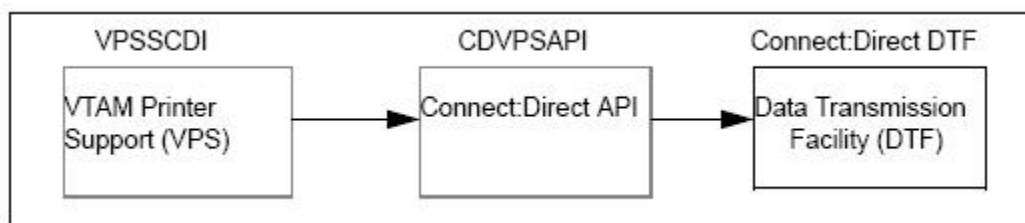
Note: To send output to JES2 or JES3 (Reader or a printer queue), you do not need the VTAM Printer Support and VPS IBM Connect:Direct Interface (VPS/CDI) Option components.

IBM Connect:Direct Spool Transfer uses standard IBM Connect:Direct facilities, which provide automation, reliability, management, interoperability, and security.

Spool Transfer Components

The major components of IBM Connect:Direct Spool Transfer are:

- VTAM Printer Support (VPS) System
- Application Program Interface (API)
- Data Transmission Facility (DTF)
- The following diagram shows the relationship between these components. This chapter summarizes the function of each component.

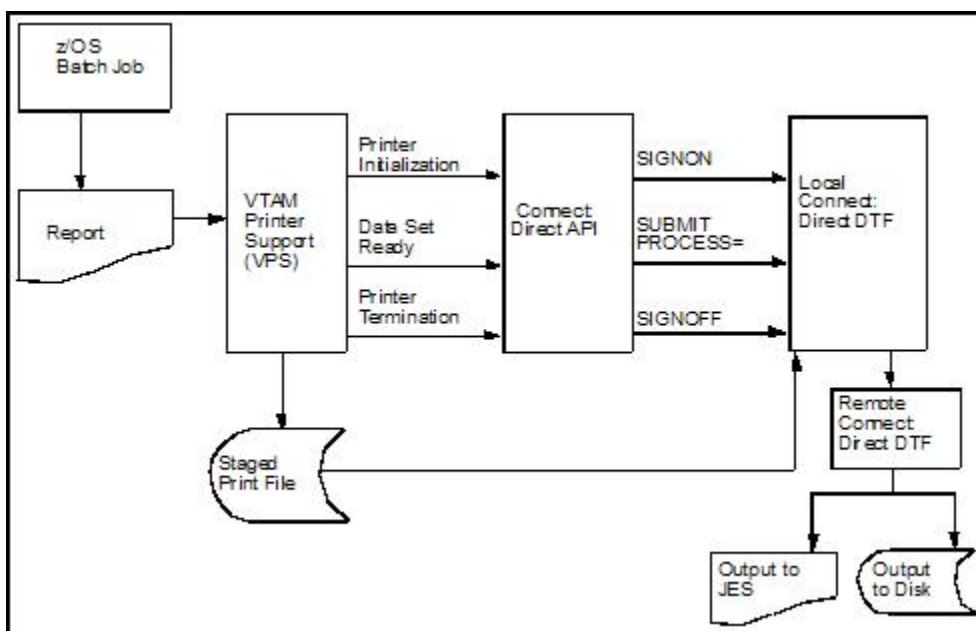


You implement IBM Connect:Direct Spool Transfer by changing definitions in the VPS initialization. Please refer to the Levi, Ray & Shoup documentation for their VPS product for correct implementation of VPS.

VTAM Printer Support

Connect:Direct for z/OS uses VTAM Printer Support (VPS) to retrieve JES spool files for processing. Expansion of the VPS application includes IBM Connect:Direct support for moving print files from the JES spool to any IBM Connect:Direct node. The VPS module (VPSSCDI) directs spooled output from JES to the IBM Connect:Direct API for file transfer.

The following diagram illustrates the flow of the print file through this process.



VPS passes control to the IBM Connect:Direct API at the following times:

Phase	Description
System Initialization	When a JES spool file is sent through IBM Connect:Direct, VPS first transfers control to the IBM Connect:Direct API for printer initialization and the API issues a SIGNON to the local DTF.
Printer Initialization	After signon, VPS copies the print files to a disk file and sends it to the IBM Connect:Direct API.
SYSOUT Data Set Ready	<p>The IBM Connect:Direct API constructs a SUBMIT PROCESS command to send printer attributes, data set name, and the submitter's user ID to the local DTF.</p> <p>If the SUBMIT fails because the Process can not be found or the SNODE is unavailable, the print file is returned to the JES print queue according to the values specified in the requeue parameters. If the SUBMIT fails for another reason, VPS deletes the staged data set, drains its DTF WRITER, and leaves the JES spool file in the JES queue for restart.</p> <p>If the SUBMIT is successful, VPS releases ownership of the staged data set to IBM Connect:Direct, and the JES spool file is PURGED from JES.</p>
Printer Termination	When no more JES files are printing to IBM Connect:Direct, a printer termination request is sent to IBM Connect:Direct API.
System Termination	The IBM Connect:Direct API then sends a SIGNOFF command to the local DTF.

You can change VPSSCDI to modify IBM Connect:Direct control blocks prior to calling DGASVPSA.

IBM Connect:Direct API

The IBM Connect:Direct API (DGASVPSA) communicates with the Data Transmission Facility through an application interface. DGASVPSA signs on to the local DTF, submits a Process, and signs off of the local DTF.

Module or Command	Description
DGASVPSA	<p>DGASVPSA is part of the VPS interface and is distributed in \$CD.SDGASAMP. This program is in source form. You can assemble and link-edit when you install a new release of IBM Connect:Direct VPS.</p> <p>It uses the VPS system log for reporting exception conditions and informational messages describing the Processes submitted to the local DTF. This log serves as one central point to determine the status of any print file controlled by VPS.</p> <p>The DGASVPSA program receives control from VPSSCDI and validates the function calls and control block fields passed by VPSSCDI. If a function executes successfully, the program returns a condition code of 0 (zero) to the VPSSCDI.</p> <p>If a noncritical error occurs, DGASVPSA returns a code of 4 to VPSSCDI to indicate that VPS requeues the print file according to VPS printer specifications.</p> <p>If DGASVPSA detects a critical error, it returns a code of 8 or greater, indicating that VPS can EDRAIN the printer.</p> <p>A function call of PRINTER INITIALIZATION causes a SIGNON to be sent to the local DTF as defined in the network map specification for this printer.</p> <p>A function call of DATA SET SYSOUT READY invokes the routines to validate the control blocks and build the IBM Connect:Direct SUBMIT PROCESS command and send it to the local DTF.</p> <p>A function call of PRINTER TERMINATION sends a SIGNOFF to the local DTF.</p> <p>DGASVPSA uses the IBM Connect:Direct API interface module (DGADCHLA) to communicate with the Data Transmission Facility.</p>
VPSSCDI	<p>The VPS module VPSSCDI, the interface to IBM Connect:Direct, is distributed in VPS. This program is in source form and you can reassemble and link-edit it when you install a new release of VPS or IBM Connect:Direct.</p>
SIGNON	<p>This command uses the network map specification for the specified IBM Connect:Direct printer.</p> <p>If you do not have a network map override specified in the VPS printer definition table (CDNETMAP), the SIGNON command uses the DMNETMAP DD statement in the VPS startup JCL for the network map.</p>
SIGNON Security	<p>The SIGNON command for DGASVPSA does not pass a user id or password. If you use the Stage 1 security exit, user id associated with the VPS started task is extracted and the user id and a special password (assigned by the Stage 1 exit) is placed in the UICB, a user interface control block used for security.</p> <p>If the Stage 1 processing is successful, the SIGNON command passes to the DTF, where the DTF Stage 2 security is invoked. The Stage 2 exit recognizes the special password, as assigned by the Stage 1 exit. All verification calls to the security system are by user id only.</p> <p>If you send to a secured IBM Connect:Direct, you can specify user id and password by adding the SNODEID specification to the Process. All Processes submitted by the VPS API have the security access level associated with the VPS job or started task, unless the security is overridden by PNODEID, SNODEID, and/or secure point-of-entry (SPOE) translation.</p>

Process Names

VPS print files are routed through the IBM Connect:Direct network by specifying the SNODE keyword on the submitted Process. To provide flexibility, you can set up the VPS IBM Connect:Direct printer either to

submit a single Process name regardless of printer class or to submit a Process name with the printer output class appended to the name.

By appending the printer class to the name, one VPS/CDI printer can submit up to 36 different Process names, one for each printer class A–Z or 0–9. You can control the Process name being submitted with the VPS printer name and VPS printer definitions as follows:

- Define the CDSNODE in the VPS Printer Definition Table
- Define the CDPMBR in the VPS Printer Definition Table
- Use a VPS printer name of seven characters or less
- Use a VPS printer name of eight characters

This flexibility enables a VPS printer to submit Processes that send print files to one or more IBM Connect:Direct SNODEs.

Use the following table to select the best method for your environment.

Where to Code	Keyword	Requirements	Results
Use Default (Do not code)	None	1 to 7-character printer name	Process name is the VPS Printer name with the class appended.
Use Default (Do not code)	None	8-character printer name	Process name is the VPS Printer name.
VPS Printer Definition Table	CDSNODE	Value for CDSNODE Process name specified in CDPMBR	Process name is the printer name. (Class is not appended.)
VPS Printer Definition Table	CDSNODE CDPMBR	Value for CDSNODE and CDPMBR	Process name is the name specified in CDPMBR. (Class is not appended.)
VPS Printer Definition Table	CDPMBR	8 characters	Process member name remains the same for all print files. (Class is not appended.)
VPS Printer Definition Table	CDPMBR	Up to 7 characters	Process name is the name specified in CDPMBR (Class is appended.)

Note: IBM Connect:Direct searches the IBM Connect:Direct Process library, SDGAPROC, for the Process name. You must define the Process names in the Process library.

For example, if you select CDPMBR=CDPROC and the output print class is A, IBM Connect:Direct searches the Process library for the Process name of CDPROCA. If you select an output print class of C for the same printer, IBM Connect:Direct searches for the Process name CDPROCC.

However, if you use an eight-character name such as CDPMBR=DGAXPRC1, the Process name for that printer is DGAXPRC1, regardless of the printer class.

Sample Process to Display Attributes in the SUBMIT Statement

The following sample Process is provided in the Install Process Library, SDGAPROC. This example shows all attributes that can pass from DGASVPSA to the local IBM Connect:Direct DTF in the SUBMIT statement. Any value of X'00' passed from VPS to DGASVPSA is considered a null value, and the SUBMIT command does not pass the corresponding field to the Process. For example, if your printer output DD statement does not code COPIES, then the symbolic &COPIES is not passed in the SUBMIT command.

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
/*
/*      C:D-ZOS JES SPOOL TRANSFER FACILITY SAMPLE PROCESS
/*
/*      CHANGE $CD.PROC TO THE Z/OS PROC LIBRARY THAT YOU HAVE
/*      INSTALLED THE PROC GENER INTO.  THE CONNECT:DIRECT
/*      INSTALLATION LOADED THIS MEMBER INTO YOUR
/*

```



```

/* CONNECT:DIRECT PROCESS LIBRARY. */
/* */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
CDPROCES PROC SNODE=&SNODE, -
&ADDR1=, /* ADDRESS LINE 1 */ -
&ADDR2=, /* ADDRESS LINE 1 */ -
&ADDR3=, /* ADDRESS LINE 1 */ -
&ADDR4=, /* ADDRESS LINE 1 */ -
&BLDG=, /* BUILDING */ -
&BURST=, /* BURST=YES OR NO */ -
&CHARS=, /* CHAR ARRANGEMENT TABLE */ -
&CKPTL=, /* CKPTLINE */ -
&CKPTP=, /* CKPTPAGE */ -
&CKPTS=, /* CKPTSEC */ -
&CLASS=, /* OUTPUT CLASS */ -
&CMOD=, /* COPY MODIFICATION MOD */ -
&CMTTC=, /* COPY MODULE TABLE REF */ -
&CNTL=, /* DEFAULT SPACING */ -
&COMPACT=, /* COMPACTATION TABLE */ -
&COPIES=, /* OUTPUT NUMBER COPIES */ -
&COPYG=, /* COPY GROUP(S) */ -
&DATCK=, /* DATAK */ -
&DEFAULT=, /* DEFAULT */ -
&DEPT=, /* DEPARTMENT */ -
&DEST=, /* OUTPUT DESTINATION */ -
&FCB=, /* OUTPUT WTR FCB */ -
&FLASH=, /* FLASH FORMS OVERLAY */ -
&FLSCT=, /* FLASH FORMS OVERLAY CNT */ -
&FMDEF=, /* FORMDEF */ -
&FORM=, /* OUTPUT FORM */ -
&GROUPID=, /* GROUPID */ -
&INDEX=, /* INDEX */ -
&JACCT=, /* JOB ACCOUNTING NUMBER */ -
&JESDS=, /* JESDS */ -
&JOBID=, /* JES ASSIGNED JOB ID */ -
&JOBNM=, /* JOB NAME */ -
&JPNAM=, /* JOB PROGRAMMERS NAME */ -
&JPROC=, /* JOB PROC NAME */ -
&JROOM=, /* JOB PROGRAMMERS ROOM */ -
&JSECL=, /* SECURITY LABEL */ -
&LINDEX=, /* LINDEX */ -
&LINECT=, /* LINES PER PAGE */ -
&NAME=, /* NAME */ -
&NOTIFY1=, /* 1ST NOTIFY ID */ -
&NOTIFY2=, /* 2ND NOTIFY ID */ -
&NOTIFY3=, /* 3RD NOTIFY ID */ -
&NOTIFY4=, /* 4TH NOTIFY ID */ -
&OPTCD=, /* OPTCD=J SPECIFIED */ -
&PGDEF=, /* PAGEDEF */ -
&PIMCT=, /* PIMSG MSG-COUNT */ -
&PIMSG=, /* PIMSG */ -
&PRMODE=, /* PRMODE */ -
&PRTY=, /* PRTY */ -
&ROOM=, /* ROOM */ -
&STEPDD=, /* STEP DDNAME */ -
&STEPNM=, /* STEP NAME */ -
&SUBNAME=, /* SUBMITTERS NAME */ -
&THRES=, /* THRESHLD */ -
&TITLE=, /* TITLE */ -
&TRC=, /* TRC */ -
&UCS=, /* OUTPUT WTR UCS */ -
&UDATA01=, /* 1ST USERDATA */ -
&UDATA02=, /* 2ND USERDATA */ -
&UDATA03=, /* 3RD USERDATA */ -
&UDATA04=, /* 4TH USERDATA */ -
&UDATA05=, /* 5TH USERDATA */ -
&UDATA06=, /* 6TH USERDATA */ -
&UDATA07=, /* 7TH USERDATA */ -
&UDATA08=, /* 8TH USERDATA */ -
&UDATA09=, /* 9TH USERDATA */ -
&UDATA10=, /* 10TH USERDATA */ -
&UDATA11=, /* 11TH USERDATA */ -
&UDATA12=, /* 12TH USERDATA */ -
&UDATA13=, /* 13TH USERDATA */ -
&UDATA14=, /* 14TH USERDATA */ -
&UDATA15=, /* 15TH USERDATA */ -
&UDATA16=, /* 16TH USERDATA */ -
&ULIB1=, /* 1ST USERLIB LIBRARY */ -
&ULIB2=, /* 2ND USERLIB LIBRARY */ -
&ULIB3=, /* 3RD USERLIB LIBRARY */ -
&ULIB4=, /* 4TH USERLIB LIBRARY */ -
&ULIB5=, /* 5TH USERLIB LIBRARY */ -

```

```

&ULIB6=, /* 6TH USERLIB LIBRARY */ -
&ULIB7=, /* 7TH USERLIB LIBRARY */ -
&ULIB8=, /* 8TH USERLIB LIBRARY */ -
&VPSDSN=, /* VPS PRINTER STAGED DSN */ -
&WRITER=, /* OUTPUT WRITER NAME */ -
&WTR=, /* OUTPUT WTR NAME (OLD) */ -
&OUTDSN=&SUBNAME..&JOBNM..&JOBID..&STEPNM..&STEPDD
/* * * * * * * * * * * * * * * * * * * * * * * * * */
/* COPY FILE STAGED BY VPSSCDI */
/* * * * * * * * * * * * * * * * * * * * * * * * */
STEP01 COPY FROM(PNODE -
DSN=&VPSDSN -
DISP=SHR ) -
CKPT=10M -
COMPRESS EXT -
TO(SNODE -
DSN=&OUTDSN -
DISP=(NEW,CATLG,DELETE) )
/* * * * * * * * * * * * * * * * * * * * * * * * */
/* DELETE FILE STAGED BY VPSSCDI */
/* * * * * * * * * * * * * * * * * * * * * * * * */
STEP02 IF (STEP01 = 0) THEN
STEP03 RUN TASK (PGM=DMRTDYN -
PARM=(C"ALLOC DSN=&VPSDSN,DISP=(OLD,DELETE)" -
F'-1' -
C"UNALLOC DSN=&VPSDSN"))
EIF
/* * * * * * * * * * * * * * * * * * * * * * * * */
/* EXECUTE PROC GENER TO PRINT THE OUTPUT FILE */
/* * * * * * * * * * * * * * * * * * * * * * * * */
STEP04 RUN TASK (PGM=DMRTSUB -
PARM=( "DSN=$CD.PROC(Gener),DISP=SHR", -
"ADDR1 &ADDR1", -
"ADDR2 &ADDR2", -
"ADDR3 &ADDR3", -
"ADDR4 &ADDR4", -
"BLDG &BLDG", -
"BURST &BURST", -
"CHARS &CHARS", -
"CKPTL &CKPTL", -
"CKPTP &CKPTP", -
"CKPTS &CKPTS", -
"CLASS &CLASS", -
"CMOD &CMOD", -
"CMTTC &CMTTC", -
"CNTL &CNTL", -
"COMPACT &COMPACT", -
"COPIES &COPIES", -
"COPYG &COPYG", -
"DATCK &DATCK", -
"DEFAULT &DEFAULT", -
"DEPT &DEPT", -
"DEST &DEST", -
"FCB &FCB", -
"FLASH &FLASH", -
"FLSCT &FLSCT", -
"FMDEF &FMDEF", -
"FORM &FORM", -
"GROUPID &GROUPID", -
"INDEX &INDEX", -
"INPUT &OUTDSN", -
"JACCT &JACCT", -
"JESDS &JESDS", -
"JNOTIFY &JNOTIFY", -
"JOBID &JOBID", -
"JOBNM &JOBNM", -
"JPNAME &JPNAME", -
"JPROC &JPROC", -
"JROOM &JROOM", -
"JSECL &JSECL", -
"LINDEX &LINDEX", -
"LINECT &LINECT", -
"NAME &NAME", -
"NOTIFY1 &NOTIFY1", -
"NOTIFY2 &NOTIFY2", -
"NOTIFY3 &NOTIFY3", -
"NOTIFY4 &NOTIFY4", -
"OPTCD &OPTCD", -
"PGDEF &PGDEF", -
"PIMCT &PIMCT", -
"PIMSG &PIMSG", -
"PRMODE &PRMODE", -

```

```

"PRTY      &PRTY",
"ROOM      &ROOM",
"STEPDD    &STEPDD",
"STEPNM    &STEPNM",
"SUBNAME   &SUBNAME",
"THRES     &THRES",
"TITLE     &TITLE",
"TRC       &TRC",
"UCS       &UCS",
"UDATA01   &UDATA01",
"UDATA02   &UDATA02",
"UDATA03   &UDATA03",
"UDATA04   &UDATA04",
"UDATA05   &UDATA05",
"UDATA06   &UDATA06",
"UDATA07   &UDATA07",
"UDATA08   &UDATA08",
"UDATA09   &UDATA09",
"UDATA10   &UDATA10",
"UDATA11   &UDATA11",
"UDATA12   &UDATA12",
"UDATA13   &UDATA13",
"UDATA14   &UDATA14",
"UDATA15   &UDATA15",
"UDATA16   &UDATA16",
"ULIB1     &ULIB1",
"ULIB2     &ULIB2",
"ULIB3     &ULIB3",
"ULIB4     &ULIB4",
"ULIB5     &ULIB5",
"ULIB6     &ULIB6",
"ULIB7     &ULIB7",
"ULIB8     &ULIB8",
"VPSDSN    &VPSDSN",
"WRITER    &WRITER",
"WTR       &WTR",
)) SNODE
EXIT

```

DGAPGNER Job

The following sample JCL is provided in the Install Process Library, SDGASAMP. This example is the DGAPGNER job that is submitted by DGADTSUB in STEP 04 of the preceding Process example. Because it runs on the SNODE, it shows how you can print files back into the JES spool at the remote site.

If IBM Connect:Direct Spool Transfer is enabled at the remote site, use IBM Connect:Direct to transfer output back into the JES spool. Refer to [Output to the JES Reader](#).



CAUTION: When you use this JCL to test your own VPS, a print loop can occur if you use the symbolic &DEST on the same VPS system from which you submitted the print job.

```

//&JOBNM    JOB (00000),&PGMR,PRTY=12,TIME=(10),
//REGION=4096K,MSGLEVEL=(1,1),MSGCLASS=X
//*
//STEP01    EXEC PGM=IEBGENER
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DSN=&INPUT,
//          DISP=(OLD,DELETE,KEEP)
//SYSUT2    DD SYSOUT=&CLASS,COPIES=&COPIES,DEST=&DEST
//*
//*ADDR1=&ADDR1
//*ADDR2=&ADDR2
//*ADDR3=&ADDR3
//*ADDR4=&ADDR4
//*BLDG=&BLDG
//*BURST=&BURST
//*CHARS=&CHARS
//*CKPTL=&CKPTL
//*CKPTP=&CKPTP
//*CKPTS=&CKPTS
//*CLASS=&CLASS
//*CMOD=&CMOD
//*CMTTC=&CMTTC
//*CNTL=&CNTL
//*COMPACT=&COMPACT
//*COPIES=&COPIES

```

```

// *COPYG=&COPYG
// *DATCK=&DATCK
// *DEFAULT=&DEFAULT
// *DEPT=&DEPT
// *DEST=&DEST
// *FCB=&FCB
// *FLASH=&FLASH
// *FLSCT=&FLSCT
// *FMDEF=&FMDEF
// *FORM=&FORM
// *GROUPIX=&GROUPIX
// *HOLD=&HOLD
// *INDEX=&INDEX
// *INPUT=&INPUT
// *JACCT=&JACCT
// *JESDS=&JESDS
// *JNOTIFY=&JNOTIFY
// *JOBID=&JOBID
// *JOBNM=&JOBNM
// *JPNAME=&JPNAME
// *JPROC=&JPROC
// *JROOM=&JROOM
// *JSECL=&JSECL
// *LINDEX=&LINDEX
// *LINECT=&LINECT
// *NAME=&NAME
// *NOTIFY1=&NOTIFY1
// *NOTIFY2=&NOTIFY2
// *NOTIFY3=&NOTIFY3
// *NOTIFY4=&NOTIFY4
// *OPTCD=&OPTCD
// *PGDEF=&PGDEF
// *PIMCT=&PIMCT
// *PIMSG=&PIMSG
// *PRMODE=&PRMODE
// *PRTY=&PRTY
// *ROOM=&ROOM
// *STEPDD=&STEPDD
// *STEPNM=&STEPNM
// *SUBNAME=&SUBNAME
// *THRES=&THRES
// *TITLE=&TITLE
// *TRC=&TRC
// *UCS=&UCS
// *UDATA01=&UDATA01
// *UDATA02=&UDATA02
// *UDATA03=&UDATA03
// *UDATA04=&UDATA04
// *UDATA05=&UDATA05
// *UDATA06=&UDATA06
// *UDATA07=&UDATA07
// *UDATA08=&UDATA08
// *UDATA09=&UDATA09
// *UDATA10=&UDATA10
// *UDATA11=&UDATA11
// *UDATA12=&UDATA12
// *UDATA13=&UDATA13
// *UDATA14=&UDATA14
// *UDATA15=&UDATA15
// *UDATA16=&UDATA16
// *ULIB1=&ULIB1
// *ULIB2=&ULIB2
// *ULIB3=&ULIB3
// *ULIB4=&ULIB4
// *ULIB5=&ULIB5
// *ULIB6=&ULIB6
// *ULIB7=&ULIB7
// *ULIB8=&ULIB8
// *VPSDSN=&VPSDSN
// *WRITER=&WRITER
// *WTR=&WTR

```

Symbolic Definitions

This topic identifies the parameters that can be passed to the Process submitted by the IBM Connect:Direct API.

Job and Jobstep Values

The following table lists the job and jobstep values.

Parameter	Definition
&JACCT	Job accounting number
&JNOTIFY	NOTIFY value
&JOBID	JOB ID assigned by JES
&JOBNM	JOB name
&JPNAME	Name of the programmer
&JPROC	Procedure name
&JROOM	Room number of the programmer
&JSECL	Security label
&STEPDD	DD name of the print file DD statement
&STEPNM	Process step name
&SUBNAME	User ID of the submitter
&VPDSN	Printer staged data set

Printer File Attributes

Refer to [Sample Process for SYSOUT](#) for the available parameters and associated symbolic names for print file attributes passed by the VPSSCDI interface program. Any non-zero attributes for a printer file cause the corresponding symbolic name to be generated on the SUBMIT command created by the IBM Connect:Direct interface program, DGASVPSA . You can reference these symbolic names in the IBM Connect:Direct Process you are submitting.

Customizing VPSSCDI

VPSSCDI creates the control blocks that the IBM Connect:Direct API uses to build the SIGNON, SUBMIT, and SIGNOFF commands. Any customized validation or modification to the IBM Connect:Direct control blocks is done at this point.

You can change printer values to reflect the destination environment. For example, you can change the destination name to a printer name at your location.

You can perform security or usage validation by validating any fields and either continuing the processing or returning to VPS with the appropriate return code settings.

The VPS return code settings are:

- RC=00 Normal execution continues.
- RC=04 Print file is queued per VPS printer specifications.
- RC=08 The printer is EDRAINED.

Note: Do not change VPS control blocks.

Transferring Data to the JES Reader or Spool

IBM Connect:Direct Spool Transfer transfers data from any IBM Connect:Direct node and routes job control language (JCL) statements to the JES reader or sends files to the JES spool.

By transferring data to the JES reader, any IBM Connect:Direct node can send z/OS JCL to the JES reader. Unlike the IBM Connect:Direct RUN JOB, z/OS JCL can be located where the Process is executing, on any media that IBM Connect:Direct supports.

The RUN JOB Security Exit provides a standard interface for security verification of job streams before they are submitted. For information on the RUN JOB Security Exit, search for [Implementing Security](#).

IBM Connect:Direct Spool Transfer uses dynamic allocation (SVC 99) to allocate the JES spool file. You can also use the OUTADD and OUTDEL macros to allocate an OUTPUT DD reference statement dynamically.

You can print a IBM Connect:Direct banner page at the beginning of each print report. For information, search for [IBM Connect:Direct Banners](#).

IBM Connect:Direct Spool Transfer supports the checkpoint/restart facility. IBM Connect:Direct takes a checkpoint at the top of form of the page printed after the checkpoint interval. Top of form is one of the following:

- X'F1' for ASA carriage control
- X'8B' for machine carriage control
- LINECT (line count) if NOCC or the TOF=X'xx' is specified in the Process

Output to the JES Reader

Use IBM Connect:Direct syntax to transfer data to the JES reader. To initiate this transfer, specify the keyword, **READER**, in the COPY TO Process statement for IBM Connect:Direct nodes where this feature is enabled.

For older releases of IBM Connect:Direct that do not support the **READER** keyword, use the **SYSOPTS** parameter **OUTPUT=READER**.

Examples of JCL Output

You can adapt the following Process language examples to output JCL into the JES reader. These examples are in the Sample Library, **SDGAPROC**.

Using the **READER** Keyword

The following sample Process (DGAPRDR2) uses the **READER** keyword to output JCL into the JES reader.

CDTORDR2	PROCESS	SNODE=xx.xxx.xxxxxx	-
STEP01	COPY	FROM(PNODE	-
		DSN=SAMPLE.JCL.LIB(JCL)	-
		DISP=SHR	-
)	-
		TO(SNODE	-
		READER	-
)	-
EXIT			

Using the **SYSOPTS**

The following sample Process (DGAPRDR1) uses **SYSOPTS** to output JCL into the JES reader.

CDTORDR1	PROCESS	SNODE=xx.xxx.xxxxxx	-
STEP01	COPY	FROM(PNODE	-
		DSN=SAMPLE.JCL.LIB(JCL)	-
		DISP=SHR	-
)	-
		TO(SNODE	-
		DSN=NULLFILE	-
		DISP=RPL	-
		SYSOPTS="OUTPUT=READER"	-

EXIT)

Output to JES Spool Files

You can also use IBM Connect:Direct syntax to direct files transferred through IBM Connect:Direct directly to the JES2/JES3 spool.

Copying a file into the JES spool is controlled through the SYSOUT=(.....) keyword on the COPY TO Process statement. Keyword and parameter definitions are in [COPY TO Syntax to Support Spool Transfer](#).

You can specify all keywords available on the z/OS JCL SYSOUT DD statement within the IBM Connect:Direct SYSOUT=(...) Process keyword. The format of all subparameters within the IBM Connect:Direct SYSOUT=(...) follows the rules in the OS/390 JCL Reference Guide for that keyword.

Note: You cannot use SYSOUT=class because the IBM Connect:Direct feature requires that you specify all subparameters within parentheses.

IBM Connect:Direct Banners

You can print a IBM Connect:Direct banner page at the beginning of each print report. If you selected the checkpoint/restart option for the file transfer, the word RESTARTED is added to the banner page during restart.

Select this option by coding the BANNER=(...) subparameter of the SYSOUT=(...) Process keyword. Banner values are supplied in pairs of literal = value. One banner line is printed for each supplied pair. Keyword and parameter definitions are in [IBM Connect:Direct COPY Process](#).

If the value contains a character other than 0–9 or A–Z, you must enclose it in quotation marks. For example, PROGRAMMER=John Doe must be specified as PROGRAMMER="John Doe". JOBNAME=OS/400 must be specified as JOBNAME="OS/400".

You can use literals of 1–25 characters. Literals longer than 25 characters are truncated. Values can be 1–30 characters. Values longer than 30 characters are truncated. If the literal or value contains a character other than 0–9 or A–Z, you must enclose it in quotation marks.

In the following example, three banner data lines are printed. JUNK is ignored because it is not paired with a value.

```
BANNER=(PROGRAMMER=&JPNAME, JOBNAME=&JOBNM,          -
          SUBMITTER=&SUBNAME, STEPNAME=&STEPNM,        -
          DDNAME=&STEPDD, JUNK)
```

Assuming that the symbolics are set to the values defined previously, the following example is generated.

```
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
****  PROGRAMMER          JOHN DOE          ****
****  JOBNAME             JOHN1X            ****
****  SUBMITTER            JOHN1             ****
****  STEPNAME             STEP1             ****
****  DDNAME              SYSUT2            ****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
```

The following example illustrates the banner that prints if the job is restarted.

```

***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** R E S T A R T E D *****
**** PROGRAMMER          JOHN DOE          ****
**** JOBNAME              JOHN1X            ****
**** SUBMITTER            JOHN1             ****
**** STEPNAME             STEP1             ****
**** DDNAME               SYSUT2            ****
***** R E S T A R T E D *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****
***** C O N N E C T : D I R E C T *****

```

IBM Connect:Direct COPY Process

The IBM Connect:Direct COPY Process controls the printing of the output data set by specifying key values within the new Process keyword SYSOUT=(...) or by using the SYSOPTS=(SYSOUT=(...)) for platforms other than z/OS.

COPY Process Examples

The following sample Process shows the simplest COPY Process. SYSOUT=A is not a valid value in this example. Put all subparameters in parentheses, and set CLASS to the output class assigned to the print file.

```

TESTP  PROCESS  SNODE=
STEP01  COPY     FROM(DSN=NAME,DISP=SHR)      -
          TO(SYSOUT=(CLASS=A))                -
          )
END

```

The next example is an alternative way to write the same Process. In this example, the CLASS= parameter is not required because of the additional parentheses in the first parenthetical group of the SYSOUT keyword. This example corresponds to the syntax of the SYSOUT DD statement.

```

TESTP  PROCESS  SNODE=
STEP01  COPY     FROM(DSN=NAME,DISP=SHR)      -
          TO(SYSOUT=((A,writer-name,form-name),...)) -
          )
END

```

The following example uses the SYSOUT= keyword and its subparameters within the SYSOPTS keyword.

```

TESTP  PROCESS  SNODE=
STEP01  COPY     FROM(DSN=NAME,DISP=SHR)      -
          TO(DSN=NULLFILE                     -
             SYSOPTS="(SYSOUT=(CLASS=A))"    -
             )
END

```

Carriage Control

The input file DCB RECFM specification determines carriage control. If RECFM is VA, VBA, FA, FBA, or UA, ASA carriage control is assigned (ASA is also the default). If RECFM is VM, VBM, FM, FBM, or UM, then machine carriage control is assigned.

You can optionally override the specification by defining the NOCC, LINECT, and TOF parameters in the SYSOUT keyword CC=. Keyword and parameter definitions are in the following section, [COPY TO Syntax to Support Spool Transfer](#).

The following example shows how to override carriage control specifications.

```

TESTP  PROCESS  SNODE=
STEP01 COPY      FROM(DSN=NAME,DISP=SHR)          -
                  TO(SYSOUT=(CLASS=A,             -
                     CC=(NOCC,LINECT=n,TOF=x'F1'))  -
                     )
                  END

```

COPY TO Syntax to Support Spool Transfer

All keywords and parameters of the COPY TO statement that support Spool Transfer are defined in this section. The table provided in the Symbolics Definitions section identifies where the parameters of the SYSOUT=(...) keyword are documented.

For detailed information on other keywords and parameters, see the [Connect:Direct Process Language](#) help.

Keyword	Description
READER	Specifies the output data set that is directed to the JES reader.
SYSOUT=(...)	Controls the printing of the output data set.
BANNER= ('literal 1' = value1 [, 'literal 2' = value2[,...]])	<p>Specifies the banner values in pairs of literal = value. One banner line is printed for each supplied pair. A literal without a corresponding value is ignored.</p> <p>You can use literals of 1–25 characters. Literals greater than 25 characters are truncated. You can use a value size of 1–30 characters. Values greater than the 30 characters are truncated.</p> <p>If the name or value contains a character other than 0–9 or A–Z, you must enclose it in quotation marks.</p>
CC=(<u>A</u> M NOCC LINECT= TOF=)	<p>Overrides carriage control supplied by the input file DCB RECFM.</p> <p><u>A</u> (for ASA) is assigned if RECFM is VA, VBA, FA, FBA, or UA. It is also the default if carriage control is not specified.</p> <p>M (for machine) is assigned if RECFM is VM, VBM, FM, FBM, or UM.</p> <p>NOCC identifies the file as having no carriage control. If you specify NOCC, you must also specify LINECT and TOF.</p> <p>LINECT=nnn identifies the number of lines printed per page. The default is <u>55</u>.</p> <p>TOF=<u>X'F1'</u> identifies the hex character that is inserted in the print line at the interval specified in the LINECT parameter. The hex character specified by the TOF parameter indicates to skip to the top of form. The default is <u>X'F1'</u>.</p>
SYSOPTS="..."	<p>OUTPUT=READER specifies that the output data set is directed to the JES reader.</p> <p>SYSOUT=(class=class) (class,writer-name,form-name) specifies that the output data set is directed to the JES spool files on platforms other than z/OS.</p>

Symbolic Definitions for the SYSOUT Keyword

The following table lists the subparameter fields for the SYSOUT keyword, identifies the format for the parameter, and the VPS symbolic name.

The As Documented by column states where to find additional information about the parameters. Spool Transfer indicates that the parameter is documented in this book. References to JCL parameters indicate that additional information is available in the MVS JCL documentation.

Parameter	Format	VPS Symbolic Name	As Documented By
ADDRESS=	('address line1', address line2', address line3, address line4)	&ADDR1, &ADDR2, &ADDR3, &ADDR4	SYSOUT DD
BANNER=	('literal 1' = value1, 'literal 2' = value2,...)		Spool Transfer
BUILDING=	'building identification'	&BLDG	OUTPUT DD
BURST=	Yes or No	&BURST	OUTPUT DD & SYSOUT DD
CC=	ASA or Machine or NOCC		Spool Transfer
CHARS=	(tablename, tablename...)	&CHARS	OUTPUT DD & SYSOUT DD
CKPTLINE=	nnnnn	&CKPTL	OUTPUT DD
CKPTPAGE=	nnnnn	&CKPTP	OUTPUT DD
CKPTSEC=	nnnnn	&CKPTS	OUTPUT DD
CLASS=	class	&CLASS	OUTPUT DD
COMPACT=	compaction-table-name	&COMPACT	
COMSETUP=	name		OUTPUT DD
CONTROL=	PROGRAM SINGLE DOUBLE TRIPLE	&CNTL	OUTPUT DD
COPIES=	nnn (,(group value, group value,...))	&COPIES ©G	SYSOUT DD OUTPUT DD
DATAck=	BLOCK UNBLOCK BLKCHAR BLKPDS	&DATCK	OUTPUT DD
DEFAULT=	Yes or No	&DEFAULT	OUTPUT DD
DEPT=	'department identification'	&DEPT	OUTPUT DD
DEST=	destination (nodename, userid)	&DEST	SYSOUT DD OUTPUT DD
DPAGELBL=	Yes or No		OUTPUT DD
FCB=	fcb-name (fcb-name, align verify)	&FCB	SYSOUT DD & OUTPUT DD

Parameter	Format	VPS Symbolic Name	As Documented By
FLASH=	(overlay-name, count) (,count)	&FLASH &FLSCT	SYSOUT DD & OUTPUT DD
FORMDEF=	membername	&FMDEF	OUTPUT DD
FORMS=	form-name	&FORM	OUTPUT DD
GROUPID=	output-group	&GROUPID	OUTPUT DD
HOLD=	Yes or No		SYSOUT DD
INDEX=	nn	&INDEX	OUTPUT DD
JESDS=	ALL JCL LOG MSG	&JESDS	OUTPUT DD
JOBNAME=	job name	&JOBNM	Spool Transfer
LINDEX	nn	&LINDEX	OUTPUT DD
LINECT=	nnn	&LINECT	OUTPUT DD
MODIFY=	module-name (module-name, trc)	&CMOD &CMTTC	SYSOUT DD & OUTPUT DD
NAME=	name of output separator	&NAME	OUTPUT DD
NOTIFY=	node.userid (node.userid, node.userid,...)	&NOTIFY1-4	OUTPUT DD
OPTJ=	Yes or No	&OPTCD	Spool Transfer
OUTBIN=	nnnnn		OUTPUT DD
OUTDISP=	(normal, abnormal)		OUTPUT DD
OUTPUT=	(reference)		SYSOUT DD
PAGEDEF=	membername	&PGDEF	OUTPUT DD
PIMSG=	(yes,msg-count) (no,msg-count)	&PIMSG &PIMCT	OUTPUT DD
PRMODE=	line page process-mode	&PRMODE	OUTPUT DD
PRTY=	nnn	&PRTY	OUTPUT DD
ROOM=	'room identification'	&ROOM	OUTPUT DD
SYSAREA=	Yes No		OUTPUT DD
SYSOUT=	(class) (class, writer- name, form-name)	&CLASS &WRITER &FORM	SYSOUT DD

Parameter	Format	VPS Symbolic Name	As Documented By
THRESHLD=	nnnnnnnn	&THRES	OUTPUT DD
TITLE=	'description of output'	&TITLE	OUTPUT DD
TOF=	X'F1'		Spool Transfer
TRC=	Yes No	&TRC	OUTPUT DD
UCS=	character-set-code (character-set-code, fold, verify)	&UCS	OUTPUT DD SYSOUT DD
USERDATA=	('user data description', 'user data description',...)	&UDATA01–16	OUTPUT DD
USERLIB=	('library name 1', 'library name 2',...)	&ULIB1–8	OUTPUT DD
WRITER=	name	&WRITER	OUTPUT DD

Sample Process for SYSOUT

The following sample Process shows all attributes that you can specify within the SYSOUT=(...) keyword. This sample Process is provided in the Sample Library, SDGAPROC.

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/*
/*          * * *   S A M P L E   P R O C E S S   * * *
/*
/*          C:D-ZOS JES SPOOL TRANSFER FACILITY SAMPLE PROCESS
/*          INPUT FROM JES SPOOL TRANSFER FACILITY
/*          OUTPUT TO VPS CONTROLLED PRINTER
/*
/*
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
CDTOJES  PROC   SNODE=&SNODE,
&ADDR1=,      /* ADDRESS LINE 1          */ -
&ADDR2=,      /* ADDRESS LINE 1          */ -
&ADDR3=,      /* ADDRESS LINE 1          */ -
&ADDR4=,      /* ADDRESS LINE 1          */ -
&BLDG=,       /* BUILDING                */ -
&BURST=,      /* BURST=YES OR NO        */ -
&CHARS=,      /* CHAR ARRANGEMENT TABLE */ -
&CKPTL=,      /* CKPTLINE                */ -
&CKPTP=,      /* CKPTPAGE                */ -
&CKPTS=,      /* CKPTSEC                 */ -
&CLASS=,      /* OUTPUT CLASS            */ -
&CMOD=,       /* COPY MODIFICATION MOD   */ -
&CMTTC=,      /* COPY MODULE TABLE REF  */ -
&CNTL=,       /* DEFAULT SPACING         */ -
&COMPACT=,    /* COMPACTATION TABLE     */ -
&COPIES=,     /* OUTPUT NUMBER COPIES    */ -
&COPYG=,     /* COPY GROUP(S)           */ -
&DATCK=,      /* DATAK                  */ -
&DEFAULT=,    /* DEFAULT                 */ -
&DEPT=,       /* DEPARTMENT              */ -
&DEST=,       /* OUTPUT DESTINATION       */ -
&FCB=,        /* OUTPUT WTR FCB          */ -
&FLASH=,      /* FLASH FORMS OVERLAY     */ -
&FLSCT=,      /* FLASH FORMS OVERLAY CNT*/ -
&FMDEF=,      /* FORMDEF                 */ -
&FORM=,       /* OUTPUT FORM             */ -
&GROUPID=,    /* GROUPID                 */ -
&INDEX=,      /* INDEX                   */ -
&JACCT=,      /* JOB ACCOUNTING NUMBER   */ -
&JESDS=,      /* JESDS                   */ -
&JOBID=,      /* JES ASSIGNED JOB ID     */ -

```

```

&JOBNM=, /* JOB NAME */ -
&JPNAME=, /* JOB PROGRAMMERS NAME */ -
&JPROC=, /* JOB PROC NAME */ -
&JROOM=, /* JOB PROGRAMMERS ROOM */ -
&JSECL=, /* SECURITY LABEL */ -
&LINDEX=, /* LINDEX */ -
&LINECT=, /* LINES PER PAGE */ -
&NAME=, /* NAME */ -
&NOTIFY1=, /* 1ST NOTIFY ID */ -
&NOTIFY2=, /* 2ND NOTIFY ID */ -
&NOTIFY3=, /* 3RD NOTIFY ID */ -
&NOTIFY4=, /* 4TH NOTIFY ID */ -
&OPTCD=, /* OPTCD=J SPECIFIED */ -
&PGDEF=, /* PAGEDEF */ -
&PIMCT=, /* PIMSG MSG-COUNT */ -
&PIMSG=, /* PIMSG */ -
&PRMODE=, /* PRMODE */ -
&PRTY=, /* PRTY */ -
&ROOM=, /* ROOM */ -
&STEPDD=, /* STEP DDNAME */ -
&STEPM=, /* STEP NAME */ -
&SUBNAME=, /* SUBMITTERS NAME */ -
&THRES=, /* THRESHLD */ -
&TITLE=, /* TITLE */ -
&TOF=X'F1', /* ASA TOP OF FORM */ -
&TRC=, /* TRC */ -
&UCS=, /* OUTPUT WTR UCS */ -
&UDATA01=, /* 1ST USERDATA */ -
&UDATA02=, /* 2ND USERDATA */ -
&UDATA03=, /* 3RD USERDATA */ -
&UDATA04=, /* 4TH USERDATA */ -
&UDATA05=, /* 5TH USERDATA */ -
&UDATA06=, /* 6TH USERDATA */ -
&UDATA07=, /* 7TH USERDATA */ -
&UDATA08=, /* 8TH USERDATA */ -
&UDATA09=, /* 9TH USERDATA */ -
&UDATA10=, /* 10TH USERDATA */ -
&UDATA11=, /* 11TH USERDATA */ -
&UDATA12=, /* 12TH USERDATA */ -
&UDATA13=, /* 13TH USERDATA */ -
&UDATA14=, /* 14TH USERDATA */ -
&UDATA15=, /* 15TH USERDATA */ -
&UDATA16=, /* 16TH USERDATA */ -
&ULIB1=, /* 1ST USERLIB LIBRARY */ -
&ULIB2=, /* 2ND USERLIB LIBRARY */ -
&ULIB3=, /* 3RD USERLIB LIBRARY */ -
&ULIB4=, /* 4TH USERLIB LIBRARY */ -
&ULIB5=, /* 5TH USERLIB LIBRARY */ -
&ULIB6=, /* 6TH USERLIB LIBRARY */ -
&ULIB7=, /* 7TH USERLIB LIBRARY */ -
&ULIB8=, /* 8TH USERLIB LIBRARY */ -
&VPSDSN=, /* VPS PRINTER STAGED DSN */ -
&WRITER=, /* OUTPUT WRITER NAME */ -
&WTR=, /* OUTPUT WTR NAME (OLD) */ -
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* COPY FILE FROM VPS STAGED DATASET INTO JES SPOOL */
/* DESTINATION HAS BEEN CHANGED TO A VPS CONTROLLED PRINTER. */
/* BANNER PAGE WILL BE PRODUCED. */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
STEP01 COPY FROM( PNODE
DSN=&VPSDSN
DISP=SHR )
CKPT=1M
COMPRESS EXT
TO( SNODE
SYSOUT=(CLASS=&CLASS,COPIES=&COPIES,DEST=&DEST
BANNER=(PROGRAMMER=&JPNAME,JOBNM=&JOBNM,
SUBMITTER=&SUBNAME,STEPNAME=&STEPM,
DDNAME=&STEPDD),JOBNM=&JOBNM)
)
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* DELETE VPS STAGE INPUT FILE */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
STEP02 IF (STEP01 = 0) THEN
STEP03 RUN TASK (PGM=DMRTDYN
PARM=(C"ALLOC DSN=&VPSDSN,DISP=(OLD,DELETE)"
F'-1'
C"UNALLOC DSN=&VPSDSN"))
EIF
EXIT

```

Chapter 9. Secure Plus for z/OS

Connect:Direct Secure Plus for z/OS Overview

IBM Connect:Direct Secure Plus for z/OS provides enhanced security for IBM Connect:Direct. It uses cryptography to secure data transmission with the security protocol you choose.

Security Concepts

Cryptography is the science of keeping messages private. A cryptographic system uses encryption keys between two trusted communication partners. These keys encrypt and decrypt information so that the information is known only to those who have the keys.

There are two kinds of cryptographic systems: *symmetric-key* and *asymmetric-key*. Symmetric-key (or secret-key) systems use the same secret key to encrypt and decrypt a message. Asymmetric-key (or public-key) systems use one key (public) to encrypt a message and a different key (private) to decrypt it. Symmetric-key systems are simpler and faster, but two parties must somehow exchange the key in a secure way because if the secret key is discovered by outside parties, security is compromised. Asymmetric-key systems, commonly known as public-key systems, avoid this problem because the public key may be freely distributed, but the private key is never transmitted.

Cryptography provides information security as follows:

- **Authentication** verifies that the entity on the other end of a communications link is the intended recipient of a transmission.
- **Non-repudiation** provides undeniable proof of origin of transmitted data.
- **Data integrity** ensures that information is not altered during transmission.
- **Data confidentiality** ensures that data remains private during transmission.

Connect:Direct Secure Plus enables you to implement multiple layers of security. Select from two security protocols to use to secure data during electronic transmission: Transport Layer Security (TLS) or Secure Sockets Layer protocol (SSL). Depending on the security needs of your environment, you can also validate certificates using the IBM® External Authentication Server application.

Note:

- TLS implies versions TLS1.0, TLS1.1 and TLS1.2
- SSL implies version SSL3.0

IBM Connect:Direct also allows you to implement security and encryption as appropriate for your environment. For example, if your company has a universal policy you want to enforce, elect to encrypt all files at all times. To provide flexibility, allow a trading partner to override security settings by specifying any of the following conditions:

- Turning Connect:Direct Secure Plus for z/OS on or off for a particular session
- Specifying one or more ciphers for encryption instead of the default cipher suites
- Encrypting only the control block information contained in Function Management Headers (FMHs), such as a user ID, password, and filename, instead of the files being transferred if performance is a factor.



CAUTION: The more flexible the configuration is, the less secure the environment becomes. Observe caution when considering the flexibility of configuration.

Security Protocols

Before you configure Connect:Direct Secure Plus for z/OS, determine the protocol you and your trading partners will use to secure communications sessions. For planning information, see [SSL and TLS Prerequisites](#).

Transport Layer Security Protocol and Secure Sockets Layer Protocol

The TLS and the SSL protocols use certificates to exchange a session key between the node that initiates the data transfer process (the primary node, or PNODE) and the other node that is part of the communications session (the secondary node, or the SNODE). A certificate is an electronic document that associates a public key with an individual or other entity. It enables you to verify the claim that a given public key belongs to a given entity. Certificates can be self-issued or issued by a certificate authority (see [Self-Signed and CA-Signed Certificates](#) for details). When a certificate authority (CA) receives an application for a certificate, the CA validates the applicant's identity, creates a certificate, and then signs the certificate. You use the CA signature to authenticate CA-issued trading partner certificates. A certificate authority issues and revokes CA-issued certificates. Self-signed certificates are created and issued by the owner of the certificate, who must export the certificate in order to create a trusted root for the certificate and supply the trusted root of the self-signed certificate to the partner in a connection.

Levels of Security

The TLS and SSL protocols provide three levels of security:

- During the first level of authentication called server authentication, the site initiating the session (PNODE) requests a certificate from its trading partner (SNODE), during the initial handshake. The SNODE returns its ID certificate (read from its key certificate file) and the PNODE authenticates it using one or more trusted root certificates stored in a trusted root certificate file (the name and location of which are specified in the remote node record for that specific trading partner in the PNODE's Connect:Direct Secure Plus parameter file). Root certificates are signed by a trusted source—either a public certificate authority, such as Thawte, or by the trading partner acting as its own CA. If the ID certificate from the SNODE cannot be validated using any root certificate found in the trusted certificate file, or if the root certificate has expired, the PNODE terminates the session. IBM Connect:Direct writes entries to the statistics logs of both nodes, and the session is aborted.
- The second level of authentication is optional and is called client authentication. If this option is enabled in the SNODE's Connect:Direct Secure Plus parameter file definition for the PNODE, the SNODE will request a certificate from the PNODE, and authenticate it using the information in its trusted root certificate file. If this authentication fails, the SNODE terminates the session and IBM Connect:Direct writes information about the failure to the statistics logs of both nodes.

In order to perform this security check, the trading partner must have a key certificate file available at its site and the IBM Connect:Direct server must have a trusted root file that validates the identity of either the Certificate Authority (CA) who issued the key certificate or the entity that created the certificate, if it is self-signed.

- The third authentication level is also optional and consists of validating the PNODE's certificate common name. When the security administrator enables client authentication, they can also specify the common name (CN) contained in the PNODE's ID certificate. During client authentication, the SNODE compares the common name it has specified for the PNODE in its Connect:Direct Secure Plus parameter file with the common name contained in the certificate sent by the PNODE. If the compare fails, that is, the information is not identical, the SNODE terminates the session, and IBM Connect:Direct writes information about the failure to the statistics logs of both nodes.

Areas of Security

The SSL and TLS protocols provide data security in the following areas:

- Authentication—Certificates used in the SSL or TLS session are digitally signed by a CA through an established procedure to validate an applicant's identity or digitally signed by the certificate owner-issuer. The SSL or TLS protocol validates the digital signature of the certificate being used.
- Proof of data origin and data integrity validation—The certificate provides proof of origin of electronic transmission and encryption validates data integrity. Message digest (hashing) and encrypting the message digest ensure that the data is not altered.
- Data confidentiality—Cipher suites encrypt data and ensure that the data remains confidential. The sending node converts sensitive information to an unreadable format (encryption) before it is sent to the

receiving node. The receiving node then converts the information back into a readable format (decryption).

TLS Features

Both the SSL protocol and the TLS protocol manage secure communication in a similar way. However, TLS provides a more secure method for managing authentication and exchanging messages, using the following features:

- While SSL provides keyed message authentication, TLS uses the more secure Key-Hashing for Message Authentication Code (HMAC) to ensure that a record cannot be altered during transmission over an open network such as the Internet.
- TLS defines the Enhanced Pseudorandom Function (PRF), which uses two hash algorithms to generate key data with the HMAC. Two algorithms increase security by preventing the data from being changed if only one algorithm is compromised. The data remains secure as long as the second algorithm is not compromised.
- While SSL and TLS both provide a message to each node to authenticate that the exchanged messages were not altered, TLS uses PRF and HMAC values in the message to provide a more secure authentication method.
- To provide more consistency, the TLS protocol specifies the type of certificate that must be exchanged between nodes.
- TLS provides more specific alerts about problems with a session and documents when certain alerts are sent.
- If you are required to have a FIPS 140-2-validated solution, a FIPS-mode of operation is available in Connect:Direct for the TLS protocol.

Planning for System SSL in FIPS Mode

Beginning with IBM z/OS Version 1 Release 11, System SSL provides the capability to execute securely in FIPS 140-2 mode. To this end, System SSL can run in either "FIPS mode" or "non-FIPS mode." By default, System SSL runs in non-FIPS mode and must be configured to run in FIPS mode. While executing in FIPS mode, System SSL continues to take advantage of the CP Assist for Cryptographic Function (CPACF) when it is available. System SSL checks for the application of certain restrictions. For information about System SSL in FIPS Mode, see [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#).

Connect:Direct for z/OS can request for System SSL to be placed into FIPS mode with the appropriate System SSL API calls. The Connect:Direct for z/OS FIPS initialization parameter attempts to place System SSL into FIPS mode. This initialization parameter instructs Connect:Direct FTP+ to initiate FIPS mode by using the appropriate System SSL API call, `gsk_fips_state_set`. Connect:Direct FTP+ issues the SITA195I message to indicate a successful request. However, if the request is not successful, Connect:Direct FTP+ terminates until the problem is resolved. For more information about FIPS-mode errors, see ["Troubleshooting" on page 817](#). For more information about the FIPS initialization parameter, see *IBM Connect:Direct for z/OS Administration Guide*. For more information about special considerations for FIPS-mode, see *IBM Connect:Direct for z/OS Release Notes*.

Secure Protocol and Security Mode

Connect:Direct for z/OS and SecurePlus have implemented support for several new Secure protocols including SSLv3.0, TLSv1.0, TLSv1.1 and TLSv1.2 and a new mode called Security Mode. The Secure protocol and the Security mode work hand in hand and the mode will apply further restrictions as to Secure protocol can be used.

The SecurePlus Admin tool, SPAdmin, will allow multiple protocols to be enabled and depending on the needs and settings of the Remote Connect:Direct negotiate to the highest supported Secure protocol. For example, if the Local parameter record enables all Secure protocols if the Remote Connect:Direct has enabled TLSv1.2 then TLSv1.2 will be selected. However, if the Remote Connect:Direct has not enabled

TLSv1.2 or does not support TLSv1.2 then the highest Secure protocol that Remote does support will be selected.

The Security mode restricts not only which Secure protocol that can be enabled and/or selected, it may also restrict cipher suite selection, certificate key strength and encryption algorithms. SPAdmin provides field level help for each Secure protocol and Security mode by pressing PF1 key while the cursor is positioned on the field.

FIPS mode requires the Secure protocol to be a minimum of TLSv1.0 and disables the use of SSLv3.0. FIPS mode also disables or ignores certain cipher suites.

SP800-131a in Transition Mode

SP800-131a is implemented in 2 modes, transition vs. strict mode. With SP800-131a Transition mode, the follow is recommended by is not enforced by SecurePlus:

- FIPS mode must be enabled - DES and RC2 cipher algorithms are disabled
- MD5 signature algorithms are disabled
- RSA and DSA certificates with key length less than 1024-bits are disabled
- Support for TLSV1.2 is enabled
- SSLV3, TLSV1.0 and TLSV1.1 are allowed but should be disabled
- Non-compliant TLS cipher suites are disabled

SP800-131a in Strict Mode

SP800-131a Strict mode will enforce these restrictions and if any parameter is outside of those parameter the SSL/TLS handshake will fail:

- FIPS mode must be enabled - DES, RC2 and two-key Triple DES cipher algorithms are disabled
- MD5 and SHA1 signature algorithms are disabled
- RSA and DSA certificates with key length less than 2048-bits are disabled
- EC certificates with key length less than 224-bits are disabled
- Protocol must be TLSV1.2
- SLV3, TLSV1.0, and TLSV1.1 are disabled

NSA Suite B 128bit Mode

The following restrictions apply to NSA Suite B in 128bit mode is enabled and further restricted in 192bit mode:

- Certificates must be ECC using elliptic curve secp256r1 or secp384r1
- Protocol must be TLSV1.2, all others are disabled
- Cipher algorithm must be AES-128
- Key exchange algorithm must be ECDH
- Digital signature algorithm must be ECDSA
- Hashing algorithm must be SHA256
- Cipher suites allowed for NSA Suite B 128 bit are:
 - TLS_ECDHE_ECDSA_W_AES_128_CBC_SHA256 (C023)
 - TLS_ECDHE_ECDSA_W_AES_128_GCM_SHA256 (C02B)

Note: To use **Suite B** and an **ECC** certificate special authorization and setup is required. For more information, see the *System SSL Programing Guide*.

NSA Suite B 192bit Mode

- Certificates must be ECC using elliptic curve secp384r1

- Protocol must be TLSV1.2, all others are disabled
- Cipher algorithm must be AES-256
- Key exchange algorithm must be ECDH
- Digital signature algorithm must be ECDSA
- Hashing algorithm must be SHA384
- Cipher suites allowed for NSA Suite B 192 bit are
 - TLS_ECDHE_ECDSA_W_AES_256_CBC_SHA384 (C024)
 - TLS_ECDHE_ECDSA_W_AES_256_GCM_SHA384 (C02C)

Secure Cipher Suite

The Secure protocol and the Security mode affect which cipher suite is selectable during an SSL handshake. The order in which the SPAdmin tool presents the cipher suite is dependent on the Secure protocol and Security mode that is enabled. The table below illustrates which cipher suites are valid for each protocol and mode. The SPAdmin tool allows you to enable or select ten from the list. However, during the SSL handshake any selection that is not valid for the Secure protocol and Security mode selected are ignored. At least one cipher must be valid for the protocol and mode enabled in both the PNODE and SNODE list.

Cipher ID	Name	Deprecated	TLS 1.2	TLS 1.1	TLS 1.0	SSL 3.0	FIPS140-2 >= TLS1.0	SP800-131 A TLS 1.2 Only	Suite B 128 TLS 1.2 Only	Suite B 192 TLS 1.2 Only	Kx	Au	Enc	Bits	Mac	Mac Bits	Mod e Stre ngth
0x0C02C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384		X				X	X	X	X	ECDHE	ECDSA	AES_256_GCM	256	SHA384	384	2
0x0C024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384		X				X	X			ECDHE	ECDSA	AES_256_CBC	256	SHA384	384	1
0x0C00A	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA		X				X	X			ECDHE	ECDSA	AES_256_CBC	256	SHA	160	1
0x0C02B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256		X				X	X	X		ECDHE	ECDSA	AES_128_GCM	128	SHA256	256	2
0x0C023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256		X				X	X			ECDHE	ECDSA	AES_128_CBC	128	SHA256	256	1
0x0C009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA		X				X	X			ECDHE	ECDSA	AES_128_CBC	128	SHA	160	1
0x0C007	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA		X								ECDHE	ECDSA	RC4_128	128	SHA	160	0
0x0C008	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA		X				X	X			ECDHE	ECDSA	3DES_EDE_CBC	168	SHA	160	1
0x0C006	TLS_ECDHE_ECDSA_WITH_NULL_SHA		X								ECDHE	ECDSA	NULL	0	SHA	160	0
0x0C030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384		X				X	X			ECDHE	RSA	AES_256_GCM	256	SHA384	384	2
0x0C028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384		X				X	X			ECDHE	RSA	AES_256_CBC	256	SHA384	384	1
0x0C014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA		X				X	X			ECDHE	RSA	AES_256_CBC	256	SHA	160	1
0x0C02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256		X				X	X			ECDHE	RSA	AES_128_GCM	128	SHA256	256	2
0x0C027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256		X				X	X			ECDHE	RSA	AES_128_CBC	128	SHA256	256	1
0x0C013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA		X				X	X			ECDHE	RSA	AES_128_CBC	128	SHA	160	1
0x0C011	TLS_ECDHE_RSA_WITH_RC4_128_SHA		X								ECDHE	RSA	RC4_128	128	SHA	160	0
0x0C012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA		X				X	X			ECDHE	RSA	3DES_EDE_CBC	168	SHA	160	1
0x0C010	TLS_ECDHE_RSA_WITH_NULL_SHA		X								ECDHE	RSA	NULL	0	SHA	160	0
0x0009D	TLS_RSA_WITH_AES_256_GCM_SHA384		X				X	X			RSA	RSA	AES_256_GCM	256	SHA384	384	2
0x0003D	TLS_RSA_WITH_AES_256_CBC_SHA256		X				X	X			RSA	RSA	AES_256_CBC	256	SHA256	256	1
0x00035	TLS_RSA_WITH_AES_256_CBC_SHA		X	X		X	X	X			RSA	RSA	AES_256_CBC	256	SHA	160	1
0x0009C	TLS_RSA_WITH_AES_128_GCM_SHA256		X				X	X			RSA	RSA	AES_128_GCM	128	SHA256	256	2
0x0003C	TLS_RSA_WITH_AES_128_CBC_SHA256		X				X	X			RSA	RSA	AES_128_CBC	128	SHA256	256	1
0x0002F	TLS_RSA_WITH_AES_128_CBC_SHA		X	X		X	X	X			RSA	RSA	AES_128_CBC	128	SHA	160	1
0x00005	TLS_RSA_WITH_RC4_128_SHA		X	X		X					RSA	RSA	RC4_128	128	SHA	160	0
0x00004	TLS_RSA_WITH_RC4_128_MD5			X	X	X					RSA	RSA	RC4_128	128	MD5	128	0

IBM Connect:Direct Access to System Resources for SSL or TLS

Before you can configure the Connect:Direct Secure Plus records to use the SSL or TLS protocol, you must ensure that the IBM Connect:Direct components have access to the resources listed in the following table.

Component	Access to Resource
IBM Connect:Direct	z/OS UNIX System Services or POSIX environment, must be installed and set up for IBM Connect:Direct access.
	Access to the following APF-authorized IBM system libraries through the STEPLIB or LINKLST: <ul style="list-style-type: none"> • CEE.SCEERUN and CEE.SCEERUN2 (language environment) • CBC.SCLBDLL (C/C++ environment) • SYS1.SIEALNKE for IBM z/OS (System SSL Environment)
	For end-user server certificates with ICSF private key type: <ul style="list-style-type: none"> • The ICSF application must be running on the same environment as IBM Connect:Direct. • The Crypto Hardware device and the ICSF application must be running and accessible by IBM Connect:Direct.
IBM Connect:Direct User ID (under which DTF runs)	Address space uses the maximum sockets (and other TCP/IP configurations) assigned by the UNIX System Services
	OMVS access
	A default UNIX directory
	UPDATE authority to the BPX.SERVER facility
	READ authority to the CSFSERV facility class
SSL/TLS	Access to key database or key ring as follows: <ul style="list-style-type: none"> • gskkyman key database • RACF, CA-ACF2, or CA-Top Secret key ring
	Access to the following APF-authorized IBM system library through the STEPLIB or LINKLST: <ul style="list-style-type: none"> • SYS1.SIEALNKE for IBM z/OS (System SSL Environment)
	Permission to read IBM Connect:Direct key ring that is created using RACDCERT, as follows: <ul style="list-style-type: none"> • Define the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources with universal access of None. • Grant the IBM Connect:Direct User ID read access to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources in the FACILITY class. • Activate the FACILITY general resource class. • Refresh the FACILITY general resource class.

Component	Access to Resource
IBM Connect:Direct User ID key database or key ring	Verification of other certificates requires access to the trusted root certificate of either: <ul style="list-style-type: none"> • A trusted CA certificate • Copy of a self-signed trusted certificate without private key
Connect:Direct Secure Plus Parameter file	Your node must have a remote node record in the parameter file of each of your trading partners that will use secure connections.

Connect:Direct Secure Plus Tools

Connect:Direct Secure Plus for z/OS consists of three components: the Administration Tool (Admin Tool), the parameter file, and the access file. The following sections describe these components and their purpose within Connect:Direct Secure Plus for z/OS.

Admin Tool

The Admin Tool enables you to configure and maintain the Connect:Direct Secure Plus environment. The Admin Tool is the only interface for creating and maintaining the Connect:Direct Secure Plus parameter file. Other operating system utilities and editing tools do not work.

The Admin Tool uses a native ISPF interface. The following sample illustrates the native ISPF interface display of the Create/Update Panel for SSL/TLS parameters. You can change the ISPF settings (Option 0) for the action bar choices and point-and-shoot fields. Changing these settings to fit your personal preferences can enhance operation and navigation in the Admin Tool.

Note: The SP Admin tool is unable to open a Secure parameter file created prior to release 5.2.00. See [“DGASCONV – Secure Parameter File Conversion Utility”](#) on page 649 for more information.

Native ISPF User Interface

The Native ISPF User Interface uses the standard mouse method of pointing and clicking to position the cursor to select an item, or to access a different panel. You can also tab to move the cursor from field to field within a panel.

Panels that make up the Secure+ Admin tool can contain the following components:

- The Panel Selection bar lists the other panels you can access from the current panel. These panels are listed from left to right in the typical order you would enter information. For example, the first panel listed on the sample panel, Node Identification, contains basic information about the node that already exists in the network map, such as its node name and communication information (TCP/IP address). The Secure+ Create/Update Panel - SSL/TLS Parameters panel is the next panel used to enter protocol information for communicating with the current node, followed by the EA Parameter panel if External Authentication is implemented.
- A selectable field is one in which you can either enter new information or edit existing information. On the sample panel, the fields you can enter information are those related to enabling SSL or TLS and the common parameters. You enter information in the underlined area next to the selectable field.
- The other selectable fields on the Secure+ Create/Update Panel - SSL/TLS Parameters panel take you to different panels where you can continue entering information related to the enabled protocol. For example, to choose cipher suites, you would select Cipher Suites by moving the cursor to that label and pressing Enter. A new panel displays allowing you to select and prioritize the ciphers you want to use.
- Non-selectable fields contain reference information that is display-only, such as the node name, and existing certificate and cipher information.
- Action selection allows you to accept and save the information entered on the panel in the Connect:Direct Secure Plus parameter file or cancel. After you select an action, the panel which was displayed prior to the current panel, is redisplayed.

SPAdmin Navigations

Much work has been done to improve the user experience with the SPAdmin tool to allow all panel movement and configuration updates to be performed without the use of a mouse. There are many new line commands or fast path commands that can be entered from the command line (==>) to perform the function of the SPAdmin tool. All pull down menu options have been define fast path commands so that the pull down menu is not required to perform those function, example FO can be use perform the File Open function. The table below defines all of the menu option fast path commands.

Menu option	Fast Path command
File New	FN
Edit Create/Update	EM
File Open	FO
File Close	FC
File Info	FI
Save Active	FS
Save As	FA
File Exit	FE
Edit Create/Update	EM
Edit Delete Record	ED
Edit Options	EO
Help	HH
Help Custom	HC
Help TLS/SSL	HS

While in create, update or view mode edit of a Parameter file record, several fast path commands exist so that many of the point-n-shoot field s are not required. For example, OK can be entered from the command line to perform the OK point-n-shoot function.

Field Option	Fast Path command
Security Options	SEC
EA Parameters	EA
SSL/TLS Parameters	SSL
OK	OK
Cancel	CAN

Several point-n-shoot fields still exist, such as those listed below, and navigation to those can be made easier using the ISPF Option “Tab to point-and-shoot fields.” Using the TAB key you can jump easily from field to field within the SPAdmin tool. Make sure that the option you place the cursor on is the function you intend to update.

Listed below are some of the point-n-shoot fields:

- Certificate Label
- Cipher Suites
- Certificate Pathname
- Certificate Common Name
- External Auth Server Def
- External Auth Server Address
- External Auth Server Port

Connect:Direct Secure Plus Parameter File

The Connect:Direct Secure Plus parameter file contains information that determines the protocol and encryption method used during security-enabled IBM Connect:Direct operations. To configure Connect:Direct Secure Plus, each site must have a parameter file that contains one local node record and a remote node record for each trading partner who uses Connect:Direct Secure Plus to perform a secure connection. The local node record defines the most commonly used security and protocol settings at the site and can be used as a default for one or more remote node records. Each remote node record defines the specific security and protocol used by a trading partner.

For additional security, the parameter file is stored in an encrypted format. The information used for encrypting and decrypting the parameter file (and private keys) is stored in the Connect:Direct Secure Plus access file.

Passwords are protected in the TCQ and AUTH files by encrypting them with Connect:Direct Secure Plus's proprietary "Polyalphabetic Substitution Cipher" which is a weak encryption. A stronger encryption algorithm, TDESCBC112, can be used if you add a .PASSWORD record to the Connect:Direct Secure Plus parameter file. After you create this record, enable the Strong Password Encryption (SPE) feature, and restart Connect:Direct Secure Plus, SPE protects Connect:Direct Secure Plus passwords stored in the TCQ and AUTH files with the stronger algorithm. For more information on using this feature, refer to [Implementing Strong Password Encryption](#).

Access File

The Connect:Direct Secure Plus access file is generated automatically when you create the Connect:Direct Secure Plus parameter file for the first time. You type a passphrase when you first initialize Connect:Direct Secure Plus. This passphrase is used to generate the keys necessary to encrypt and decrypt the entries in the Connect:Direct Secure Plus parameter file. The passphrase itself is not retained.

Your Connect:Direct Secure Plus administrator must secure the access file. This requires full create and update capability. The IBM Connect:Direct server must have read authority. To maintain a secure access file, the general user community should not have access permission.

This file can be secured with any available file access restriction tools. Availability of the access file to unauthorized personnel can compromise the security of data exchange.

Parameter and Access file best practices

Some best practices to manage the *Parameter* and *Access* files are listed below:

- Always take a backup of the *Parameter* and *Access* file before applying updates to the parameter file.
- Save your changes often by performing a **Save As** or **Save Active** function.
- After a **Save As** or **Save Active** function, close the current *Parameter* file and open it again if more updates are required.

Control Center

Once you have created your Connect:Direct Secure Plus parameter file, you can use IBM Control Center to perform the following functions when implementing the SSL or TLS protocol:

- Create and update a remote node
- Update the local node
- Add and update certificates
- Create an alias node
- Select cipher suites
- Delete a remote node

To perform these functions, you must have a Control Center user ID with IBM Connect:Direct administration authority including privileges to read and write to the Connect:Direct Secure Plus parameter file.

For more information, see *Customizing Levels of IBM Connect:Direct Functional Authority* in the *IBM Connect:Direct for z/OS Administration Guide*. For more information on how to perform these functions, see the documentation for Control Center.

Prerequisites

Before you configure Connect:Direct Secure Plus for z/OS, ensure that you complete the following tasks.

Expert Security Administrator

The instructions and information provided to assist you in implementing the SSL/TLS protocol assume that you have an expert z/OS security administrator who is familiar with the terms associated with digital certificates and has experience using the tools required to generate and manage certificates, including:

- UNIX System Services
- IBM ICSF application and Crypto Hardware device
- System security applications, for example, gskkyman, RACF, CA-Top Secret, or CA-ACF2
- Security terminology associated with digital certificates (see [Terminology and Security Applications for SSL and TLS Certificates](#))
- Working knowledge of the IBM Connect:Direct application and its environment

IBM Connect:Direct ISPF Libraries in TSO

To ensure that you can perform Connect:Direct Secure Plus parameter file functions and generate the SAVE AS JCL for the Connect:Direct Secure Plus parameter file, you must allocate the same release of the following IBM Connect:Direct ISPF libraries in your TSO session before you try to perform Connect:Direct Secure Plus parameter file functions and generate and submit the Save As JCL as described in [Connect:Direct Secure Plus Operation Enablement and Validation](#) or the Save Active JCL as described in [Connect:Direct Secure Plus Maintenance](#):

- \$CD.SDGAISPC (must be allocated as SYSPROC)
- \$CD.SDGLINK (to ISPLLIB)
- \$CD.SDGAPENU (to ISPPLIB)
- \$CD.SDGASENU (to ISPSLIB)
- \$CD.SDGMENU (to ISPMLIB)

If these required libraries have not been allocated, or have been allocated incorrectly, when you perform the save and submit procedure, the JCL for the SAVE AS job is not generated, and you have to repeat the configuration tasks. For more information on the required libraries and how to allocate them, see the *IBM Connect:Direct for z/OS Configuration Guide* and the *Program Directory for IBM Connect:Direct for z/OS*.

Security Requirements of Trading Partners

Security planning is a collaborative effort between you and your trading partners. You must know the expectations of your trading partners and plan your security implementation to meet these requirements. Consider the following guidelines for configuring communications sessions using the SSL or TLS protocol:

- You must acquire the certificates before you configure Connect:Direct Secure Plus.
- Determine whether you and your trading partner will use self-signed certificates or certificates signed by a Certificate Authority.
- Determine whether to use client authentication.
- Using the External Authentication Server application in conjunction with Connect:Direct Secure Plus to validate the other node's certificate for a secure session requires the following:
 - Using the TLS or SSL protocol for connections to the External Authentication Server
 - Enabling client authentication in remote node records so that the SNODE can validate the PNODE certificate
 - Exchanging certificates between Connect:Direct Secure Plus for z/OS and the External Authentication Server node

Implementation Plan for Connect:Direct Secure Plus

After you identify your security administrator and determine the security requirements of your trading partners, review [SSL and TLS Prerequisites](#) for protocol-specific configuration information.

Worksheets

Before you configure Connect:Direct Secure Plus for z/OS, complete the worksheets in [Configuration Worksheets](#). Use this information to configure the local and remote nodes to use Connect:Direct Secure Plus for z/OS.

Connect:Direct Secure Plus for z/OS Documentation

The *IBM Connect:Direct Secure Plus for z/OS Implementation Guide* describes how to implement peer-to-peer security into IBM Connect:Direct operations with Connect:Direct Secure Plus. This document includes information to plan, configure, and use Connect:Direct Secure Plus. The *IBM Connect:Direct Secure Plus for z/OS Implementation Guide* is for programmers and network operations staff who install and maintain Connect:Direct Secure Plus.

This guide assumes knowledge of IBM Connect:Direct, including its applications, network, and environment and security policies and applications used in your environment.

Task Overview

The following table guides you to the information required to perform Connect:Direct Secure Plus tasks.

Task	Reference
Understanding Connect:Direct Secure Plus and assessing your security requirements	Connect:Direct Secure Plus for z/OS Overview
Planning to use the TLS or SSL protocol	SSL and TLS Prerequisites Certificate Parameter Definitions
Navigating the Secure+ Admin Tool and populating the parameter file	Admin Tool and Parameter File Usage Configuration for a Secure Connection between z/OS and OpenVMS Nodes

Task	Reference
Setting up local and remote node records for the SSL or TLS protocol	Manual Creation of the SSL or TLS Parameter File Local Node Record Imported from Network Map Configuration Remote Node Record Imported from Network Map Configuration Configuration Worksheets Configuration for a Secure Connection between z/OS and OpenVMS Nodes
Configuring special-purpose remote node records to perform one of the following functions: <ul style="list-style-type: none"> • Validate certificates using the Authentication Server application • Block nonsecure TCP API connections • Secure passwords at rest within the IBM Connect:Direct TCQ and AUTH files. 	Additional Configuration Options
Saving the parameter file the first time after creating it and preparing Connect:Direct Secure Plus for operation	Connect:Direct Secure Plus Operation Enablement and Validation
Validating and testing connections by session	Connect:Direct Secure Plus Operation Enablement and Validation
Performing exception processing by overriding Connect:Direct Secure Plus settings in the PROCESS statement	Override Settings in IBM Connect:Direct Processes
Maintaining the Connect:Direct Secure Plus parameter file and individual remote nodes	Connect:Direct Secure Plus Maintenance
Viewing Connect:Direct Secure Plus statistics	Connect:Direct Secure Plus Statistics
Understanding error messages and resolving errors	Troubleshoot

Plan Your Implementation of the SSL or TLS Protocol

Before you configure Connect:Direct Secure Plus, review the following concepts, requirements, terms, and tool descriptions to ensure that you have the resources and information necessary to implement the Transport Layer Security (TLS) protocol or Secure Sockets Layer (SSL) protocol.

For more information on using the TLS protocol and NIST requirements with z/OS, see the IBM publication **z/OS Cryptographic Services System SSL Programming (SC14-7495-00)**.

Transport Layer Security Protocol and Secure Sockets Layer Protocol

The Transport Layer Security protocol (TLS) and the Secure Sockets Layer (SSL) protocol use certificates to exchange a session key between the node that initiates the data transfer process (the primary node, or PNODE) and the other node that is part of the communications session (the secondary node, or the SNODE). A certificate is an electronic document that associates a public key with an individual or other entity. It enables you to verify the claim that a public key belongs to an entity. Certificates can be self-issued or issued by a certificate authority (CA). See [Self-Signed and CA-Signed Certificates](#). When a CA

receives an application for a certificate, it validates the applicant's identity, creates and signs certificate. A CA issues and revokes CA-issued certificates. Self-signed certificates are created and issued by the owner of the certificate, who must export the certificate in order to create a trusted root for the certificate and supply the trusted root of the self-signed certificate to the partner in a connection.

External Authentication Server validates certificates during an SSL or TLS session. Use the application to configure certificate chain validation, including the option to validate certificates against one or more Certificate Revocation Lists (CRLs) stored on an LDAP server. You can also configure the application to return attributes associated with the incoming certificate, such as group information, stored on an LDAP server. See *Sterling External Authentication Server Implementation Guide* for information.

To use External Authentication Server, configure your application to connect to the host name and port where the External Authentication Server application resides and specify a certificate validation definition. See the instructions for creating the Connect:Direct Secure Plus parameter file manually or using the network map for the TLS or SSL protocols for instructions to create the remote node record for the External Authentication Server application (.EASERVER).

FIPS 140-2 Mode for the TLS Protocol

Enhanced security is available for Connect:Direct using System SSL FIPS mode available in IBM z/OS Version 1 Release 11 to meet FIPS 140-2 criteria. FIPS-mode operation is available only for the TLS protocol. For more information, see [“Planning for System SSL in FIPS Mode” on page 751](#).

TLS or SSL Protocol Processing

After you configure Connect:Direct Secure Plus, you are ready to exchange data securely with other security-enabled IBM Connect:Direct nodes. Data is securely exchanged between two nodes using the protocol defined in the parameter file.

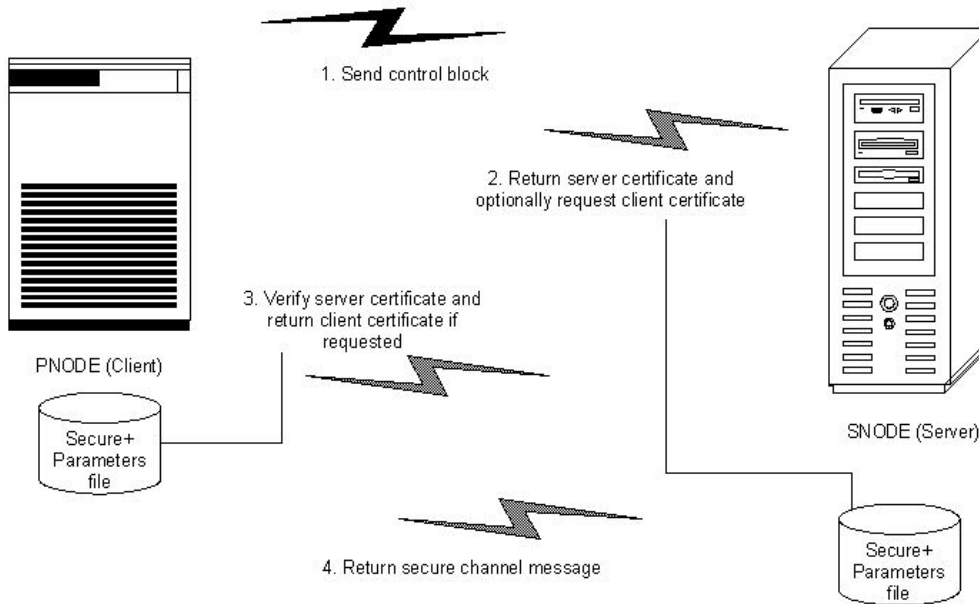
Note: You can implement the protocol you want to use for all data transfers or on a Process-by-Process basis. To specify a protocol each time you submit a Process, you must disable the protocol (but allow overrides) when you create the local and remote nodes in the Connect:Direct Secure Plus parameter file, and then specify it in the PROCESS statement using the SECURE parameter. For more information, see [Override Settings in IBM Connect:Direct Processes](#).

Connect:Direct Secure Plus Data Exchange

Data exchange consists of two processes: authentication and sending/receiving data. The TLS or SSL protocol data exchange process is described in the following sections.

Authentication

The following figure illustrates the authentication process using the TLS or SSL protocol:



The following steps occur during authentication:

1. The PNODE (client) sends a control block containing protocol (TLS or SSL) and cipher information to the SNODE (server). The SNODE confirms that it has a record defined in its Connect:Direct Secure Plus parameter file for the PNODE, and determines if a common cipher can be found and used for secure communication. Cipher suites are used to encrypt the data being sent between nodes. If the SNODE finds a record for the PNODE in its Connect:Direct Secure Plus parameter file and verifies it has a cipher defined in common with the PNODE, a common cipher is negotiated and the session continues.
2. The SNODE sends its ID certificate to the PNODE who confirms that it has a record defined in the Connect:Direct Secure Plus parameter file. Information for creating a public key is included. The PNODE verifies the ID certificate of the SNODE using the trusted root certificate file defined in its Connect:Direct Secure Plus parameter file, and generates a session key.
3. If client authentication is enabled on the SNODE, the SNODE requests an ID certificate from the PNODE. The PNODE sends its ID certificate defined in its Connect:Direct Secure Plus parameter file to the SNODE for verification against the trusted root certificate file specified in the SNODE's Connect:Direct Secure Plus parameter file. If a common name was also specified in the Connect:Direct Secure Plus parameter file for the PNODE, this name is used to verify the common name field of the PNODE's certificate.
4. The SNODE confirms that a secure environment is established and returns a secure channel message.

Customer Data Transmission

Once a Connect:Direct Secure Plus session has been established, all control blocks and customer data transmitted between the PNODE and SNODE are encrypted using the negotiated cipher.

Note: You can override certain security settings including what is encrypted during a session. If encrypting all data files is excessive in your environment, you can encrypt only the information necessary to establish a session and not the data files being transferred. For more information, see [Control Block and Data Encryption Default Override](#).

Self-Signed and CA-Signed Certificates

Determining the type of certificate to use for secure communications sessions and the method to generate the certificate is challenging. Self-signed certificates and digital certificates issued by certificate authorities offer advantages and disadvantages. You may also be required to use both types of certificates, depending on the security requirements of your trading partners. The following table compares the advantages and disadvantages of self-signed and CA-signed certificates.

Note: When System SSL is in FIPS mode, the certificates and certificate store have FIPS requirements. For more information about FIPS requirements, see [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#)

Type of Certificate	Advantages	Disadvantages
Self-signed certificate	No cost	Requires you to distribute your certificate, minus its private key, to each trading partner in a secure manner.
	Easy to generate	Difficult to maintain; anytime the certificate is changed, it must be distributed to all clients.
	Self-validated	Not validated by a third-party entity
	Efficient for small number of trading partners	Inefficient for large number of trading partners
CA-signed certificate	Not required to store the public key of trading partner The public key signed by the CA is exchanged at SSL negotiation and authenticated against the CA's Trusted Root Key, which is stored in the Trusted Root directory and the z/OS UNIX System Services key database or key ring of the Connect:Direct Secure Plus server.	Must be purchased from third-party vendor
	Tools used to generate certificates typically load the currently available CA certificates to the key database or key ring being generated, which means that you can connect your trading partner's certificates to the key ring to verify its trustworthiness.	
	Eliminates having to send your certificate to each trading partner	Trading partners must download digital CA-signed certificate used to verify the digital signature of trading partner public keys only if the CA certificate is not available
	Requires the remote client to store only the CA's digitally signed certificate (trusted key) in the Trusted Root directory	Must store the CA-signed certificate in the z/OS UNIX System Services key database or key ring and in the Trusted Root file
	No changes required on the trading partner's system if you recreate the CA signed certificate using the same CA	

Terminology and Security Applications for SSL and TLS Certificates

The following table defines the security terms associated with SSL and TLS certificates and applies them to communications sessions between a IBM Connect:Direct PNODE (client) and SNODE (server). The terms are listed in alphabetical order.

Term	Definition
CA-Signed Certificate	Digital document issued by a certificate authority that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. An identity certificate issued by a CA is digitally signed with the private key of the certificate authority.
Certificate Authority (CA)	An organization that issues digitally-signed certificates. The certificate authority authenticates the certificate owner's identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates belonging to users who are no longer authorized to use them. The CA digital signature is assurance that anybody that trusts the CA can also trust that the certificate that it signs is an accurate representation of the certificate owner.
Certificate Signing Request (CSR)	Message sent from an applicant to a certificate authority in order to apply for a digital identity certificate. Before creating a CSR, the applicant first generates a key pair, keeping the private key secret. The CSR contains information identifying the applicant (such as a directory name in the case of an X.509 certificate), and the public key chosen by the applicant. The CSR may be accompanied by other credentials or proofs of identity required by the certificate authority, and the certificate authority may contact the applicant for further information.
Key Database	<p>Database generated by the GSKKMAN utility for creating and managing public and private keys and certificates. Typically, the files in this database are password-protected to ensure that they are inaccessible to unauthorized users.</p> <p>Note: If System SSL is in FIPS mode, a FIPS keybase is required. See z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08.</p>
Key ring	<p>File that contains public keys, private keys, trusted roots, and certificates. A key ring is a collection of certificates that identify a networking trust relationship (also called a trust policy) and are stored in a database. Key rings are associated with specific user IDs, which can have more than one key ring. Key rings enable you to share key rings across multiple servers.</p> <p>Note: If System SSL is in FIPS mode, the key ring might have FIPS requirements. See z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08.</p>

Term	Definition
Private Key	String of characters used as the private, “secret” part of a complementary public-private key pair. The asymmetric cipher of the private key is used to sign outgoing messages and decrypt data that is encrypted with its complementary public key. Data that is encrypted with a Public Key can only be decrypted using its complementary Private Key. The private key is never transmitted.
Public Key	String of characters used as the publicly distributed part of a complementary public-private key pair. The asymmetric cipher of the public key is used to confirm signatures on incoming messages and encrypt data for the session key that is exchanged between server and client during negotiation for an SSL/TLS session. The public key is part of the ID (public key) certificate. This information is stored in the key certificate file and read when authentication is performed.
Self-Signed Certificate	Digital document that is self-issued, that is, it is generated, digitally signed, and authenticated by its owner. Its authenticity is not validated by the digital signature and trusted key of a third-party certificate authority. To use self-signed certificates, you must exchange certificates with all your trading partners.
Session Key	Asymmetric cipher used by the client and server to encrypt data. It is generated by the SSL software.

System Security Applications

The following table describes some system security applications available for generating certificates. Review the documentation for your security application for detailed instructions for generating certificates. See [Certificate Parameter Definitions](#), for more information on creating certificates using these tools.

Certificate Tool	Description
gskkyman	<p>IBM utility for creating and managing digital certificates and public and private keys stored in a key database. Files created using the gskkyman utility have the following default names:</p> <ul style="list-style-type: none"> • key.kdb = private key file • certreq.arm = Certificate Signing Request (CSR) file • cert.arm = public key file <p>The gskkyman utility loads currently available CA certificates to the key database.</p>

Certificate Tool	Description
Resource Access Control Facility (RACF)	<p>An IBM application that provides access control by identifying users to the system; verifying users of the system; authorizing access to protected resources; logging detected, unauthorized attempts to enter the system; and logging detected accesses to protected resources. The RACF utility can be used to create, store, and manage keys, digital self-signed or CA-signed certificates, and key rings. Because the RACF application can manage multiple key rings, certificates and key rings are added to the RACF database independently and then a certificate is associated with one or more key rings. For example, you can add the CA public key to your database and associate the certificates of your trading partners created by that CA with its public key.</p> <p>The RACF utility does not assign default names to the files you generate with it.</p>
Computer Associates Access Control Facility (CA-ACF2)	<p>Security application, similar to the RACF application, that enables you to authenticate users and to protect a variety of z/OS resources. You can generate, administer, and process certificate requests, export keys, and manage key rings.</p> <p>The CA-ACF2 application does not assign default names to the files you generate with it.</p>
CA-Top Secret	<p>Security application, similar to the RACF application, that protects your mainframe computer systems and data by controlling access to resources and enables you to generate, administer, and process certificate requests, export keys, and manage key rings.</p> <p>The CA-Top Secret application does not assign default names to the files you generate with it.</p>

General Requirements for Certificates

The certificate for the Connect:Direct Secure Plus for z/OS server defined in the local node record has the following general requirements:

- X.509 version 3 end-user server certificate that can interpret digital signatures and can encrypt and decrypt data
- Must be defined to the key database or key ring
- Must be stored in the key database or key ring
- Must have a private key
- Must be valid and not expired
- Must be signed by a CA or self-signed
- Must be set as default in the key database or key ring

Application-Specific Requirements

In addition to the general requirements for certificates, see [Certificate Parameter Definitions](#), for details on the minimum parameter definitions required for the security applications described in [Terminology and Security Applications for SSL and TLS Certificates](#).

Server Certificates and IBM Connect:Direct

To use the SSL or the TLS protocol to perform a secure connection, you must obtain a server certificate and set up IBM Connect:Direct to use certificates.

Note: An optional feature provides the ability to monitor certificates and issue automatic and on-demand warnings when certificates expire and are soon to expire within a specified number of days. Refer to *Checking the Validity of Certificates Used by Connect:Direct Secure Plus* in the *IBM Connect:Direct for z/OS Administration Guide*.

Server Certificate

Certificates require key settings that define the type of security to implement at your site, including authentication, non-repudiation, data integrity, and data confidentiality, as described in [Security Concepts](#). Although the security application that you use to create a digital certificate may use different terms to describe these security concepts (for example, digital signature, key encipherment, data encipherment, and non-repudiation), both self-signed certificates and certificate requests sent to a certificate authority must designate all these key usage items to ensure that Connect:Direct Secure Plus can use the certificates to perform the intended security functions.

You can use the following methods to obtain an X.509 version 3 server certificate:

- Your registration authority can contract with a formal certificate authority (CA) to obtain a server certificate. When you obtain the server certificate, you then import this certificate into the IBM System SSL toolkit key database or key ring.
- Your registration authority can create a self-signed private and public key using one of the system security applications described in [Terminology and Security Applications for SSL and TLS Certificates](#).
- Using one of the system security applications described in [Terminology and Security Applications for SSL and TLS Certificates](#), your registration authority can generate a certificate signing request (CSR) for submission to third-party Certificate Authority to obtain a CA-signed public key. You forward this certificate to a certificate authority to be signed. When you receive the signed certificate, you import this certificate into the IBM System SSL key database or key ring. Refer to the IBM documentation *IBM Cryptographic Services System Secure Sockets Layer Programming Guide and Reference* for details.

Setting Up IBM Connect:Direct to Use Certificates

About this task

Before using the TLS or SSL protocol, you must set up IBM Connect:Direct to use certificates.

Note: When System SSL is in FIPS mode, there might be additional requirements. See [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#).

To set up IBM Connect:Direct to use certificates:

Procedure

1. Ensure that the local IBM Connect:Direct node to be configured for the TLS or SSL protocol has either a key ring or a key database on the z/OS image that contains its certificate.
2. Record the following information on your local node record worksheet for use when you configure the local node record in the Connect:Direct Secure Plus parameter file:
 - Name of the key ring or full file name of the key database
 - Label of the certificate in your key ring or key database
 - Password used when the key database was created

Note: Key rings do not use passwords.

Note: A certificate can be designated as the default certificate in the key ring or key database. The *Secure parameter* file definition can specify the use of the default certificate.

3. If you are using a key database, issue the UNIX command **chmod 666** to ensure that IBM Connect:Direct has permission to read from and write to the key database.

Note: Write permission is not a requirement for normal functionality of IBM Connect:Direct Secure Plus. However, remote management of Secure Plus through IBM Control Center may require write permissions to insert and update certificates in the key database.

Using the SecurePlus Admin Tool and Populating the Parameter File

Use the following information to familiarize yourself with the Connect:Direct Secure Plus administration tool and to determine whether to use Quickstart to populate the parameter file from the network map or populate the parameter file manually.

Note: You can only use the Quick Start option the first time you create a parameter file, and if your IBM Connect:Direct TCP/IP netmap entries use standard IPV4 addresses.

Note: The Quick Start option is not recommended if the IBM Connect:Direct NETMAP defines a large partner community. This will likely generate updates as SecurePlus is required for only a few partners.

Starting the Administration Tool

About this task

Use the Administration (Admin) Tool to set up and maintain a Connect:Direct Secure Plus operation.

Procedure

1. To start the Admin Tool, do one of the following:
 - From the Connect:Direct Administrative Options Menu, select **Secure+** from the action bar and press Enter. Type **1** to select Secure+ Admin Tool and press Enter.
 - On the Connect:Direct Administrative Options Menu command line, type **S** and press Enter to select the Executive Secure Plus Commands option, then type **SA** on the command line and press Enter.

After the Admin Tool is initialized, the Secure+ Admin Tool: Main Screen is displayed.

2. To continue configuring Connect:Direct Secure Plus, refer to [Connect:Direct Secure Plus Configuration](#).

About the Admin Tool

When you start the Secure+ Admin Tool and open a parameter file, the panel displays all node records that are defined in the parameter file including a summary of the attributes for each node, unless you have filtered the records by node name using **Options** on the **Edit** menu.

The following table describes the fields that are displayed on the Secure+ Admin Tool: Main Screen, including field descriptions and valid values for each field.

Field Name	Field Description	Valid Values
LC	Line Command. For more information, see Node Record Maintenance .	U = Update node I = Insert node H = View History V = View node D = Delete node CL = Clone node
Node Name	Displays the node record name.	Node name

Field Name	Field Description	Valid Values
Type	Displays the current record type.	L = local node record R = remote node record A = alias record Note: Alias is valid only for remote records.
Secure+ Protocol	Displays the Secure+ protocol used.	Disabled * (default to local node) SSL TLSV10 TLSV11 TLSV12
Override	Displays the status of Override. Enabling Override in a remote node record allows values specified in the COPY statement or the PROCESS statement to override values set in the remote node record. For more information, see Override Settings in IBM Connect:Direct Processes .	Y = enabled N = disabled * = default to local node
Encryption	Indicates if data encryption is enabled.	Y = enabled N = disabled * = default to local node
External Auth	Identifies whether external authentication is enabled for the node. For more information, see “Adding a Remote Node Record for the External Authentication Server” on page 786	Y = enabled N = disabled * = default to local node
Client Auth	Identifies whether client authentication is enabled for the node.	Y = enabled N = disabled * = default to local node

Node-Specific Parameters

When you insert a node, in addition to entering the node name and Type (local or remote), you also must specify parameters on three additional screens that are accessed from the Secure+ Create Update Panel.

Security Options

The Security Options panel is displayed when you insert a node manually and select Security Options from the Create/Update panel. The following table lists the Connect:Direct Secure Plus parameters according to the type of record (local or remote) to which they apply.

Node Security Option	Valid for Local Node Record?	Valid for Remote Node Record?
Security Protocol	Yes	Yes

Node Security Option	Valid for Local Node Record?	Valid for Remote Node Record?
Security Mode	Yes	Yes
Enable Overrides	No	Yes
Alias Names	No	Yes
TCP Information	No	No
Note: When you create the Connect:Direct Secure Plus parameter file from the NETMAP, the TCP Information field is populated automatically; however, data in the TCP Information field of the remote record is not used to initiate IBM Connect:Direct communications sessions. IP address and port number are acquired only from the NETMAP.		

EA Parameters

The following table describes the parameters displayed when you select EA Parameters from the Create/Update Panel:

EA Parameters	Valid for Local Node Record?	Valid for Remote Node Record?
Enable External Auth	Yes. Not a good idea to enable this parameter in the local node record.	Yes. Valid only for .EASERVER remote node record.
External Auth Server Def	No	Yes. Valid only for .EASERVER remote node record.
External Auth Server Address	No	Yes. Valid only for .EASERVER remote node record.
External Auth Server Port	No	Yes. Valid only for .EASERVER remote node record.

SSL/TLS Parameters

The following table describes the parameters displayed when you select SSL/TLS Parameters from the Secure+: Create/Update panel.

Note: If System SSL is in FIPS mode, then TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode” on page 751](#)

Table 3.

Secure+ Create/Update Panel		
Option ==>		

Node Name: CD.ZOS.NODE	Type: L	(Local or Remote)
Security Options	EA Parameters	SSL/TLS Parameters
---	--	---

Enable Client Auth	Y	(Yes, No, Default to Local)
Enable Data Encrypt	Y	

Certificate Label:		
Cipher Suites:		
003B003C003D009C000100020003000400050006		
Certificate Pathname:	/u/user2/CDTEST.kdb	
Certificate Common Name:	-----	
OK Cancel		

SSL/TLS Parameters	Valid for the Local Node?	Valid for the Remote Node?
Enable Client Auth	Not a good idea to enable this parameter in the local node record.	Yes. Valid only for remote nodes that use the SSL or TLS protocol.
Enable Data Encrypt	Yes	Yes
Certificate Label	Yes Optional, if a default certificate is defined in the key store.	Yes, if required to over ride to Local record
Certificate Path name	Yes, required	No, can only be specified on the Local record
Certificate Common Name	Yes, but generally not specified on the Local record	Yes, commonly used in the Remote record

Admin Tool Navigation

Use the following standard function keys to navigate the Admin Tool:

Key	Function
PF8	Move forward
PF7	Move backward
PF12	Go back to the previous panel
PF3	Exit
Enter	Select an option

Admin Tool Help

You can access several types of Help information within the Secure+ Admin Tool as described in the following table:

Help	How to Access
General Help	From any Secure+ Admin Tool screen, select Help from the action bar and press Enter . Type I to select the general Help option.
Action Bar Help	Position the cursor on the action bar item and press PF1 .
Screen and Panel-Level Help	Position the cursor in any uneditable part of the screen or panel and press PF1 .
Field-Level Help	Position the cursor in the editable part of a field and press PF1 .

Connect:Direct Secure Plus Configuration

You must configure Connect:Direct Secure Plus before you begin using it for secure communications. You create and save a Connect:Direct Secure Plus parameter file that contains a single local node record and a remote node record for every trading partner that uses Connect:Direct Secure Plus. The way you populate the parameter file depends on your environment. [Parameter File Creation](#) and [Node Configuration](#) describe two common scenarios and the most effective method of creating and populating the parameter file and configuring the local and remote nodes records for each scenario.

Parameter File Creation

The configuration procedures are based on the scenarios described in this section. Use the following table to help you decide how to create a Connect:Direct Secure Plus parameter file.

Scenario	Method to Create parameter file	Result
<ul style="list-style-type: none">• First time to create a parameter file.• Large number of trading partners that use the same protocol.	Use Quickstart to copy the network map file and save it as the Connect:Direct Secure Plus parameter file. See Populating the Parameter File Using Quick Start .	<ul style="list-style-type: none">• File is created automatically with a local node record and a record for each remote node in the network map that uses the TCP or UDT protocol.• Configure Connect:Direct Secure Plus for all remote node records, including trading partners that do not use Connect:Direct Secure Plus.• Connect:Direct Secure Plus protocols are disabled for all records created from the network map.• Establishes default settings for most parameters in the local node record.
<ul style="list-style-type: none">• First time to create a parameter file.• Large number of trading partners.• Few trading partners use Connect:Direct Secure Plus.	Manually create a parameter file and add the local node record and remote node records. See Manual Parameter File Creation .	<ul style="list-style-type: none">• You create the local node record and a record for each remote node that uses Connect:Direct Secure Plus.• Reduces the number of records to configure.• No default settings are established for parameters in the local node record. You must define all settings.

Node Configuration

After you create and populate the parameter file, you decide how to configure the local node record. The method that you use to configure the local node record then determines how you configure remote node records.

Use the following table to help you decide how to configure the local node:

Scenario	How to Configure Node Records	Result
Most trading partners use the same protocol.	Enable the most commonly used protocol in the local node record. Depending on the protocol, see the relevant procedure in Local Node Record Imported from Network Map Configuration .	<ul style="list-style-type: none"> Enables the same protocol in all remote node records. You have to modify only the records for remote nodes that do not use the settings for the local node.
Most trading partners do not use Connect:Direct Secure Plus.	Disable the Connect:Direct Secure Plus protocols in the local node record and enable the Override parameter. Depending on the protocol, see one of the following procedures: <ul style="list-style-type: none"> Adding the Local Node Record to the Parameter File Manually for the SSL or TLS Protocol Configure remote node records only for those trading partners who use Connect:Direct Secure Plus.	<ul style="list-style-type: none"> You define default protocol settings in the local node record so remote nodes can use default values. You configure only those remote node records that use Connect:Direct Secure Plus.
Trading partners need to disable or enable security for a session.	Set OVERRIDE=Y in both the local and remote node records in the parameter files of both trading partners.	Either trading partner can disable or enable security for a particular session by setting the SECURE keyword in a Process or Copy statement. See Override Settings in IBM Connect:Direct Processes for more information.
Some trading partners use Connect:Direct Secure Plus and the External Authentication Server application.	Disable external authentication in the local node record and enable the Override parameter. Create a .EASERVER remote record. See Adding a Remote Node Record for the External Authentication Server .	<ul style="list-style-type: none"> You can enable external authentication only for those remote nodes that use it with Connect:Direct Secure Plus. You can verify certificates exchanged during an SSL or TLS session using the External Authentication Server application.
Nonsecure TCP API connections are not allowed to connect to a Connect:Direct for z/OS server.	Create a .CLIENT remote node record and disable override. See Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server .	<ul style="list-style-type: none"> Communications from nonsecure connections is not allowed.

To see a scenario for setting up a secure connection between a Connect:Direct Secure Plus for OpenVMS node and a Connect:Direct for z/OS node, see [Configuration for a Secure Connection between z/OS and OpenVMS Nodes](#). That topic provides a concrete example for defining a remote node record in both a Connect:Direct Secure Plus parameter file and a Connect:Direct Secure Plus for OpenMVS parameter file.

Populating the Parameter File Using Quick Start

About this task

The Quick Start option enables you to create a parameter file by importing information from the IBM Connect:Direct network map and requires that you configure all remote node records, including those of trading partners that do not use Connect:Direct Secure Plus.

Note: You can only use the Quick Start option the first time you create a parameter file, and if your IBM Connect:Direct TCP/IP netmap entries use standard IPV4 addresses.

To configure only the nodes that use Connect:Direct Secure Plus, refer to [Manual Parameter File Creation](#).

To import node records to the Connect:Direct Secure Plus parameter file from the IBM Connect:Direct network map.

Procedure

1. From the Secure+ Admin Tool: Main Screen, select **File** on the action bar and press Enter.
2. Type **2** to select **Open** and press Enter.
3. Type the IBM Connect:Direct network map file name prefix or partial prefix followed by an asterisk (*) and select **Browse**. and press Enter.

Note: You can also type the entire IBM Connect:Direct network map file name and press **Enter**.

4. Type **S** next to the file name of the network map you want to use and press Enter.
5. When the Quick Start prompt screen is displayed, select **Yes** and press Enter.

After a few seconds, the Secure+ Admin Tool: Main Screen displays nodes populated from the IBM Connect:Direct network map.

6. Update the local and remote node records using the following procedures:

- [Local Node Record Imported from Network Map Configuration](#)
- [Remote Node Record Imported from Network Map Configuration](#)

Manual Parameter File Creation

If you determine that populating the parameter file manually is most efficient for your environment, see [Manual Creation of the SSL or TLS Parameter File](#) for configuring the local and remote node records.

Create the Parameter File Manually for the SSL or TLS Protocol

If you communicate with a large group of trading partners, but only a few trading partners use Connect:Direct Secure Plus, you can manually create and populate the parameter file by creating a single local node record and a remote node record for each trading partner that uses Connect:Direct Secure Plus. This method minimizes the number of remote node records to configure in the parameter file.

For instructions on additional configuration options, see:

- [Adding a Remote Node Record for the External Authentication Server](#)
- [Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server](#)

To validate and test a Connect:Direct Secure Plus connection between two business partners, see [Validating and Testing Connections by Session](#).

Configuration Guidelines

When you use the manual method to populate the parameter file, you should consider disabling all protocols and external authentication and allow override in the local node record. Review the [Node](#)

[Configuration Table](#) to determine the configuration approach that best suits your needs, and use the following guidelines when you configure the local node record manually:

- Disable the Connect:Direct Secure Plus protocols (TLS/ SSL) in the local node record. Then configure each remote node record with the protocol used by that trading partner. To disable all protocols and the External Authentication Server application, you must change Default to Local Node settings in the following panels: SSL/TLS Parameters and EA Parameters. Allow overrides in the Local Node settings.
- Disable external authentication.
- For all environments, you must define required settings in the local node record, including certificate information used with the TLS or SSL protocol. You can also define optional settings in the local node record and use them in all remote node records.
- Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you may be unable to perform secure communications. You must resolve all errors before saving the parameter file.
- To enable secure connections using Connect:Direct Secure Plus, you must complete the procedures in [Adding the Local Node Record to the Parameter File Manually for the SSL or TLS Protocol](#), [Adding a Remote Node Record to the Parameter File Manually for the SSL or TLS Protocol](#), and [Connect:Direct Secure Plus Operation Enablement and Validation](#).
- Before you start the Secure+ Admin Tool to add your local and remote node records, verify that you have allocated the ISPF libraries in your TSO session that are required to save the Connect:Direct Secure Plus parameter file (see [IBM Connect:Direct ISPF Libraries in TSO](#) for details).

Adding the Local Node Record to the Parameter File Manually for the SSL or TLS Protocol

About this task

When you perform this procedure, refer to the [Local Node Security Feature Definition Worksheet](#) that you completed for the local node.

To add the local node record manually:

Procedure

1. Select **Edit** from the Secure+ Admin Tool Main Screen and press **Enter**.
2. On the **Edit** menu, select **1** for **Create/Update Record** and press **Enter** to display the **Secure+ Create/Update** panel:

Secure+ Create/Update Panel

Option ==>

Node Name: MY.LOCAL Type: L (Local or Remote)

Security Options	EA Parameters	SSL/TLS Parameters
---	--	---

Secure+ Protocol: Security Mode (Yes , No , Default to Local)

Enable SSL	N	Enable FIPS	N
Enable TLS 1.0	N	Enable SP800-131a Transition	N
Enable TLS 1.1	N	Enable SP800-131a Strict	N
Enable TLS 1.2	N	Enable NSA Suite B 128 bit	N
		Enable NSA Suite B 192 bit	N

Auth Timeout:	120	Enable Override	N
---------------	-----	-----------------	---

Alias Names:	TCP Information:		
	IPAddr:		
	Port:		

Cancel
OK

3. On the Secure+ Create/Update panel:
 - a) In the **Node Name** field, type the name for the local node.
 - b) Type **L** in the **Type** (Local or Remote) field.
 4. To implement SSL, do one of the following, depending on whether you want to use SSL for all data transfers or on a Process-by-Process basis:
 - Type **Y** beside the **Enable SSL** field to enable the SSL protocol for this local node.
 - Type **N** beside the **Enable SSL** field to disable the SSL protocol.
 5. To implement TLS, do one of the following, depending on whether you want to use TLS for all data transfers or on a Process-by-Process basis:
 - Type **Y** beside the **Enable TLS 1.0** field to enable the TLS protocol for this remote node. Repeat for **TLS 1.1** and **TLS 1.2**.
 - Type **N** beside the **Enable TLS** field to disable the TLS protocol.
- Note:**
- If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.
 - Set the protocols in the Local record as your defaults, allow the Remote record that require different protocols to enable those, and override the Local record.
6. In the Security Mode field, type **Y, N, or D** to enable or disable multiple protocols such as, FIPS, SP800-131a and NSA Suite B.
 7. **Alias Names** field do not apply to local node. This field should be left blank.
 8. **TCP Information** fields (**IP addr** and **Port**) do not apply to local node. This field should be left blank.
 9. Take one of the following actions, depending on whether you want to use the Connect:Direct Secure Plus parameter settings override feature. Override feature for Local nodes allows the remote parameter record to override any setting defined in the Local node record
 - To enable the Connect:Direct Secure Plus parameter settings override feature, type **Y** beside the **Override** field.
 - To disable the Connect:Direct Secure Plus parameter settings override feature, type **N** beside the **Override** field.
 10. Select the **SSL/TLS parameters** panel by typing **SSL** and press **Enter** to display the **Secure+ Create/Update** panel:

Option ==>
Secure+ Create/Update Panel
<Change Pending>

Node Name: MY.LOCAL
Type: L (Local or Remote)

Security Options	EA Parameters	SSL/TLS Parameters
Enable Client Auth	N	(Yes , No , Default to Local)
Enable Data Encrypt	N	
Certificate Label	*	
Cipher Suites	FFFF	
Certificate Pathname	*	
Certificate Common Name		

OK
Cancel

11. To implement Client Authentication and/or Data Encryption, type **Y** for enable or **N** to disable beside the **Client Auth** and/or **Data Encrypt** field.

Note: **D** (Default to Local) is not applicable for the Local node record.

12. Select **Certificate Label** field by placing the cursor on the text and press **Enter**. On the entry panel specify the Certificate Label as defined in the certificate or leave blank to use the default certificate defined in the key database or key ring. Leaving the certificate label blank will generate a warning message up on saving the parameter file. This is meant as a warning that the key store must define a default certificate. Select the **Certificate Label** field and press **Enter**.

Note: *(default to Local) does not apply to the Local node record.

13. Select **Certificate Pathname** field by placing the cursor on the text and press **Enter**.

- a) Enter the complete path of the key database or the key ring name.
- b) This is a scrollable panel, use **F8** key to scroll forward and for the key database enter the case-sensitive password.

Note: A key ring does not have a password and password field should be left blank.

Note: If the SECURE.SSL.PATH.PREFIX initialization parameter has specified a prefix then the complete path name is not required only the name of the key database is sufficient.

Note: **Certificate pathname** is a required for the Local record and cannot be specified on a Remote record.

14. Select **Cipher Suites** by placing the cursor on the text and press Enter.

- a) To select ciphers, order the list in **All Available Cipher-Suites** by placing them 1 through n (maximum of 10).
- b) As ciphers are selected they move to the **Enabled Cipher-Suites** on the right side. This list is the default cipher list.
This is a scrollable panel so use the **F8** key to move forward and **F7** to move back.

0	All Available Cipher-Suites	Enabled Cipher-Suites
==	=====	
=====		More:
+		
1	TLS_RSA_WITH_DES_CBC_SHA	
	TLS_RSA_WITH_DES_CBC_SHA	
2	TLS_RSA_WITH_AES_256_CBC_SHA	
	TLS_RSA_WITH_AES_256_CBC_SHA	
3	TLS_RSA_WITH_AES_256_CBC_SHA256	
	TLS_RSA_WITH_AES_256_CBC_SHA256	
4	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_NULL_MD5	
*DEPRECATE	TLS_RSA_WITH_NULL_SHA	
*DEPRECATE	SA_EXPORT_WITH_RC4_40_MD5	
*DEPRECATE	TLS_RSA_WITH_RC4_128_MD5	
*DEPRECATE	TLS_RSA_WITH_RC4_128_SHA	
*DEPRECATE	EXPORT_WITH_RC2_CBC_40_MD5	
*DEPRECATE		
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	
	TLS_RSA_WITH_AES_128_CBC_SHA	
	TLS_RSA_WITH_NULL_SHA256	
	TLS_RSA_WITH_AES_128_CBC_SHA256	
	TLS_RSA_WITH_AES_128_GCM_SHA256	
	TLS_ECDHE_ECDSA_WITH_NULL_SHA	
	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	
	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	

Note: DEault_T0_LOCAL_NODE does not apply to the Local node record.

Note: Select Ciphers carefully since deprecated ciphers may not be available on all systems. Check with your Security Administrator before selecting these ciphers.

15. Select the **EA parameters** option from the panel selection bar and press **Enter** to display the **EA parameters** panel.

Secure+ Create/Update Panel <Change

Pending>
Option
===>

Node Name: MY.LOCAL Type: L (Local or Remote)

Security Options	EA Parameters	SSL/TLS Parameters
---	--	---

Enable External Auth N (Yes , No , Default to Local)

External Auth Server Def
External Auth Server Address
External Auth Server Port

Cancel OK

-- ---

16. To implement the External Authentication Server application:
 - a) Type **N** in the **External Auth** field to disable External Authentication Server application.
 - b) Type **Y** in the **External Auth** field to enable External Authentication Server application
 - c) External Auth Server Def, External Auth Server Address, and External Auth Server Port are unavailable because they are valid only for the .EASERVER remote node record.
17. Select **OK** and press **Enter** to display the values for the local node record.
18. Using the **Save As** or **Save Active** option displays error and warning messages. Read all warning and error messages. Continue configuring the environment without resolving warning messages, but resolve errors before you save the parameter file.
19. After you configure the local node record, you can save and submit the parameter file using the procedures in [Connect:Direct Secure Plus Operation Enablement and Validation](#), but if you have not added a remote node record, connections are not secure.

Adding a Remote Node Record to the Parameter File Manually for the SSL or TLS Protocol

About this task

Refer to the [Remote Node Security Feature Definition Worksheet](#) that you created for the remote node you are adding when you complete this procedure. The following procedure assumes that this remote node uses the SSL or TLS protocol and client authentication with Connect:Direct Secure Plus unless you

want to override the Connect:Direct Secure Plus parameter settings from the PROCESS statement. For more information, see [Override Settings in IBM Connect:Direct Processes](#).

To add a remote node record manually for the SSL or TLS protocol:

Procedure

1. On the **Option** line type **I** (Insert Node) on the **Secure+ Admin Tool Main Screen** and press **Enter** to add a node. The **Secure+ Create/Update Panel** displays.

```

File  Edit
Help
-----
1              Secure+ Admin Tool: Main Screen              Row 1 of
Option ==>                                         Scroll
CSR

Table Line Commands are:

  U Update node          H View History          D Delete
  I Insert node          V View node             CL Clone

Node Filter :
*

Secure+              External Client
LC Node Name        Type Protocol Override Encryption  Auth
Auth
-----
I MY_LOCAL          L  TLSV12          Y          Y          N
Y
***** BOTTOM OF DATA *****

```

2. On the **Secure+ Create/Update Panel**:
 - a) In the **Node Name** field, type the name for the remote node that corresponds to its name in the network map.
 - b) Type **R** in the **Type** (Local or Remote) field.

```

                                Secure+ Create/Update Panel
Option ==>

Node Name:  MY.REMOTE              Type:  R              (Local or Remote)
-----
| Security Options | EA Parameters | SSL/TLS Parameters |
| ---            | --          | ---                |
-----
Secure+ Protocol:
Enable SSL          N          Security Mode (Yes , No , Default to Local)
Enable TLS 1.0      N          Enable FIPS          N
Enable TLS 1.1      N          Enable SP800-131a Transition  N
Enable TLS 1.2      N          Enable SP800-131a Strict    N
                          Enable NSA Suite B 128 bit  N
                          Enable NSA Suite B 192 bit  N

Auth Timeout:      120          Enable Override          N

Alias Names:
TCP Information:
IPaddr:
Port:

                                OK
Cancel

```


3. To implement SSL, do one of the following, depending on whether you want to use SSL for all data transfers or on a Process-by-Process basis:
 - Type **Y** beside the **Enable SSL** field to enable the SSL protocol for this remote node.
 - Type **N** beside the **Enable SSL** field to disable the SSL protocol
 4. To implement TLS, do one of the following, depending on whether you want to use TLS for all data transfers or on a Process-by-Process basis:
 - Type **Y** beside the **Enable TLS 1.0** field to enable the TLS protocol for this remote node. Repeat for **TLS 1.1** and **TLS 1.2**.
 - Type **N** beside the **Enable TLS** field to disable the TLS protocol.
- Note:** If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.
- Note:** Set the required protocols to override the default protocols in the Local record.
5. In the Security Mode field, type **Y, N, or D** to enable or disable multiple protocols such as, FIPS, SP800-131a and NSA Suite B.
 6. **Alias Names** field do not apply to remote node. This field should be left blank.
 7. **TCP Information** fields (**IP addr** and **Port**) do not apply to remote node. This field should be left blank.
 8. Depending on whether you want to use the Connect:Direct Secure Plus parameter settings override feature, type **Y** to enable or **N** to disable beside the Override field. Enabling **Override** for the Remote record allows not only the Process to override the security settings but also allows the SNODE to override the security setting. Use caution when enabling this option on the Remote record to override any setting defined in the Local node record
 - .
 9. Select the **SSL/TLS parameters** panel by typing **SSL** and press **Enter** to display the **Secure+ Create/Update** panel:

Option ==>		Secure+ Create/Update Panel	<Change Pending>
Node Name: MY.REMOTE		Type: R	(Local or Remote)
Security Options	EA Parameters	SSL/TLS Parameters	

Enable Client Auth	D	(Yes , No , Default to Local)	
Enable Data Encrypt	D		
Certificate Label	*		
Cipher Suites	FFFF		
Certificate Pathname	*		
Certificate Common Name			

		OK	Cancel

10. To implement Client Authentication and/or Data Encryption, type **Y** for enable or **N** to disable, or **D** for default to local record, beside the **Client Auth** and/or **Data Encrypt** field.
11. Select **Certificate Label** field by placing the cursor on the text and press Enter. On the entry panel specify the **Certificate Label** as defined in the certificate or leave blank to use the default certificate defined in the key database or key ring. Leaving the certificate label blank will generate a warning message up on saving the parameter file. This is meant as a warning that the key store must define a default certificate. Select the Certificate Label field and press **Enter**.

Note: The **Certificate label** field is automatically set to '*' (Default to Local) in the Remote Node record. You are not allowed to update this field for a remote node.

12. **Certificate Pathname** does not apply to a remote node. This field should be left blank

13. Select **Cipher Suites** by placing the cursor on the text and press **Enter**:

- To select ciphers, order the list in **All Available Cipher-Suites** by placing them 1 through n (maximum of 10).
- As ciphers are selected they move to the **Enabled Cipher-Suites** on the right side. This list is the default cipher list.

This is a scrollable panel so use the **F8** key to move forward and **F7** to move back.

0	All Available Cipher-Suites	Enabled Cipher-Suites
==	=====	
=====		
		More:
+		
1	TLS_RSA_WITH_DES_CBC_SHA	
	TLS_RSA_WITH_DES_CBC_SHA	
2	TLS_RSA_WITH_AES_256_CBC_SHA	
	TLS_RSA_WITH_AES_256_CBC_SHA	
3	TLS_RSA_WITH_AES_256_CBC_SHA256	
	TLS_RSA_WITH_AES_256_CBC_SHA256	
4	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_NULL_MD5	
*DEPRECATE		
	TLS_RSA_WITH_NULL_SHA	
*DEPRECATE		
	SA_EXPORT_WITH_RC4_40_MD5	
*DEPRECATE		
	TLS_RSA_WITH_RC4_128_MD5	
*DEPRECATE		
	TLS_RSA_WITH_RC4_128_SHA	
*DEPRECATE		
	EXPORT_WITH_RC2_CBC_40_MD5	
*DEPRECATE		
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	
	TLS_RSA_WITH_AES_128_CBC_SHA	
	TLS_RSA_WITH_NULL_SHA256	
	TLS_RSA_WITH_AES_128_CBC_SHA256	
	TLS_RSA_WITH_AES_128_GCM_SHA256	
	TLS_ECDHE_ECDSA_WITH_NULL_SHA	
	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	
	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	

Note: DEault_TO_LOCAL_NODE can be used to use the Cipher list defined on Local node record.

Note: Select Ciphers carefully since deprecated ciphers may not be available on all systems. Check with your Security Administrator before selecting these ciphers.

14. Select the **EA parameters** option from the panel selection bar and press **Enter** to display the **EA parameters** panel.

Secure+ Create/Update Panel <Change Pending>

Option
====>

Node Name: MY.REMOTE Type: R (Local or Remote)

Security Options	EA Parameters	SSL/TLS Parameters
---	--	---

Enable External Auth N (Yes , No , Default to Local)

External Auth Server Def

External Auth Server Address

External Auth Server Port

Cancel OK

-- ---

15. To implement the External Authentication Server application:
 - a) Type **N** in the **External Auth** field to disable External Authentication Server application.
 - b) Type **Y** in the **External Auth** field to enable External Authentication Server application
 - c) **External Auth Server Def**, **External Auth Server Address**, and **External Auth Server Port** are unavailable because they are valid only for the .EASERVER remote node record.
16. Select **OK** and press **Enter** to display the values for the local node record.
17. Using the **Save As** or **Save Active** option displays error and warning messages. Read all warning and error messages. Continue configuring the environment without resolving warning messages, but resolve errors before you save the parameter file.
18. After you configure the remote node record, you can save and submit the parameter file using the procedures in [Connect:Direct Secure Plus Operation Enablement and Validation](#), but if you have not added a remote node record, connections are not secure.

Additional Configuration Options

You can configure special-purpose remote node records to perform the following functions:

- Validate certificates using External Authentication Server
- Allow for secure TCP API connections (TLS and SSL)
- Secure passwords at rest within the IBM Connect:Direct TCQ and AUTH files

With the SSL and TLS protocols, you can validate certificates using the External Authentication Server application. To use External Authentication Server, configure your application to connect to the host name and port where the External Authentication Server application (.EASERVER) resides. Specify a certificate

validation definition. For configuration instructions, see [“Adding a Remote Node Record for the External Authentication Server”](#) on page 786.

Use only secure TCP API connections to connect to a Connect:Direct for z/OS server. For configuration instructions, see [Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server](#).

In IBM Connect:Direct, passwords can be used in IBM Connect:Direct when Processes are submitted, during API signons, and when the AUTH file is maintained. You can use Strong Password Encryption SPE) to secure passwords at rest within the IBM Connect:Direct TCQ and AUTH files. See [Implementing Strong Password Encryption](#).

Adding a Remote Node Record for the External Authentication Server

About this task

To verify certificates using External Authentication Server, create a remote node record for the External Authentication (EA) Server in the Connect:Direct Secure Plus parameter file. Before you begin, complete the [.EASERVER Node Security Feature Definition Worksheet](#).

To add a remote node record for the External Authentication Server:

Procedure

1. Select **Edit** from the **Secure+ Admin Tool Main Screen** and press **Enter**.
2. On the **Edit** menu, type **1** to select **Create/Update Record** and press **Enter**.
3. On the Secure Plus: Create/Update Panel:
 - a) Type **.EASERVER** in the **Node Name** field.
 - b) Type **R** beside the **Type** field.
 - c) Select **EA Parameters** and press **Enter**.
 - d) Type ***** beside the **Override** field because it is not relevant to External Authentication.
 - e) Type **N** beside the **External Auth** field.
4. On the **EA Parameters** screen:
 - a) Type information from the worksheet for the .EASERVER record in the following fields:

Field	Description
External Auth Server Def	Name of the certificate validation definition configured on the External Authentication Server that defines how to validate certificates. This field is case sensitive.
External Auth Server Address	IP address of server for External Authentication Server
External Auth Server Port	Port number to connect to the External Authentication Server.

Note: After you create the .EASERVER remote node record, the **External Auth Server Def**, **External Auth Server Address**, and **External Auth Server Port** fields are populated in the EA Parameters panel of all Connect:Direct Secure Plus parameter file records, but the only field that can be modified from a record other than the .EASERVER record is the **Enable External Auth** field.

- b) Select **SSL/TLS Parameters** in the panel selection bar and press **Enter**.
5. To enable client authentication, Type **Y** beside the **Enable Client Auth** field.
 6. To specify the certificate label:
 - a) Select the **Certificate Label** field and press **Enter**.

Note: Leave this field blank to use the default certificate defined in the keystore.

- b) Press **F8** to move to the editable portion of the label field.
 - c) This field is case-sensitive, therefore, type the label of the certificate exactly as you defined it when you generated it using one of the security applications described in [Configuration Worksheets](#), or type an asterisk (*) to specify the same label as the local node, and press **Enter**. Leave this field blank to use the default certificate defined in the key store.
- Note:** The Certificate Pathname field is automatically set to '*' (Default to Local) in the Remote Node record. You are not allowed to update this field for a remote node.
7. To enable ciphers:

Note: If System SSL is in FIPS mode, only certain ciphers are valid. See the *IBM Connect:Direct for z/OS Release Notes* for a list of valid FIPS-mode ciphers.

 - a) Select **Security Options** and press **Enter**.
 - b) Type **Y** by the cipher you want to enable.
 - c) Continue typing **Y** or **N** next to the ciphers you want to enable or disable.
 - d) Press **OK** when you have enabled all necessary ciphers.
 8. Select **OK** and press **Enter** to save and close this remote node record.
 9. Read all warning and error messages. You can configure the environment without resolving warning messages, but you must resolve errors before you save the parameter file.
 10. Press **Cancel** to display current settings for the EA node.
 11. Save the parameter file using the procedure in [Connect:Direct Secure Plus Operation Enablement and Validation](#).

Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server

About this task

IBM Connect:Direct servers that use Connect:Direct Secure Plus allow you to allow secure TCP API connections. Secure API applications can include Control Center and Sterling Connect:Direct Browser User Interface.

The Connect:Direct CICS Option, *IBM Connect:Direct for z/OS* batch interface, ISPF IUI, Console interface and Interconnect Option (ICO) do not support a secure connection. If a .CLIENT record is enabled, ensure that SNA protocol is available and configured for these User Interfaces.

Note: To run Connect:Direct Secure Plus using a nonsecure API connection, set the S +.CMD.ENFORCE.SECURE.CONNECTION parameter to NO. See *Global Initialization Parameters*, in the *IBM Connect:Direct for z/OS Administration Guide*. In addition, specify OVERRIDE=YES in step 4 in the following procedure.

To enable secure TCP API connections, define a remote node record called .CLIENT and disable override. Additionally, identify the protocol to use for secure API connections. Defining a remote node called .CLIENT and disabling override prevents nonsecure connections to the IBM Connect:Direct server without disabling override settings in the local node record.

An API configuration follows the same rules as other remote node connections with the following exceptions:

- API connections use either the SSL or the TLS security protocol.
- The IBM Connect:Direct server supports TCP and defines a TCP API port for these connections. Refer to *IBM Connect:Direct for z/OS Administration Guide* for instructions on setting up TCP API support on the server.
- Settings in the .CLIENT node definition automatically override the local node.

To configure a .CLIENT remote node record when Connect:Direct Secure Plus is enabled:

Procedure

1. From the **Secure+ Admin Tool Main Screen**, select **Edit** and press **Enter** to display the **Edit** menu.
2. On the **Edit** menu, type **1** to select **Create/Update Record** and press **Enter**.
3. On the **Secure+ Create/Update** panel:
 - a) Type **.CLIENT** in the **Node Name** field.

Note: You must name this node **.CLIENT** in order for IBM Connect:Direct to read this node and allow secure TCP API connections.
 - b) Type **R** next to the **Type** field.
 - c) Select **EA Parameters** and press **Enter**.
4. In the **EA Parameters** panel:
 - a) Type **N** beside the **Enable External Auth** field to disable it. The remaining EA parameters are unavailable because they are valid only for the **.EASERVER** remote node record.
 - b) Select **SSL/TLS Parameters** and press **Enter**.
5. Take one of the following actions, depending on whether you want to use the Connect:Direct Secure Plus parameter settings override feature:

Note: If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.

 - a) Type **N** beside the **Enable Client Auth** field to disable it.
 - b) Click **Security Options**.
6. The remaining fields are not valid for the **.CLIENT** record.
7. Click **OK** and press **Enter** to save and close the **.CLIENT** node record.
8. Save the parameter file using the procedure in [Connect:Direct Secure Plus Operation Enablement and Validation](#).
9. Ensure that the ISPF IUI and batch interface connections define SNA as the connection protocol.

Note: If the **.CLIENT** node record disables the Override function, ISPF IUI and must use the SNA protocol.

Implementing Strong Password Encryption

To implement the Strong Password Encryption (SPE), you add an SPE record to the Connect:Direct Secure Plus parameter file in the same way you would any remote node record. After you go through the following procedure and restart Connect:Direct Secure Plus for z/OS, the SPE feature will be in effect.

About this task

To add an SPE record to the Connect:Direct Secure Plus parameter file and enable the SPE feature follow the procedure given below:

Note: After SPE is implemented, follow the procedure described in [“Disabling Strong Password Encryption”](#) on page 789 to disable or remove encrypted passwords.

Procedure

1. Select **Edit** from the Secure+ Admin Tool Main Screen and press Enter.
2. On the **Edit** menu, select **1** to display the Secure+ Create/Update Panel and press **Enter**.
3. On the **Secure+ Create/Update** panel type **.password** in the **Node Name** field and press **ENTER**. This changes the screen to enable only fields that are appropriate for the **.password** record.
4. On the **SPE Parameters** panel, type **Y** to the **Enable SPE** field and
and
N to the disable SPE.

Press **Enter** to enable SPE and finish creating the SPE record by clicking **OK**.

5. Save the parameter file using the procedure in [Connect:Direct Secure Plus Operation Enablement and Validation](#).
6. Restart Connect:Direct Secure Plus for z/OS.
7. To verify that Connect:Direct Secure Plus for z/OS initialization is complete along with the SPE feature, after you restart Connect:Direct Secure Plus for z/OS, review the task output for the following messages interspersed with the other initialization messages:

```
SITA460I Strong Password Encryption Initiated; CONNECT.CD.AUTH
SITA462I Strong Password Encryption Completed; CONNECT.CD.AUTH
SITA460I Strong Password Encryption Initiated; CONNECT.CD.TCQ
SITA462I Strong Password Encryption Completed; CONNECT.CD.TCQ
```

Note: These messages display even if no .PASSWORD record exists and no encryption is possible. If you return to the Secure+ Create/Update Panel - SPE Parameters screen where you enabled SPE, you should see (SPE currently in use) displayed to confirm that SPE has indeed been implemented.

Disabling Strong Password Encryption

About this task

If the Strong Password Encryption feature is backed out inappropriately by deleting the .PASSWORD record while encrypted passwords exist in the TCQ and AUTH files in the SPE format, you will see one of the messages listed in the following section, [SPE Problem Troubleshooting](#). If the SPE feature is backed out inappropriately, the only option is to COLD start the TCQ and rebuild the entire AUTH file.

Follow the procedure in this section, restart Connect:Direct Secure Plus for z/OS, and then enable the SPE feature again. Perform this procedure as documented, do not vary from the procedure.

To disable the SPE feature:

Procedure

1. Start the Secure+ Admin Tool to display the **Secure+ Admin Tool: Main Screen**, which displays the nodes populated from the Connect:Direct Secure Plus for z/OS network map along with other records in the Connect:Direct Secure Plus parameter file.

```

File Edit Help
-----
CD.ZOS.NODE          Secure+ Admin Tool: Main Screen          Row 1 of 7
Option ==> ----- Scroll CSR

                        Table Line Commands are:

U Update node        H View History        D Delete node
I Insert node        V View node

Node Filter :
* -----

LC Node Name      Type  Secure+ Protocol Override Encryption  External Client
-----
-- .CLIENT        R    *          N          *          *          *
-- .EASERVER       R    TLSV10      N          *          N          *
-- .PASSWORD       R    Disabled    *          *          *          *
-- CD.UNIX.NODE    R    TLSV10      *          *          *          *
-- CD.UNIX.NODE2   R    TLSV12      *          *          *          *
-- CD.ZOS.NODE     L    Disabled    Y          N          N          N
-- CD.ZOS.NODE2    R    *          *          *          *          *
***** BOTTOM OF DATA *****

```

2. Type **U** next to the **.PASSWORD** record and press **Enter** to display the **Secure+ Create/Update Panel - SPE Parameters** screen.
3. On the **SPE Parameters** panel, type **N** next to the **Enable SPE** field and press **Enter**.

4. Save the parameter file using the procedure in [Connect:Direct Secure Plus Operation Enablement and Validation](#).
5. Restart Connect:Direct Secure Plus for z/OS.

SPE Problem Troubleshooting

When SPE is implemented, each time the IBM Connect:Direct server is started, a new encryption key is generated, the passwords decrypted with the old key, passwords encrypted with the new key then both keys saved. If the Strong Password Encryption key stored in the .PASSWORD record is out of sync with the SPE key used to encrypt the passwords, errors can occur and you must reset all SPE passwords and reimplement the SPE feature.

The .PASSWORD record can get out of sync if one of the following occurs:

- You restore the .PASSWORD record from a backup of the Connect:Direct Secure Plus parameter file—The .PASSWORD record is updated and a new encryption key generated each time the Connect:Direct Secure Plus for z/OS server is restarted, so the backup will probably not contain the current parameters.
- The .PASSWORD record is deleted outside of IBM Connect:Direct and Connect:Direct Secure Plus—The .PASSWORD record is recreated as needed, so the SPE key used to encrypt the passwords no longer exists.
- The .PASSWORD record is corrupt—The SPE encryption key used to encrypt the passwords is not accessible.
- After implementing SPE, IBM Connect:Direct server is restarted with Secure Plus disabled.

The following tables identify errors you may experience when using the SPE feature, along with solutions to fix each issue.

Condition: Because of SPE errors, Connect:Direct Secure Plus for z/OS either initializes with a SITA461I message or does not initialize at all with a SITA463E message.

Error	Cause	Action
SITA461I SITA463E	<p>SPE-formatted passwords exist In the TCQ and/or AUTH files, but Connect:Direct Secure Plus has not been enabled.</p> <ul style="list-style-type: none"> • SPE-formatted passwords exist In the TCQ and/or AUTH files, but there is no .PASSWORD record in the Connect:Direct Secure Plus parameter file. • SPE-formatted passwords exist in the TCQ and/or AUTH files, but the .PASSWORD record in the Connect:Direct Secure Plus parameter file has OLD encryption keys. This can only occur if an old Connect:Direct Secure Plus parmfile is restored with a backup that contains old keys. 	<p>IBM Connect:Direct for z/OS has not been set up to run with Connect:Direct Secure Plus for z/OS. Add the <code>SECURE.DSN=filename</code> parameter to the initialization parameters, where <i>filename</i> is the name of the Connect:Direct Secure Plus parameter file. Restart Connect:Direct Secure Plus for z/OS. To see more detailed information about individual errors related to the general failure, see the ESTAE trace output.</p> <p>Reset all passwords in the TCQ and AUTH files by performing these actions:</p> <ul style="list-style-type: none"> • Select the AUTH file record in the AUTH file. Provide a new password and blank out all unusable data. • In the TCQ file, delete all Processes and resubmit. <p>To see more detailed information about individual errors related to the general failure, see the ESTAE trace output.</p>

Condition: You encounter errors while trying to maintain the AUTH file.

Error	Cause	Action
SAFB023W SAFF016W SAFC016W SAFE016W	While inserting and updating users through the IUI (INSERT/UPDATE/SELECT/DELETE USER RECORD screen), Connect:Direct Secure Plus for z/OS could not read or record passwords. The .PASSWORD record does not contain the correct encryption key pair. The Connect:Direct Secure Plus parameter file may have been restored with an old copy of the .PASSWORD record.	<ol style="list-style-type: none">1. Disable the SPE feature.2. Restart Connect:Direct Secure Plus for z/OS.3. Enable the SPE feature again.4. Restart Connect:Direct Secure Plus for z/OS. <p>To see more detailed information about individual errors related to the general failure, see the ESTAE trace output.</p>

Configure the Local Node Record Imported from the Network Map

The following procedures assume that you populated the parameter file by importing the network map. The Quick Start method creates a remote node record in the parameter file for each remote node record in the network map and a local node record. Using the Quick Start method to populate the parameter file is most efficient if you have a large number of trading partners that use the same protocol. You can enable that protocol in the local node record and because the remote node records are set automatically to Default to Local Node, they inherit the settings of the local node.

Note: The Quick Start method is a one time procedure and if a large number of partner nodes are defined in the NETMAP more storage may be required to save the parameter file. Steps should be taken to ensure the TSO/ISPF region has adequate size.

Depending on how you configure the local node record, you may or may not need to modify the remote node records. You must disable the Connect:Direct Secure Plus protocols in the records for all remote nodes that do not use Connect:Direct Secure Plus, and update all remote node records that use a protocol that is different from the protocol defined in the local node record.

Use the following procedures to configure the local node record imported from the network map for the SSL and TLS protocols:

- [Configuring the Local Node Record for the SSL or TLS Protocol](#)

Configuration Guidelines

Observe the following guidelines when you configure node records imported from the network map:

- Connect:Direct Secure Plus protocols are disabled initially for all records created from the network map when you use Quick Start to populate the parameter file.
- Before you start the Secure+ Admin Tool to configure the local node record, you may want to save and submit the parameter file to verify that you can generate the SAVE AS JCL. If you are unable to generate the JCL for the SAVE AS job, verify that you have allocated the ISPF libraries in your TSO session that are required to save the Connect:Direct Secure Plus parameter file (see [Saving and Submitting the Connect:Direct Secure Plus Parameter File](#)).
- To enable secure connections using Connect:Direct Secure Plus for z/OS, you must complete the following procedures:
 - [Configuring the Local Node Record for the SSL or TLS Protocol](#)
 - [Relevant procedures in Remote Node Record Imported from Network Map Configuration](#)
 - [Procedures in Connect:Direct Secure Plus Operation Enablement and Validation](#)

- Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you may be unable to perform secure communications. You must resolve all errors before saving the parameter file.

Configuring the Local Node Record for the SSL or TLS Protocol

About this task

All Connect:Direct Secure Plus protocols are disabled when you import the network map. This procedure updates the local node record for the SSL (or TLS) protocol and enables the **Override** parameter. Remember that all options set for the local node are inherited by all remote node records.

To update the local node record for the SSL (or TLS) protocol:

Procedure

1. From the Connect:Direct Secure Plus Admin Main screen, type **U** next to the local node record and press **Enter** to display the **Secure+ Create/Update Panel** and the current values for the selected node.

Note: When you import the network map, the system enables **Override** in the local node record automatically, as shown in the following illustration.

```

File  Edit  Help
-----
CD.ZOS.NODE          Secure+ Admin Tool: Main Screen          Row 1 of 7
Option ==>          -----          Scroll CSR

                          Table Line Commands are:

    U Update node          H View History          D Delete node
    I Insert node          V View node

Node Filter :
*-----

LC Node Name          Type  Secure+ Protocol Override Encryption External Client
-----
-- .CLIENT            R    *          N          *          *          *
-- .EASERVER           R    TLSV10       N          *          N          *
-- .PASSWORD           R    Disabled    *          *          *          *
-- CD.UNIX.NODE        R    TLSV10       *          *          *          *
-- CD.UNIX.NODE2       R    TLSV12       *          *          *          *
-- CD.ZOS.NODE         L    Disabled    Y          N          N          N
-- CD.ZOS.NODE2       R    *          *          *          *          *
***** BOTTOM OF DATA *****

```

2. Select **SSL/TLS Parameters** in the panel selection line and press **Enter** to display the **SSL/TLS Parameters** panel.
3. To select the protocol you want to enable, type **Y** beside the Secure+ Protocol you want to enable, or **N** beside the protocol you want to disable.

Note: If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.

Note: You can enable multiple protocols support and the highest supported protocol will be used.

4. Type **Y** in the **Enable Override** field.
5. On the EA Parameters screen, type **N** next to the Enable External Auth field.
6. On the Security Options screen, do one of the following depending on the Encrypt option you want to implement:
 - a) To encrypt both the control block information and the files being transferred, type **Y** beside the **Encrypt** field.

- b) To encrypt only the control block information used to establish the session, type **N** beside the **Encrypt** field.
 - c) To default to the local node record, type **D** beside the **Encrypt** field.
7. If necessary, update the certificate label:
- a) Select the **Certificate Label** field and press **Enter**.

Note: If System SSL is in FIPS Mode, the Certificate Label has FIPS requirements. See [“Planning for System SSL in FIPS Mode”](#) on page 751.

Note: To use the Default Certificate defined in the certificate file, leave the Certificate Label field blank.
 - b) Press **F8** to move to the editable portion of the panel containing the label field.
 - c) This field is case sensitive; type the certificate label exactly as you defined it when you generated it and press **Enter**. To use the default certificate defined in key store, leave the **Certificate Label** field blank.
8. If necessary, update the location where the certificate information is stored:
- a) Select the **Certificate Pathname** field and press **Enter** to display the **Certificate Pathname** panel.

Note: If System SSL is in FIPS Mode, the Certificate Pathname has FIPS requirements. See [“Planning for System SSL in FIPS Mode”](#) on page 751.
 - b) Press **F8** to scroll to the **Certificate Path Name** field.
 - c) Type the UNIX path name of the key database (.kdb) or the security system key ring name that contains all the certificates referred to in the parameter file.

Note: This value is case sensitive. Ensure that you type it exactly as it appears in the certificate file. Refer to the information you recorded in [Local Node Security Feature Definition Worksheet](#).
 - d) If you are using a key database:
 - i) Press **F8** to scroll to the password field.
 - ii) Type the password used when the key database was created and press **Enter**.

Note: This value is case sensitive. Ensure that you type it exactly as it appears in the certificate file. Refer to the information you recorded in [Local Node Security Feature Definition Worksheet](#).

Note: If you are using a key ring, leave the password field blank.
9. Select **Cipher Suites** by placing the cursor on the text and press **Enter**.
- a) To select ciphers, order the list in **All Available Cipher-Suites** by placing them 1 through n (maximum of 10).
 - b) As ciphers are selected they move to the **Enabled Cipher-Suites** on the right side. This list is the default cipher list.
- This is a scrollable panel so use the **F8** key to move forward and **F7** to move back.

```

0  All Available Cipher-Suites          Enabled Cipher-
Suites
==  =====
=====
More:

+
1  TLS_RSA_WITH_DES_CBC_SHA
   TLS_RSA_WITH_DES_CBC_SHA
2  TLS_RSA_WITH_AES_256_CBC_SHA
   TLS_RSA_WITH_AES_256_CBC_SHA
3  TLS_RSA_WITH_AES_256_CBC_SHA256
   TLS_RSA_WITH_AES_256_CBC_SHA256
4  TLS_RSA_WITH_AES_256_GCM_SHA384
   TLS_RSA_WITH_AES_256_GCM_SHA384
   TLS_RSA_WITH_NULL_MD5
*DEPRECATE
   TLS_RSA_WITH_NULL_SHA
*DEPRECATE
   SA_EXPORT_WITH_RC4_40_MD5
*DEPRECATE
   TLS_RSA_WITH_RC4_128_MD5
*DEPRECATE
   TLS_RSA_WITH_RC4_128_SHA
*DEPRECATE
   EXPORT_WITH_RC2_CBC_40_MD5
*DEPRECATE

TLS_RSA_WITH_3DES_EDE_CBC_SHA

TLS_RSA_WITH_AES_128_CBC_SHA

TLS_RSA_WITH_NULL_SHA256

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_NULL_SHA

TLS_ECDHE_ECDSA_WITH_RC4_128_SHA

TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA

```

Note: DEault_T0_LOCAL_NODE does not apply to the Local node record.

Note: Select Ciphers carefully since deprecated ciphers may not be available on all systems. Check with your Security Administrator before selecting these ciphers.

Note: If you do not know what cipher suites are available, run a trace on the Connect:Direct system. Setting debug=8C0000AE in the initialization parameter file dynamically allocates DD R00000001. Available cipher suites are listed in the trace DD. Turn global tracing off before you continue.

10. Select **EA Parameters** and press **Enter**.
11. Verify that External Authentication (**External Auth**) is disabled (set to **2**). The remaining external authentication fields are unavailable because they are valid only for the .EASERVER remote node record.
12. Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you may be unable to perform secure communications. You must resolve all errors before you can save the parameter file.
13. After you configure the local node record, you can save and submit the parameter file using the procedures in [Connect:Direct Secure Plus Operation Enablement and Validation](#), but if you have not added a remote node record, connections are not secure.

Configure Remote Node Records Imported from the Network Map

The following procedures assume that you populate the parameter file by importing the network map. The Quick Start method creates a remote node record in the parameter file for each remote node record in the network map and a local node record. Using the Quick Start method to populate the parameter file is most efficient if you have a large number of trading partners that use the same protocol. You can enable that protocol in the local node record and because the remote node records are set automatically to Default to Local Node, they inherit the settings of the local node.

Note: The following procedures require you to edit entries populated into the parameter file to ensure appropriate environment setting.

Depending on how you configured the local node record, you may or may not need to modify the remote node records. You must disable the Connect:Direct Secure Plus protocols in the records for all remote nodes that do not use Connect:Direct Secure Plus, and update all remote node records that use a protocol that is different from the protocol defined in the local node record.

Use the following procedures to modify remote node records imported from the network map for the SSL and TLS protocols, and to disable all protocols for a remote node:

- [Configuring a Remote Node Record for the SSL or TLS Protocol](#)
- [Disabling Connect:Direct Secure Plus in a Remote Node Record](#)

For instructions on configuring these special-purpose remote node records for the TLS and SSL protocol, see the following procedures:

- [Establishing Secure TCP API Connections to a Connect:Direct Secure Plus-Enabled Server](#)
- [Adding a Remote Node Record for the External Authentication Server](#)

Configuration Guidelines

Observe the following guidelines when you configure node records imported from the network map:

- Connect:Direct Secure Plus protocols are disabled for all records created from the network map when you use Quick Start to populate the parameter file.
- For all environments, define required settings in the local node record. If desired, you can define optional settings in the local node record and use them in all remote node records.
- To enable secure connections using Connect:Direct Secure Plus for z/OS, you must complete the relevant procedure for configuring the local node record in [Local Node Record Imported from Network Map Configuration](#), the relevant procedures in this chapter, and the procedures in [Connect:Direct Secure Plus Operation Enablement and Validation](#).
- Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you may be unable to perform secure communications. You must resolve all errors before saving the parameter file.

Configuring a Remote Node Record for the SSL or TLS Protocol

About this task

After you configure the local node, you can configure remote node records. When you import the network map file, you create a remote node record in the parameter file for each remote node record in the network map. Depending on how you configured the local node record, you may or may not need to update the remote node records.

- If you disabled the Connect:Direct Secure Plus protocols in the local node record, Connect:Direct Secure Plus is disabled for all remote node records. You must update all remote node records that use Connect:Direct Secure Plus to identify which protocol is used by the trading partner.
- If you enabled a protocol in the local node record, that protocol is enabled in all remote node records. You must disable the Connect:Direct Secure Plus protocols in the records for all remote nodes that do

not use Connect:Direct Secure Plus, and update all remote node records that use a protocol that is different from the protocol defined in the local node record.

Note: To override security functions for a particular session, you can use the SECURE parameter in the PROCESS statement. For more information, see [Override Settings in IBM Connect:Direct Processes](#). Note that the more flexible you make the environment by allowing override, the less secure that environment becomes.

The following procedure assumes that you enabled the SSL (or TLS) protocol in the local node record, this remote node uses the SSL (or TLS) protocol, and that you need to modify some SSL (or TLS) parameters for this remote node record.

To update a remote node record for the SSL (or TLS) protocol:

Procedure

1. Type **U** next to the remote node record to update and press **Enter** to display the current values for the selected node in the Secure+ Create/Update Panel - **SSL/TLS Parameters** panel.

Note: An asterisk in a field on the Secure+ Admin Main Screen indicates the value **Default to Local Node**. If the TLS protocol is enabled in the Local Node record, **Y** appears in the third position instead of the second position in the Secure 123C column below.

```

File  Edit  Help
-----
CD.ZOS.NODE      Secure+ Admin Tool: Main Screen      Row 1 of 7
Option ==>      -----      Scroll CSR

                        Table Line Commands are:

U Update node      H View History      D Delete node
I Insert node      V View node

Node Filter : *-----

LC Node Name      Type      Secure+      Protocol      Override      Encryption      External      Client
-----      ---      ---      ---      ---      ---      ---      ---
-- .CLIENT      R      *      N      *      *      *
-- .EASERVER      R      TLSV10      N      *      N      *
-- .PASSWORD      R      Disabled      *      *      *      *
-- CD.UNIX.NODE      R      TLSV10      *      *      *      *
-- CD.UNIX.NODE2      R      TLSV12      *      *      *      *
-- CD.ZOS.NODE      L      Disabled      Y      N      N      N
-- CD.ZOS.NODE2      R      *      *      *      *      *
***** BOTTOM OF DATA *****

```

2. Select **EA Parameters** and press **Enter**.

3. In the **EA Parameters** panel:

- a) Specify a value for the External Authentication parameter, if required, using the following table as a guide:

Field	Description	Valid Values
External Auth	Allows validating certificates for secure sessions using External Authentication Server.	Y to enable N to disable D to Default to local node

- b) Select **SSL/TLS Parameters** in the panel selection line and press **Enter**.

4. Take one of the following actions depending on the protocol you are implementing:

- If you defined default SSL settings in the local node record that this remote node record uses, verify that the **Enable TLS** field is disabled (set to **N**) or set to **Default to Local Node (D)**. If you do not need to change any other settings, continue with [step 10](#).

- If you defined default TLS settings in the local node record that this remote node record uses, verify that the **Enable SSL** field is disabled (set to **N**) or set to **Default to Local Node (D)**. If you do not need to change any other settings, continue with [step 10](#).

Note: If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.

- To modify SSL (or TLS) protocol settings in a remote node record, continue with [step 7](#).
5. Take one of the following actions, depending on what information you want to encrypt:
 - Type **Y** beside the **Encrypt** field to encrypt all information sent during the handshake to set up communication sessions and the actual files being transferred.
 - Type **N** beside the **Encrypt** field to encrypt only the control block information sent during the handshake to set up communication sessions and not the actual files being transferred.
 6. Take one of the following actions, depending on whether you want to use the Connect:Direct Secure Plus parameter settings override feature.
 - To enable the Connect:Direct Secure Plus parameter settings override feature in the PROCESS or COPY statement, type **Y** beside the **Override** field. For more information, see [Override Settings in IBM Connect:Direct Processes](#).

Note: Allowing override in the remote record not only allows the Process to override the security settings but also allows the SNODE to override. Use caution with this option, the more you allow override the less security the environment becomes.

 - To disable the Connect:Direct Secure Plus parameter settings override feature, type **N** beside the **Override** field.
 7. Select **Cipher Suites** by placing the cursor on the text and press **Enter**:
 - a) To select ciphers, order the list in **All Available Cipher-Suites** by placing them 1 through n (maximum of 10).
 - b) As ciphers are selected they move to the **Enabled Cipher-Suites** on the right side. This list is the default cipher list.

This is a scrollable panel so use the **F8** key to move forward and **F7** to move back.

0	All Available Cipher-Suites	Enabled Cipher-Suites
==	=====	
=====		
		More:
+		
1	TLS_RSA_WITH_DES_CBC_SHA	
	TLS_RSA_WITH_DES_CBC_SHA	
2	TLS_RSA_WITH_AES_256_CBC_SHA	
	TLS_RSA_WITH_AES_256_CBC_SHA	
3	TLS_RSA_WITH_AES_256_CBC_SHA256	
	TLS_RSA_WITH_AES_256_CBC_SHA256	
4	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_AES_256_GCM_SHA384	
	TLS_RSA_WITH_NULL_MD5	
*DEPRECATE		
	TLS_RSA_WITH_NULL_SHA	
*DEPRECATE		
	SA_EXPORT_WITH_RC4_40_MD5	
*DEPRECATE		
	TLS_RSA_WITH_RC4_128_MD5	
*DEPRECATE		
	TLS_RSA_WITH_RC4_128_SHA	
*DEPRECATE		
	EXPORT_WITH_RC2_CBC_40_MD5	
*DEPRECATE		
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	
	TLS_RSA_WITH_AES_128_CBC_SHA	
	TLS_RSA_WITH_NULL_SHA256	
	TLS_RSA_WITH_AES_128_CBC_SHA256	
	TLS_RSA_WITH_AES_128_GCM_SHA256	
	TLS_ECDHE_ECDSA_WITH_NULL_SHA	
	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	
	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	

Note: DEault_TO_LOCAL_NODE does not apply to the Local node record.

Note: Select Ciphers carefully since deprecated ciphers may not be available on all systems. Check with your Security Administrator before selecting these ciphers.

8. To enable client authentication:

- Type **Y** beside the **Client Auth** field.
- To have the common certificate name verified during the authentication process, select **Client Auth. Compare** and when the next panel displays, type the certificate common name of the local node certificate and press **Enter**. To not have the name verified, leave this field blank by not selecting the **Client Auth. Compare** field. If the common name is not entered, the client name verification process is not performed but client authentication is.

Note: This value is case-sensitive. Type it exactly as it appears in the certificate file.

9. To specify the certificate label:

- Select the **Certificate Label** field and press **Enter**.

Note: To use the default certificate of the key store, leave Certificate Label as blank.

- Press **F8** to move to the editable portion of the panel containing the label field.

- This field is case sensitive; therefore, type the label of the certificate exactly as you defined it when you generated it using one of the security applications described in [Configuration Worksheets](#), or type an asterisk (*) to specify the same label as the local node record, and press **Enter**. To use the Default Certificate defined in the keystore, leave the **Certificate Label** field blank.

Note: The Certificate Pathname field is automatically set to '*' (Default to Local) in the Remote Node record. You are not allowed to update this field for a remote node.

10. Select **OK** and press **Enter** to display the updated values.
11. Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you must resolve all errors that occur before you can save the parameter file.
12. Save the parameter file using the instructions in [Connect:Direct Secure Plus Operation Enablement and Validation](#).

Disabling Connect:Direct Secure Plus in a Remote Node Record

About this task

If you have remote nodes that do not use Connect:Direct Secure Plus, then you must disable all protocols for those node.

To disable all protocols in a remote node record imported from the network map:

Procedure

1. Type **U** next to the remote node record to update and press **Enter** to display the current values for the selected node.

Note: An asterisk in a field on the Secure+ Admin Main Screen indicates the value **Default to Local Node**.

```

File  Edit  Help
-----
CD.ZOS.NODE      Secure+ Admin Tool: Main Screen      Row 1 of 7
Option ==> -----      Scroll CSR

                        Table Line Commands are:

      U Update node      H View History      D Delete node
      I Insert node      V View node

      Node Filter :
      *-----

      LC Node Name      Type  Secure+      External Client
      -----      ---  Protocol Override Encryption      Auth      Auth
      ---
      .CLIENT          R    *              N              *              *
      .EASERVER         R    TLSV10          N              *              N
      .PASSWORD         R    Disabled        *              *              *
      CD.UNIX.NODE      R    TLSV10          *              *              *
      CD.UNIX.NODE2     R    TLSV12          *              *              *
      CD.ZOS.NODE       L    Disabled        Y              N              N
      CD.ZOS.NODE2     R    *              *              *              *
***** BOTTOM OF DATA *****

```

2. Select **EA Parameters** and press **Enter**.
3. In the **EA Parameters** panel, disable the External Authentication parameter by typing **N** beside the **External Auth** field, if necessary. The remaining external authentication parameters are unavailable because they are valid only for the .EASERVER remote node record.
4. Select **SSL/TSL Parameters** in the panel selection line and press **Enter**.
5. If necessary, disable the SSL and TLS protocols by typing **N** beside the **Enable SSL** and **Enable TLS** fields, if necessary.
6. Select **OK** and press **Enter** to display the updated values.
7. Read all warning and error messages. You can continue configuring the environment without resolving warning messages, but you may be unable to perform secure communications. You must resolve all errors before saving the parameter file.
8. Save the parameter file using the instructions in [Connect:Direct Secure Plus Operation Enablement and Validation](#).

Configuration for a Secure Connection between z/OS and OpenVMS Nodes

This topic provides a detailed example for defining a remote node record in both a Connect:Direct Secure Plus for z/OS parameter file and a Connect:Direct Secure Plus for OpenVMS parameter file to set up a secure connection between the two nodes.

In this example, two nodes have set up records in their respective Connect:Direct Secure Plus parameter files:

- The Connect:Direct Secure Plus for z/OS node is named Q1A.ZOA.V4700 and is defined in a remote node record in the Connect:Direct Secure Plus for OpenVMS parameter file and in the network map.
- The Connect:Direct for OpenVMS node is named Q1A.ITAN.V3400 and is defined in a remote node record in the Connect:Direct Secure Plus for z/OS parameter file and the network map.

The Connect:Direct Secure Plus records are defined to allow each node to act as either the client (PNODE) or the server (SNODE), depending on which one initiates the session.

Records Settings in the z/OS Parameter File

In the Connect:Direct Secure Plus for z/OS parameter file, the local node record has the following settings:

- **Y** in the **Override** field
- **N** for all Secure+ protocols, including SSL, TLS1.0, TLS1.1 and TLS1.2
- **N** in the **Client Auth** field

The settings for the local node record have the following effects: Disabling Connect:Direct Secure Plus in the local node record means that the protocol and other settings for secure connections must be defined in each remote node record; enabling the **Override** parameter allows settings in remote node records to override those in the local node record; client authentication is not enabled for all remote nodes.

The remote node record defined for the OpenVMS node named Q1A.ITAN.V3400 in the z/OS Connect:Direct Secure Plus parameter file has the following settings:

- **Node Identification** is Q1A.ITAN.V3400. This value must correspond to the node name specified in the Connect:Direct Secure Plus for z/OS network map.
- **Override** is not applicable in the remote record and defaults to **N**.
- The TLS 1.0 protocol is enabled for sessions to connect to this node.
- This OpenVMS node will not request client authentication of z/OS nodes with which it communicates.

The following **Secure+ Create/Update Panel - SSL/TLS Parameters** panel for Connect:Direct Secure Plus for z/OS illustrates the settings for the OpenVMS node named Q1A.ITAN.V3400 and commentary on the values set for the parameters.

The information in the bottom half of the screen pertains to the key certificate for the z/OS node. The OpenVMS remote node record for the z/OS node has enabled client authentication, as shown in [Records Settings in the z/OS Remote Node Record for OpenVMS Parameter File](#). Therefore, when the z/OS node initiates the session, the OpenVMS node (the server) requests that the client send its ID certificate so that the OpenVMS node can authenticate the client by validating the key certificate defined on this panel (mfcert_a) against the key certificate specified in the Root Certificate file field (mfcert_a.txt) of the z/OS remote node record in the Connect:Direct Secure Plus for OpenVMS parameter file, as illustrated in [Records Settings in the z/OS Remote Node Record for OpenVMS Parameter File](#). When the z/OS node is the server, it must send its public key, which is stored in the mfcert_a file, to the OpenVMS node during server authentication.

In this example, the z/OS key certificate resides in the default key database defined for the local node (indicated by *). If the certificate location does not default to the local node, the remote node definition must point to the absolute path. Definitions for the default key database are stored in the local node record. Certificate information identifying the z/OS node to remote nodes and remote nodes to the z/OS node is stored in the GSKKMAN database. When certificates are exchanged, trading partners send the ID

certificate portion of their keys to each other. In the z/OS system, this information must be imported into the GSKKMAN database.

Note: In the OpenVMS system, fully qualified paths are always required for file locations.

The TLS ciphers previously selected are shown using the standard two-byte IBM convention for displaying ciphers (352F04050A09030601). The systems negotiate a cipher suite common to both the z/OS and OpenVMS nodes to encrypt information during the handshake and when actual data is being transmitted.

Records Settings in the z/OS Remote Node Record for OpenVMS Parameter File

The following example shows the remote node record that defines the Connect:Direct for z/OS node named Q1A.ZOA.V4700. The OpenVMS network map contains an adjacent (remote) node record with the exact same name.

```
Node Name:      Q1A.ZOS.V4700
Node type:      R
1. Protocol:     TLS1.0
2. Client Authentication: y
3. Authentication timeout: 100
4. Certificate common name: mfsscert_a
5. Root Certificate file: disk$data:[qaitan.q1a]mfcert_a.txt
6. Key Certificate file: disk$data:[qaitan.q1a]2048sskeycert.txt
7. Passphrase:   ****
8. Cipher suites: EXP_RC4_MD5,RC4_MD5,RC4_SHA,EXP_RC2_CBC_MD5,IDEA_CBC_SHA,
EXP_DES_CBC_SHA,DES_CBC_SHA,DES_CBC3A
```

When the OpenVMS node is the server, it requests that the client authenticate itself (Client Authentication = Y) and send its certificate common name (mfsscert_a) for an extra layer of authentication. The public key information for the z/OS node is stored in the Root Certificate file named mfcert_a.txt; its location is specified (disk\$data:[qaitan.q1a]).

The key certificate file contains the information that identifies the OpenVMS node to other nodes (disk\$data:[qaitan.q1a]2048sskeycert.txt). In order for the OpenVMS system to access its private key to send information to the other node, the passphrase must be entered as well. The z/OS node validates this key certificate information against the information stored in its GSKYYMAN database.

The cipher suites are listed in the order of preference, and the first one that matches a cipher suite defined for the other node is used to establish a session.

Enable and Validate Connect:Direct Secure Plus Operation

After you initially configure the local and remote nodes for Connect:Direct Secure Plus for z/OS, save and submit the parameter file and prepare IBM Connect:Direct for operation. As a final step, validate and test connections between you and your business partners to establish secure communications and then test to make sure you can change your security defaults for a session.

During normal maintenance after you make changes to the parameter file, you can use the Save Active option to dynamically save updates without having to restart IBM Connect:Direct. See [Saving Changes to Node Records Using the Save Active Option](#).

If IBM Connect:Direct is running and Connect:Direct Secure Plus is enabled, you must use the Save Active option. Otherwise, you must use the Save As option. (When Connect:Direct Secure Plus is enabled, the Connect:Direct Secure Plus parameter file is open and IBM Connect:Direct cannot perform the Save As option since that option deletes, redefines, and reloads the parameter file.)

Saving and Submitting the Connect:Direct Secure Plus Parameter File

About this task

This procedure assumes that you have verified that the following required IBM Connect:Direct ISP libraries (all of the same release) have been allocated in your TSO session:

- \$CD.SDGAISPC (must be allocated as SYSPROC)
- \$CD.SDGALINK
- \$CD.SDGAPENU
- \$CD.SDGASENU
- \$CD.SDGMENU

If these required libraries have not been allocated, or have been allocated incorrectly, when you perform this procedure, the JCL for the SAVE AS job is not generated, and you have to repeat the procedures to configure the local and remote nodes. For information on how to allocate these libraries, see *IBM Connect:Direct for z/OS Configuration Guide* and *Program Directory for IBM Connect:Direct for z/OS*.

To save the Connect:Direct Secure Plus parameter file:

Procedure

1. From the Connect:Direct Secure Plus Admin Tool: Main Screen, select **File** and press **Enter**.

```

File  Edit  Help
+-----+-----+-----+-----+-----+-----+-----+-----+
1  1. New
   2. Open
   3. Close
   4. Info...
   *. Rekey
   6. Save Active
   7. Save as...
   8. Unload
+-----+-----+-----+-----+-----+-----+-----+
9. Exit
+-----+-----+-----+-----+-----+-----+-----+
LC Node Name      Type  Secure+  Protocol  Override  Encryption  External  Client
                               Auth      Auth
--- .CLIENT       R    *        N          *          *          *
--- .EASERVER      R    *        N          *          N          *
--- .PASSWORD      R    Disabled *          *          *          *
--- CD.UNIX.NODE   R    TLSV10   *          *          *          *
--- CD.UNIX.NODE2  R    TLSV12   *          *          *          *
--- CD.ZOS.NODE    L    Disabled Y          N          N          N
--- CD.ZOS.NODE2   R    *        *          *          *          *
***** BOTTOM OF DATA *****

```

2. Type **7** to select **Save As**.
3. Type the file name you want to use for the Connect:Direct Secure Plus parameter file and press **Enter**. (You will use this same file name in [step 1](#) when you add this information as a parameter to the IBM Connect:Direct initialization parameter file to tell IBM Connect:Direct where security information is located.)



CAUTION: The default **Save As** file name is the name of the last file that you opened. When you create the Connect:Direct Secure Plus parameter file from the IBM Connect:Direct network map, you risk overwriting the network map file with the Connect:Direct Secure Plus parameter file if you do not change the name in this field. If the file specified here exists, you will be prompted that it will be deleted, redefined, and built anew if you select OK.

4. On the **Save As** information panel, type site-specific job card information, allocation information, STEPLIB DSNs, and Access file Dsname, using the library names created when you saved the parameter file.



CAUTION: For the Access file Dsname, you must not use the same name specified for the Connect:Direct Secure Plus parameter file. However, the Access file and Connect:Direct Secure Plus parameter file act as a pair and one is not any good without the other. To more easily track the files if necessary, it is recommended that at least the first two qualifiers match. For example, you could use \$CD.SECURE.PARM and \$CD.SECURE.ACCESS for the names of the related Connect:Direct Secure Plus parameter and access files.

5. Type **3** to select **Submit** and press **Enter** to save your parameter file. (If you need to change anything later, type **2** to edit the JCL before submitting the job. Edit the JCL and then submit the job.)



CAUTION: Closing the JCL without submitting the job loses all of the changes you made to the parameter file.

6. Research any return code other than zero before closing the parameter file or exiting the Admin Tool.

Preparing IBM Connect:Direct for Secure Plus Operations

About this task

After you set up the Connect:Direct Secure Plus environment, you must prepare IBM Connect:Direct to use Connect:Direct Secure Plus.

To set up IBM Connect:Direct to run with Connect:Direct Secure Plus:

Procedure

1. Add the following parameter to the Connect:Direct for z/OS initialization parameters:
SECURE.DSN=filename, where **filename** is the name of the Connect:Direct Secure Plus parameter file for that node.
2. If you are operating in a CD/Plex environment or if you did not specify the entire path to the Certificate Path Name on the Local record, add the **SECURE.SSL.Path.Prefix=prefix** parameter, where **prefix** is the prefix location of the key database or key ring that contains the certificates for the TLS or SSL protocol.
3. Restart IBM Connect:Direct on that node.
4. To verify that Connect:Direct Secure Plus for z/OS initialization is complete, after you restart IBM Connect:Direct with SECURE.DSN, review the started task output for the following messages:
SITA028I Secure+ initialization and *SITA165I Secure+ initialization complete*, if you are using the TLS or SSL protocol.

What to do next

See [“Planning for System SSL in FIPS Mode”](#) on page 751.

Parameter File Saving After the Initial Setup

After you save the Connect:Direct Secure Plus parameter file the first time using the **Save As** option, you must stop IBM Connect:Direct before using the **Save As** option again, and then restart IBM Connect:Direct.

When you are maintaining Connect:Direct Secure Plus for z/OS and want to save your changes without recycling IBM Connect:Direct, use the Save Active option. For more information, see [Saving Changes to Node Records Using the Save Active Option](#).

Validating and Testing Connections by Session

About this task

To validate and test a connection between two business partners, follow this general procedure. After you confirm that the secure connection has been established and that you can change your default security

settings for a session, you can finalize the settings in the Connect:Direct Secure Plus parameter file of each business partner, save the files, and begin transferring data.

Procedure

1. For the selected protocol, make sure all prerequisites outside of Connect:Direct Secure Plus have been taken care of, such as the obtaining of server certificates and exchanging of keys.
2. Make sure each node is defined in the partner's network map.
3. For both the local and remote nodes, specify the protocol to be used when a secure connection is required (TLS or SSL).
4. For the selected protocol, make sure to define all settings required for a successful connection in the local and remote node records in the parameter files.
5. Perform the procedures in this chapter, namely, [Saving and Submitting the Connect:Direct Secure Plus Parameter File](#), and [“Preparing IBM Connect:Direct for Secure Plus Operations”](#) on page 803.
6. To test the connection, perform a file transfer between the two partners.

Once you have successfully performed a file transfer using a secure connection, you are ready to finalize the parameter files.

7. Take one of the following actions, depending on whether you want to make your sessions default to secure or non-secure:
 - To have your sessions default to secure, specify **OVERRIDE=Y** in both the local and remote node records in the parameter files of both business partners.
 - To have your sessions default to non-secure, specify **OVERRIDE=Y** in both the local and remote node records in the parameter files of both business partners. Disable the selected protocol in the remote node record.
8. To test changing your security defaults for a session, take one of the following actions depending on whether you want to make your sessions default to secure or non-secure. For a complete description of the **SECURE** parameter and how to use it in the PROCESS statement, see the *IBM Connect:Direct Process Language Reference Guide*. Also, see [Security Settings Override Examples](#).
 - To make a session non-secure, specify **SECURE=OFF** in the PROCESS statement preceding the COPY statement to transfer the file.
 - To make a session secure, specify **SECURE=OFF|SSL|TLS|TLS11|TLS12** in the PROCESS statement.
9. After you valid and test your connections by session, save the parameter files and restart IBM Connect:Direct.

Override Settings in IBM Connect:Direct Processes

After you configured Connect:Direct Secure Plus, security is either turned on or off each time that you use IBM Connect:Direct with a node defined in the Connect:Direct Secure Plus parameter file. However, you can override some default security settings in a remote node record from a IBM Connect:Direct Process using the **SECURE** parameter in the PROCESS or COPY statement.

Note: The more flexible you make the environment by allowing override the less security the environment becomes. Allowing override in the Remote record allows the SNODE to override the security settings as well.

To allow a business partner to override the default security setting of whether security is turned on or off for another business partner and to choose the protocol for the remote node, the following conditions must be in place:

- Each business partner agrees all sessions are secure or non-secure as the default
- Each business partner agrees to allow the override of the Connect:Direct Secure Plus parameters by specifying **OVERRIDE=Y** for both the local and remote nodes in their Connect:Direct Secure Plus parameter file.

- The remote node definition in each Connect:Direct Secure Plus parameter file specifies the parameters necessary for a secure session even if the protocol is disabled including all information necessary for exchanging and validating each partner's identity. All parameters related to a protocol are defined, such as SSL/TLS cipher suites and key databases.
- Connect:Direct Secure Plus is active on both nodes.

Once the Connect:Direct Secure Plus parameter files for both business partners have been set up properly, you can override the default security settings on a Process-by-Process basis to perform exception processing.

For a complete description of the SECURE parameter and how to use it in the PROCESS or COPY statement, see the *IBM Connect:Direct Process Language Reference Guide*.

PROCESS Statement Overrides for Connect:Direct Secure Plus Defaults

The first statement in all IBM Connect:Direct Processes is the PROCESS statement which defines the attributes of a Process. The SECURE keyword in the PROCESS statement allows you to perform one or more of the following functions:

- Turn on security when non-secure sessions are the default
- Select the protocol (SSL or TLS) when non-secure sessions are the default
- Specify one or more cipher suites to override the default cipher suites defined in the Connect:Direct Secure Plus parameter file
- Turn off security when secure sessions are the default (if **OVERRIDE=Y** is specified in the Remote Node record settings in the Connect:Direct Secure Plus parameter file).
- Encrypt only the control block information contained in Function Management Headers (FMHs), such as a user ID, password, and filename. (The default is to encrypt both the control block information and the data being transferred.)

Note: If System SSL is in FIPS mode, TLS is the only supported protocol. See [“Planning for System SSL in FIPS Mode”](#) on page 751.



Warning: The use of TCPNAME in PROCESS is not supported when Secure+ is used. If this value is specified like the following example: the process can either fail, or not use Secure+:

```
SNODE=TCPNAME=&SNODE
```

The following syntax example shows the options available for the SECURE keyword:

```
SECURE=OFF | SSL | TLS | TLS11 | TLS12
or
SECURE=ENCRYPT.DATA=Y | N
or
SECURE = (OFF | SSL | TLS | TLS11 | TLS12, ENCRYPT.DATA=Y | N)
or
SECURE = (OFF | SSL | TLS | TLS11 | TLS12, <cipher_suite> | (cipher_suite_list), ENCRYPT.DATA=Y | N)
```

If you use multiple **SECURE** subparameters, **ENCRYPT.DATA** must be the last (or only) value specified on the **SECURE=** parameter.

COPY Statement Overrides for Connect:Direct Secure Plus Defaults

By using the COPY statement's SECURE parameter in a IBM Connect:Direct Process to override the settings in the Connect:Direct Secure Plus parameter file and enabling the override feature in the remote node record, you can disable security for a particular file transfer. Connect:Direct Secure Plus uses the most secure connection available. Therefore, if the remote node record enables encryption, the PNODE cannot turn those options off using the COPY statement override.

After the security settings of the PNODE and SNODE are merged, the strongest setting is always used. Therefore, the value specified from the COPY statement cannot disable data encryption or digital signatures if the SNODE has enabled them.

In an SSL or TLS environment, the following syntax example shows the options available for the **SECURE** keyword in a **COPY** statement (for the destination file that you are copying to):

```
SECURE = ENCRYPT.DATA=Y|N  
or  
SECURE = ENC=Y|N
```

Security Settings Override Examples

These examples illustrate how business partners use the **SECURE** parameter to override the security defaults for a particular session.

Secure Sessions Default Override

The business partners agree by default all sessions are secure and choose SSL as the default protocol. Both partners enable the SSL protocol in the Connect:Direct Secure Plus parameter files and specify **OVERRIDE=Y** in both the Local and Remote Node records.

To override the default and make a particular session non-secure, they use the following **PROCESS** statement:

```
SSLOFF PROCESS SNODE=OTHERBP SECURE=OFF
```

Non-Secure Sessions Default Override

The business partners agree by default all sessions are non-secure. When a secure communication line is required for a particular session, the non-secure default is overridden and the SSL protocol used. The Remote Node records specify **OVERRIDE=Y**, but the SSL protocol is not enabled in the Connect:Direct Secure Plus parameter files. However, all other parameters required to perform the handshake to establish an SSL session are defined in the Remote Node records. To specify that the session for this **PROCESS** is to be secure using SSL, the business partners use the following **PROCESS** statement:

```
SSLON PROCESS SNODE=OTHERBP SECURE=SSL
```

Default Cipher Suite Override

The business partners agreed by default all sessions are secure and chose TLS as the default protocol. Both partners specified the following configuration in their Connect:Direct Secure Plus parameter files:

- Enabled the TLS protocol
- Specified **OVERRIDE=Y** in both the Local and Remote Node records
- Selected **TLS_RSA_WITH_RC4_128_MD5** as the cipher suite to use when executing Processes

To override the default cipher suite and use **TLS_RSA_WITH_3DES_EDE_CBC_SHA** when executing a particular Process, they use the following **PROCESS** statement:

```
NEWCIPHER PROCESS SNODE=OTHERBP SECURE=(TLS,TLS_RSA_WITH_3DES_EDE_CBC_SHA)
```

Cipher Suite List Override of Single Default Cipher Suite

The business partners agreed by default all sessions are secure and chose TLS as the default protocol. Both partners specified the following configuration in their Connect:Direct Secure Plus parameter files:

- Enabled the TLS protocol
- Specified **OVERRIDE=Y** in both the Local and Remote Node records
- Selected **TLS_RSA_WITH_RC4_128_MD5** as the cipher suite to use when executing Processes

To override the default protocol and use a list of other TLS cipher suites when executing a particular Process, they use the following PROCESS statement:

```
NEWCIPHERS PROCESS SNODE=OTHERBP SECURE=(TLS,(TLS_RSA_WITH_3DES_EDE_CBC_SHA,
TLS_RSA_AES_128_SHA,TLS_RSA_AES_256_SHA,TLS_RSA_WITH_DES_CBC_SHA) )
```

Control Block and Data Encryption Default Override

The business partners agreed by default to encrypt all information sent during the handshake to set up communication sessions and the actual files being transferred. Both partners specified the following configuration in their Connect:Direct Secure Plus parameter files:

- Specified **ENCRYPT=Y** in both the Local and Remote Node records
- Specified **OVERRIDE=Y** in both the Local and Remote Node records

To not go through the expense of encrypting and decrypting data being transferred, they use the following PROCESS statement when transferring a particular file:

```
ENCNO PROCESS SNODE=OTHERBP SECURE=ENCRYPT.DATA=N
```

In this scenario, both business partners are more concerned with increasing throughput and using less CPU while protecting the information being exchanged to establish the session.

Maintain Connect:Direct Secure Plus

After you set up the Connect:Direct Secure Plus environment, you will need to maintain both the Connect:Direct Secure Plus parameter file and the records it contains whenever changes in your system or a trading partner's environment require updates. For SSL and TLS node records, you can also use Control Center to display and modify information, and to delete nodes.

After you make changes, refer to [Saving Changes to Node Records Using the Save Active Option](#) to put the updates in effect immediately.

Parameter File Maintenance

The File Menu on the Secure+ Admin Tool: Main Screen contains options for maintaining the parameter file. To open this menu, select **File** from the action bar and press **Enter**. The following panel sample shows the available options when a parameter file is already open.

```

File Edit Help
+-----+ Secure+ Admin Tool: Main Screen Row 1 of 7
| 1 1. New |-----| Scroll CSR
| 2. Open |
| 3. Close |
| 4. Info |
| *. Report |
| *. Save Active |
| 7. Save as... |
| 8. Unload |
| 9. Exit |
+-----+
+-----+ Secure+ External Client
LC Node Name Type Protocol Override Encryption Auth Auth
+-----+
-- .CLIENT R * N * * *
-- .PASSWORD R * * * *
*
-- CD.ZOS.LNODE L TLSV12 Y Y N Y
-- CD.UNIX.RNODE R * * Y * N
***** BOTTOM OF DATA *****

```

The following options are available on the File menu:

- 1—**New (FN)** checks if any current table record has been modified and prompts to save, and then the table will be reset to empty. To create a new parameter file, select the **Create/Update** option on the **Edit** Menu or the **Insert** node table line command (see [Inserting a Node](#)).
- 2—**Open (FO)** displays the File Selection panel which allows you select the parameter file you want to open. For more information, see [Opening a Connect:Direct Secure Plus Parameter File](#).
- 3—**Close (FC)** shuts the File menu allowing you to select options from the Secure+ Admin Tool. Main Screen.
- 4—**Info (FI)** displays general information about the Connect:Direct Secure Plus parameter file, such as the version of Connect:Direct Secure Plus that you are using, the name of the parameter file, and the number of remote node records it contains. For more information, see [Viewing Information about the Parameter File](#).
- 5— **Report (FR)** produces a report of all entries in the current open Parameter file. Report is generated into a temporary data set for browsing. You can select the data and save in a member for printing.
- 6—**Save Active (FS)** allows you to dynamically update records in the parameter file without bringing IBM Connect:Direct down. For more information, see [Saving Changes to Node Records Using the Save Active Option](#).
- 7—**Save as (FA)** is the option you use the first time you save the parameter file. Subsequently IBM Connect:Direct must be down to use this option which saves updates to the parameter file and you have to submit the job which restarts IBM Connect:Direct. For more information, see [Saving and Submitting the Connect:Direct Secure Plus Parameter File](#).
- 8—**Unload (FX)** retrieves information from the Connect:Direct Secure Plus parameter file and displays it as an ISPF TMP data set on your screen.
- 9—**Exit (FE)** takes you out of the Secure+ Admin Tool and displays the **Connect:Direct Administrative Options Menu**.

Note: The code within the () parameter for example, **(FS)** is a Fast Path. The code can be directly entered on the Option line to perform its corresponding function without using the menu.

Opening a Connect:Direct Secure Plus Parameter File

About this task

Before you can modify node records, you must open the parameter file that contains these records.

To open a Connect:Direct Secure Plus parameter file:

Procedure

1. With the **Secure+ Admin Tool Main Screen** open, select **File** and press **Enter**:
2. Type **2** to select **Open** and press **Enter** to display the file selection screen:

Secure+ Admin Tool: File Selection	
Enter file name for: INPUT SECURE PARM FILE	
File Name: \$CD.SECURE.PARMFILE	Browse
File System Type: 1 1. MVS 2. HFS	Cancel

3. Type the complete parameter file name, prefix or partial prefix followed by an asterisk (*), select **Browse**, and press **Enter**. If a prefix is entered as a parameter file name, the following displays:

```
Secure+ Admin Tool: File Selection                                Row 1 of 3

Option: _____ Scroll CSR

Enter "S" on the line of the file for  for MVS.

LC Filename or Directory
S $CD.PARMFILE
- $CD.PARMFILE.DATA
- $CD.PARMFILE.INDEX
***** Bottom of data *****
```

Note: You can also type the complete parameter file name and press **Enter**.

4. Type **S** next to the file name to open and press **Enter**. The **Secure+ Admin Tool: Main Screen** displays nodes populated from the parameter file you opened. See [Node Record Maintenance](#) for more information on this panel and how to select the available options.

Viewing Information about the Parameter File

About this task

To view information about the Connect:Direct Secure Plus parameter file:

Procedure

1. Open the Admin Tool.
2. Select **File** and press Enter.
3. Type **4** to select **Info**. The **File Information Panel** is displayed:

```
+----- Secure+ Admin Tool File Information Panel -----+
|
|      Secure+ Admin Tool File Information Panel
|
|      +-----+
|      | Secure+ Admin for Connect:Direct for z/OS      |
|      | Node: CSG.PROD390                               |
|      | Name Filter Applied: *                          |
|      | File: $CD.SECURE.PARMFILE                      |
|      | $CD.SECURE.ACCESS                               |
|      | Update                                           |
|      | Events:           0   Current                   |
|      |                               Records:           |
|      |                               9                   |
|      | Toolkit msg/Rc:CSPA000I/           0             |
|      | Last 3                                           |
|      | Events:                                           |
|      +-----+
|
+-----+
```

The fields in the **File Information Panel** are described in the following table:

Field Name	Description
Node	The name of the local node for the parameter file that is open.
Admin Version	The version of Secure+ Admin Tool being used.
Name Filter Applied	Name of the filter used to determine which remote node records to display.

Field Name	Description
File	The name of the current parameter file and the access file.
Update Events	Number of updates to the parameter file.
Current® Records	Total number of remote node records.
Toolkit msg/Rc:CSPA0001	Message ID of the last Toolkit call.
Last 3 Events	List of the last 3 updates.

Saving Changes to Node Records Using the Save Active Option

About this task

The **Save Active** option on the Secure+ Admin Tool **File** menu enables you to dynamically save changes to all records in an existing Connect:Direct Secure Plus parameter file.

You must use the **Save As** option the first time you create the parameter file. This option deletes, defines, and reloads the Connect:Direct Secure Plus parameter file. See [Saving and Submitting the Connect:Direct Secure Plus Parameter File](#).

Note: Connect:Direct Secure Plus for z/OS must be running in order to use the **Save Active** option. Any changes made to node records take effect immediately.

To save the Connect:Direct Secure Plus parameter file and put any changes made in effect immediately:

Procedure

1. Open the Admin Tool.
2. Select **File** and press **Enter**.
3. Type **6** to select **Save Active**.
4. Read all warning and error messages. You can continue without resolving warning messages, but you must resolve all errors before you save the parameter file.



CAUTION: Because Connect:Direct Secure Plus for z/OS cannot restrict multiple users from attempting to apply updates to the same parameter file, measures should be in place to ensure that only one user is accessing a particular parameter file at one time. If precautions are not taken, updates will be lost and conflicts among multiple user interfaces will occur.

Re-securing the Parameter File and Access File

About this task

Routinely, or if your passphrase is compromised, you should re secure the Connect:Direct Secure Plus parameter and access files. You must open a parameter file before you perform this procedure.

Note: Re-securing the parameter file requires a "Save As" function. This procedure must be executed when the IBM Connect:Direct is down.

To re secure the Connect:Direct Secure Plus parameter file and access file:

Procedure

1. From the **Secure+ Admin Tool Main Screen**, select **File** and press **Enter**.
2. Type **7** to select **Save As** and press **Enter**.
3. If any warning messages are displayed, read them and press **F3** to close the warning panel.
4. On the **File Selection** panel, the file name of the parameter file that you have open is displayed. Press **Enter**.

5. At the confirmation prompt, select **OK**. The old parameter file is deleted and a new parameter file with the same name is created.
6. On the **Save As** screen, type **2** to select **Edit**, then select **Make Pass Phrase** by positioning the cursor on **Make** and press **Enter**.
7. Select **OK** to confirm that you want to create a new passphrase.
8. Type a 32-byte string, using uppercase, lowercase, numeric, and alphabetic characters.
9. On the **Save As** panel, type **3** to select **Submit** and press **Enter**.
10. Select **OK** to submit the job.
11. When the *Job Submitted* message is displayed, press **Enter**.
12. Verify that the job completed with a return code of zero before closing the parameter file or exiting the **Secure+ Admin Tool**. Research any return codes other than zero.

Node Record Maintenance

When you start the Secure Plus Admin Tool, the main screen is displayed.

Secure Plus Admin Tool Main Screen						
<pre> File Edit Help ----- MY_LOCAL Secure+ Admin Tool: Main Screen Row 1 of 4 Scroll Option ==> CSR Table Line Commands are: U Update node H View History D Delete node I Insert node V View node CL Clone node Node Filter : * Client Secure+ External LC Node Name Type Protocol Override Encryption Auth Auth ----- .CLIENT R Disabled N * * * .PASSWORD R Disabled * * * * MY_LOCAL L TLSV12 Y Y N Y MY_REMOTE R * * Y * N ***** BOTTOM OF DATA ***** </pre>						

The following table line commands are available to use on the nodes displayed:

- **U Update node** allows you to perform the following functions:
 - [Disabling Connect:Direct Secure Plus on an SSL or TLS Node](#)
 - [Changing the Cipher Suites](#)
- **I Insert node** allows you to create a new remote node record. For more information, see [Inserting a Node](#).
- **H View History** lists the individual dates including the time the selected node was updated.

- **V View node** allows you to view the remote node record.
- **D Delete node** allows you to delete the selected node. For more information, see [Deleting a Remote Node Record](#).
- **CL Clone Node** – Allows for easy cloning (copy) an entire Parameter File or cloning single Remote records. For more information see, [“Cloning a parameter file” on page 814](#).

Disabling Connect:Direct Secure Plus on an SSL or TLS Node

To disable Connect:Direct Secure Plus on a SSL or TLS node:

Procedure

1. From the **Secure+ Admin Tool Main Screen**, type **U** next to the SSL or TLS node to update and press **Enter**. The **Secure+ Create/Update Panel** displays the information for the selected node.
2. Type **N** beside the **Enable SSL** and **Enable TLS** fields. Select **OK** and press **Enter**.
3. Select **OK** and press **Enter**.
4. Save the Connect:Direct Secure Plus parameter file using the procedure in [Saving Changes to Node Records Using the Save Active Option](#).

Note: To continue IBM Connect:Direct operations with Connect:Direct Secure Plus disabled, *both* trading partners must disable Connect:Direct Secure Plus.

Changing the Cipher Suites

About this task

When you activate the SSL or the TLS protocol for a node, cipher suites are used to encrypt transmitted data. The same cipher suite must be defined at both ends of the transmission. Connect:Direct Secure Plus searches the enabled cipher suite list and locates the first cipher suite that is common for communications at both the PNODE and the SNODE. It then uses this cipher suite to encrypt data. You defined cipher suites when you configured the local node record.

Note: If System SSL is in FIPS mode, only certain ciphers are valid. See the *IBM Connect:Direct for z/OS Release Notes* for a list of valid FIPS-mode ciphers.

To change the cipher suites enabled for a node and the priorities assigned to them:

Procedure

1. From the **Secure+ Admin Tool Main Screen**, type **U** next to the node to update.
2. On the **Create/Update Panel**, select the **Cipher Suites** field and press **Enter** to display the **Update Cipher Suites** panel.

More: +

Update the order field below to enable and order cipher suites.

Order	All Available Cipher-Suites	Enabled Cipher-Suites
1	SSL_RSA_AES_128_SHA	SSL_RSA_AES_128_SHA
2	SSL_RSA_AES_256_SHA	SSL_RSA_AES_256_SHA
3	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA
4	SSL_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA
5	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
6	SSL_RSA_WITH_RC4_128_SHA	SSL_RSA_WITH_RC4_128_SHA
7	SSL_RSA_WITH_RC4_128_MD5	SSL_RSA_WITH_RC4_128_MD5
8	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSL_RSA_EXPORT_WITH_RC4_40_MD5
9	SSL_RSA_WITH_NULL_SHA	SSL_RSA_WITH_NULL_SHA
10	SSL_RSA_WITH_NULL_MD5	SSL_RSA_WITH_NULL_MD5

The list on the left side contains all available cipher suites. The active cipher suites are listed on the right side of the screen and are assigned a numerical order in the **Order** column on the left side of the screen.

3. Take one or more of the following actions as needed:

- Type **1** by the cipher you want to enable and give the highest priority. Type **2** by the cipher suite you want to enable and place second in priority. Continue typing numbers next to the ciphers you want to enable (a maximum 10), in order of priority. The ciphers you enable appear in the order of priority in the **Enabled Cipher-Suites** list.
- To deactivate a cipher suite, clear the number in the **Order** field and press **Enter**.
- To change the order of a cipher suite, type new numbers in the **Order** fields of the cipher suites to reorder and press **Enter**.

4. Press **PF3** to save the new enabled cipher-suite list and return to the **Secure+ Create/Update Panel**.

5. Save the parameter file using the procedure described in [Saving Changes to Node Records Using the Save Active Option](#).

Inserting a Node

About this task

The **Insert** node option follows the same procedure as creating a remote node manually using the **Edit** menu.

To create a remote node:

Procedure

1. To create a remote node, type **I** next to any node on the **Secure+ Admin Tool Main Screen** and press **Enter**.
2. Type the name of the existing remote node that corresponds to its name in the network map and enter the rest of the information on the **Node Identification** panel. The **Secure+ Create/Update Panel** is displayed.
3. After you finish the procedure and save the record by selecting **OK** and pressing **Enter**, save the parameter file using the procedure described in [Saving Changes to Node Records Using the Save Active Option](#).

Deleting a Remote Node Record

If you remove a remote node record from the network map in IBM Connect:Direct, you can also remove it from the Connect:Direct Secure Plus parameter file. This process deletes nodes from the Connect:Direct Secure Plus parameter file.

About this task

Note: If you have implemented the Strong Password Encryption feature, you cannot use the **Delete** node table line command to delete the **PASSWORD** record. To determine if SPE is in effect, see [Disabling Strong Password Encryption](#) for instructions on how to access the SPE Parameters screen. You will see either the message, *SPE not currently in use* or *SPE currently in use* on that panel. If SPE is not in use, you can delete the **PASSWORD** record; otherwise, follow the procedure to disable SPE, and then reinitialize IBM Connect:Direct. This initialization with SPE disabled will remove any SPE encryption that has been previously applied.

To delete a remote node record:

Procedure

1. Type **D** next to the node to delete and press **Enter**.

The Connect:Direct Secure Plus **Confirmation Prompt** displays the message *Are you sure you want to delete 'selected node'?*.

2. Select **OK** and press **Enter** to delete the record.
3. Save the Connect:Direct Secure Plus parameter file using the procedure in [Saving Changes to Node Records Using the Save Active Option](#).

Cloning a parameter file

The SPAdmin tool introduces a new line command, CL to perform the cloning process. The CL line command can be used on the Local record as long as there are no outstanding unsaved updates to the Secure Parameter file. The CL line command can also be used on a Remote record to quickly replicate that record.



CAUTION: It is recommended that you take a backup of the Secure Parameter and Secure Access files before you attempt the cloning process. The SPAdmin tool takes all precautions to protect your data but a backup provides an extra security step to prevent lost of data.

The following section provides information on cloning process for different record types.

Alias Record Clone

An Alias Record cannot be cloned and the SPAdmin tool will issue an appropriate error if the CL line command is attempted for a record type of A.

Remote Record Clone

A remote record clone retains the settings from the original record and allows for entry of a new node name. This allows for a quick and easy replication of records.

Local Record Clone

A local record clone turns the local record into a remote record, retaining all the settings from the original local and allow for entry of a new node name. This allows for quick and easy replication of a PARMFILE. The local record clone requires a **Save As** function to be performed; the SPAdmin dialog perform that automatically.



CAUTION:

When performing a local record clone, there are several prompts throughout the process to provide ample time to cancel the function. Once the updates are saved they cannot be reversed.

To clone a PARAM file using Local record clone follow the steps given below:

1. Execute the CL line command on the Local record
2. At the Local clone prompt confirmation prompt, select **OK**
3. Type a new Local Nodename on the **Create/Update Panel** and select **OK** at the confirmation prompt.
4. At the Local clone prompt confirmation prompt, select **OK**
5. Type a new PARMFILE name in the **Output Secure Parm File** field
6. Type a new ACCESS name in the **Access file** field
7. Select **OK** at the Confirmation prompt before the submit to allow JOB to run with RCO
8. Enter the clone PARMFILE name in the **Input Secure Parm File** field

Connect:Direct Secure Plus Statistics

IBM Connect:Direct logs statistics for IBM Connect:Direct Process activity. The IBM Connect:Direct statistics include Connect:Direct Secure Plus information for a Process.

The following samples of IBM Connect:Direct Process statistics records contain information for Connect:Direct Secure Plus support. For information about viewing Connect:Direct for z/OS Process statistics, refer to the *IBM Connect:Direct for z/OS User's Guide*.

SSL or TLS Statistics Record

When you use the **Select Statistics** command to view the information about a IBM Connect:Direct Process that uses SSL or TLS security, you see a screen similar to the following. Be sure to use the Display option by typing **D** on the CMD line to see the detailed version of the SELECT STATISTICS report and specify extended options to see the Session Begin record. The Connect:Direct Secure Plus fields are in bold. A description for the fields follows the samples.

Select Statistics Output

```

=====
CD.NODEA                SELECT STATISTICS                DATE : 08.28.2018
=====
Function      => Session Begin                Start Time  => 10:27:51
                                           Start Date  => 08.30.2018

Process Name => STATS60A
Process Num  => 1                               Comp Code   => 00000000
                                           Comp Msg    => SVTM055I

Session Class=> 001
Userid       => USER11
Primary Node  => CD.NODEA
Secondary Node => CD.NODEB
Submitter Node => CD.NODEA
TLS V1.0 Enabled => Yes      (Process Override)
TLS Ciphersuite => TLS_RSA_WITH_AES_128_CBC_SHA      (Process Override)
Subject => (SN=52:9e:29:dc:00:0b:e5:14/C=US/ST=TEXAS/L=IRVING/O=IBM/OU=Quality/CN=ursa1024sha256/)
Issuer  => (C=US/ST=TEXAS/L=IRVING/O=IBM/OU=Quality/CN=ursa1024sha256/)

Session Protocol      => TCP
Socket for Origin     => 15053 ; 10.120.131.91
Socket for Destination => 05610 ; 10.120.131.92
Bind Attempts => 1
Remote Node Communications Address => 10.120.131.92

-----
Function      => COPY                Step Start Time => 10:27:51
Process Name  => STATS60A            Step End Time  => 10:28:35
Process Num   => 1
SMFID         => SYSK                Comp Code      => 00000000
Userid        => USER11              Comp Msg       => SCPA000I
Job Name      => USER11              Job ID         => TSU04479
Secondary Node => CD.NODEB            Step Name      => STEP1
Pnode Version=> 6.00.00              Snode Version=> 6.00.00
Other addr    => 10.120.131.92
Other port    => 05610
Session Class=> 001

Session Protocol      => TCP
FASP                 => No
    Session is not with SSP
V2 Buffer Size        => 262,144
Negotiated V2 Buffer Size => 262,144
TCP Buffer Size Used   => 262,144
Compression Control Feature: ON
Negotiated: NO Compression

TLS V1.0 Enabled      => Yes      Encrypt.Data=Y      (Process Override)
TLS Ciphersuite      => TLS_RSA_WITH_AES_128_CBC_SHA      (Process Override)
Subject => (SN=52:9e:29:dc:00:0b:e5:14/C=US/ST=TEXAS/L=IRVING/O=IBM/OU=Quality/CN=ursa1024sha256/)
Issuer  => (C=US/ST=TEXAS/L=IRVING/O=IBM/OU=Quality/CN=ursa1024sha256/)

```

Select Statistics Output (*continued*)

```

***** CHECKPOINTED;      Interval => 300,000,000
From ( Pnode
Dsn=>USER11.SMSTEST.DSN1)
  DCB => LRECL => 00080  BLKSIZE => 27920  RECFM => FB   DSORG => PS-E
  DISP=> (S, , )
    recs => 0                      blks => 358,167
  Bytes Read => 10,000,000,000
  Bytes Sent => 10,024,288,983
  Cmpr Perc => - 0.3%
  VOL=SER => ARTS09 ARTS0F ARTS0D ARTS08 ARTS06 ARTS0B
             ARTS04 ARTS05
  Time on CP => 00:00:14.598
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000
To ( Snode
Dsn=>USER11.SMSTEST.DSN2)
  DCB => LRECL => 00080  BLKSIZE => 27920  RECFM => FB   DSORG => PS-E
  DISP=> (R, , )
    recs => 0                      blks => 358,167
  Bytes Written => 10,000,000,000
  Bytes Received=> 10,024,289,751
  Cmpr Perc => - 0.3%
  VOL=SER => ARTS0C ARTS0F ARTS0E ARTS0D ARTS08 ARTS09
             ARTS06 ARTS0B
  Time on CP => 00:00:15.084
  Time on zIIP => 00:00:00.000
  zIIP Qualify => 00:00:00.000

```

The following statistics are displayed for the copy function:

Field	Description	Valid Values
TLS (or SSL) Enabled	Specifies whether TLS (or SSL) x.509 certificate use is enabled.	Yes No
TLS (or SSL) Ciphersuite	Specifies the cipher suite used in the session and whether the Process defined an override in the Copy termination record.	Any valid cipher suite
Subject	Specifies the subject name on the certificate.	Any valid subject name
Issuer	Specifies the issuer name on the certificate.	Any valid issuer name

Troubleshooting

Use the following table to help troubleshoot problems with Connect:Direct Secure Plus:

Note: For all errors related to Strong Password Encryption, see [SPE Problem Troubleshooting](#).

Problem	Possible Cause	Solution
System initialization failed, and the following SITA196E error message is displayed: FIPS Mode Requested but SECURE.DSN parameter is not specified.	You specified the FIPS initialization parameter as YES, but you did not specify the SECURE.DSN parameter to enable Connect:Direct Secure Plus.	Update the initialization parameters and restart IBM Connect:Direct.
System initialization failed, and the following error message is displayed: Connect:Direct FIPS keyword requires z/OS release 1.11 or later.	Your current, active z/OS release level does not support FIPS mode for System SSL.	Either update the FIPS initialization parameter to NO or execute IBM Connect:Direct on the appropriate release level of z/OS.

Problem	Possible Cause	Solution
IBM Connect:Direct was terminated, and the following error message is displayed: Secure+ Severe FIPS Mode Error, &var1.	<p>During operation of a TLS FIPS mode request, a severe error occurred causing IBM Connect:Direct to terminate with a U4079 abend due to one of the following:</p> <ul style="list-style-type: none"> • KEY database (not FIPS-mode) • Random number generation failure • RSA or DSA keypair generation failure • gsk_perform_kat API failure 	Contact IBM Support or correct the error and restart IBM Connect:Direct .
The following message is received at startup: SITA166I or SITA167I Secure+ SSL or TLS initialization failed. rc=00000134, rs=NO DFLT UNIX PATH.	The IBM Connect:Direct system does not have a default directory created for it in UNIX system services. The DLL files and other facilities related to SSL or TLS require the presence of a default UNIX directory.	Contact your z/OS system programmer.
The following message is received at startup: SITA166I Secure+ SSL or TLS initialization failed. rc=00000002, rs=GSK_KEYFILE_OPEN_ FAILED.	The Connect:Direct Secure Plus parameter file, with the SECURE.SSL.PATH.PREFIX initialization parameter, specifies a nonexistent key database, the key database has incorrect file permissions, OR the PASSWORD typed IS INCORRECT.	Correct the name specified in the initialization parameter or the Connect:Direct Secure Plus parameter file, the UNIX permissions, or the password.
<p>The following message is received at startup or when IBM Connect:Direct performs a certificate validation check and discovers a certificate that will soon expire: CSPA600W WARNING Cert: &cert for Node: &node expires: &date. The named certificate will expire on the specified date.</p> <p>Note: A message will not contain the node name if the certificate did not have a Connect:Direct Secure Plus parameter file record associated with it.</p>	The warning message will appear based on the validation check controlled by the following initialization parameters, CHECK.CERT.EXPIRE, CHECK.CERT.WARN.DAYS and CHECK.CERT.EXPIRE.TIME.	Take the appropriate action to generate or obtain a new certificate.
<p>The following message is received at startup or when IBM Connect:Direct performs a certificate validation check and discovers that a certificate has expired: CSPA601E ERROR Cert: &cert for Node: &node expired on: &date. The named certificate has expired on the specified date.</p> <p>Note: A message will not contain the node name if the certificate did not have one associated with it.</p>	The warning message will appear based on the validation check controlled by the following initialization parameters, CHECK.CERT.EXPIRE.	Take the appropriate action to generate or obtain a new certificate.

Problem	Possible Cause	Solution
<p>The following message is received at startup or when IBM Connect:Direct performs a certificate validation check and discovers a certificate it cannot validate: CSPA607W WARNING Cert: &cert for Node: &node does not exist. The Certificate Expiration Validation function has obtained a Certificate label for the Secure Parmfile however that certificate can not be retrieved.</p> <p>Note: A message will not contain the node name if the certificate did not have one associated with it.</p>	<p>The most likely cause of this is the certificate does not exist in the Key database or Key ring.</p>	<p>Ensure that the Certificate exist and that the Secure Parmfile entry specifies the correct label name. The label is case sensitive and must match exactly.</p>
<p>The following message is received when an SSL or TLS Process is run: SSL or TLS handshake failure, reason= GSK_ERROR_SOCKET_CLOSED.</p>	<p>The trading partners have not enabled a matching cipher suite.</p>	<p>Update the remote node record for the trading partner to enable a cipher suite recognized by the trading partner and resubmit the Process.</p>
<p>The following message is received: CSPA202E SSL handshake failure, reason=GSK_ERROR_BAD_CERTIFICATE.</p>	<p>The certificate is not valid on the system issuing GSK_ERROR_BAD_CERT. This error occurs if the certificate is not validated on any local trusted CA certificate.</p> <p>This error is common if you use self-signed certificates because the remote IBM Connect:Direct system does not have the CA certificate.</p>	<p>Verify that each trading partner can validate the certificates of other trading partners and resubmit the Process.</p> <p>Ensure that the remote node record for the trading partner has enabled the correct protocol.</p>
<p>The following error is received from the SNODE: CSPA202E SSL or TLS handshake failure, reason= GSK_ERROR_UNKNOWN_ERROR.</p>	<p>A conflict within the IBM System SSL toolkit occurred because a certificate being processed did not use version 3 of the toolkit.</p>	<p>Ensure that all certificates and CA certificates are using version 3.</p>
<p>Connect:Direct Secure Plus features are enabled in the Connect:Direct Secure Plus parameter file, but the statistics record indicates that these functions are disabled.</p>	<p>The IBM Connect:Direct network maps do not contain entries for the PNODE and SNODE.</p> <p>The node that you are connecting with is a V1 flow (such as LU0 or Netex). Connect:Direct Secure Plus is not supported for V1 flows because of reliance on XDR support.</p>	<p>Verify that the network map entries for both the PNODE and the SNODE exist, and use a V2 protocol such as LU6.2 ,TCP/IP, or UDT. Check for the existence of the extended statistics record for Session Begin (the SB record). This record is only created in V2 flows. The absence of this record indicates V1 flows were used.</p>

Problem	Possible Cause	Solution
Connect:Direct Secure Plus parameters specified from the copy statement cause the copy step to fail with message CSPA077E.	The node that you are connecting with is a V1 flow (such as LU0 or Netex). Connect:Direct Secure Plus is not supported for V1 flows because of reliance on XDR support.	Check for the existence of the extended statistics record for Session Begin (the SB record). This record is only created in V2 flows. The absence of this record indicates V1 flows were used.
An error occurs in ESTAE with a bad return code (RC=3) when running a Process with a remote node and the Process fails.	The value for Connect:Direct Secure Plus Export version is incorrect in the remote node definitions for one or both of the nodes. If one node is EXPORT and the other node is NOT EXPORT, the elliptic curves that enable you to create keys and generate Diffie-Hellman shared secrets are not correct.	Verify that the remote node definitions on both sites accurately state the Connect:Direct Secure Plus Export information.
Running a Process with a remote node fails with an authentication error.	Unique public/private key pairs are generated for the remote node record and the local node record is set to OVERRIDE=N.	Change the local node record to OVERRIDE=Y or do not use unique public/private key pairs in the remote node record.
The Save Active option is not selectable.	You can only use the Save Active function once each time you open the Connect:Direct Secure Plus parameter file.	Reopen the Connect:Direct Secure Plus parameter file to use the Save Active function or use the Save As function.
The text entry fields on the Create/Update panel of the Secure+ Admin Tool are not visible.	The CUA attributes in your ISPF profile are not set correctly.	Change the value for Normal Text entry in the CUA attributes of the ISPF profile to uscore in the Highlight column.
The Connect:Direct Secure Plus parameter, ENCRYPT.DATA specified from the copy statement causes the copy step to fail with an error message CSPA080E.	The algorithm name used in the COPY statement is not in the supported algorithm list for both nodes.	Verify that the algorithm name in the copy statement is in the supported algorithm list for both nodes.
A Process including a COPY statement with a SECURE parameter was submitted and failed. The following CSPA011E error message is displayed: Illegal attempt to override Connect:Direct Secure Plus parameters	You attempted to use the SECURE parameter in a COPY statement but did not specify OVERRIDE=Y in the remote node record to enable the security override feature.	Take one of the following actions: <ul style="list-style-type: none"> Remove the SECURE= parameter from the COPY statement and resubmit the Process. Change the OVERRIDE setting in the remote node record in the parameter file and make sure all other necessary protocol settings are specified. Resubmit the Process including the SECURE= parameter. See Override Settings in IBM Connect:Direct Processes .

Problem	Possible Cause	Solution
An SSL or TLS session was attempted with a IBM Connect:Direct system that does not implement SSL or TLS.	The trading partner does not have the protocol enabled.	Request that the trading partner configure its node for the correct protocol or disable Connect:Direct Secure Plus for the node.
Either the CSPA203E error message or the CSPA204E message is displayed: SSL or TLS send failure, rc=&RC, rsn=&RSN or SSL or TLS receive failure, rc=&RC, rsn=&RSN.	The client cannot validate the server's certificate.	Ensure that client authentication is turned on and certificate information is defined in the remote node record.
The following CSPA205E error message is displayed: SSL or TLS support requires the TCP/IP protocol.	One of the trading partners is not using TCP/IP for communications.	Determine which trading partner does not have TCP/IP enabled and change the configuration of that trading partner.
The following CSPA200E error message is displayed: Connect:Direct Secure Plus version mismatch.	You are attempting to use the SSL or TLS protocol to securely communicate with a trading partner that does not have the protocol enabled.	Change the configuration of the remote node record to enable the correct protocol.
The following CSPA206E error message is displayed: Remote certificate is invalid.	The root certificate was not found.	Check the parameter file configuration and ensure that the correct certificate is identified in the remote node record.
The following CSPA207E error message is displayed: Root certificate not found.	The remote certificate could not be validated.	Check the parameter file configuration and ensure the correct key database file is identified in the remote node record.
The following SITA1901 error message is displayed: Sec+ Init failed. Secure= No. Override=No.	The local node record has all Connect:Direct Secure Plus protocols disabled and has override set to no.	Either enable the appropriate protocol in the remote node record or enable override=yes in the local node record.

Problem	Possible Cause	Solution
A Process was submitted and failed. The following CSPA078E error message is displayed: Invalid specification of SECURE= on PROCESS statement. SECURE= cannot be specified in a non-IBM Connect:Direct Secure Plus environment or when the Remote Node record in the Connect:Direct Secure Plus Parmfile does not specify OVERRIDE=Y.	You attempted to use the SECURE parameter in a PROCESS statement but did not specify OVERRIDE=Y in the remote node record to enable the security override feature.	<p>Take one of the following actions:</p> <ul style="list-style-type: none"> Remove the SECURE= parameter from the PROCESS statement and resubmit the Process. Change the OVERRIDE setting in the remote node record in the parameter file and make sure all other necessary protocol settings are specified. Resubmit the Process including the SECURE= parameter. <p>See Override Settings in IBM Connect:Direct Processes.</p>
The submit within a Process failed with a reason code of 8. The following SCBI514E or SSUB267E error message is displayed: Equal sign required after SECURE keyword. The SECURE keyword in the PROCESS must be followed by an equal sign.	You attempted to use the SECURE parameter in a PROCESS statement but did not include an equal sign after the SECURE keyword.	Correct the PROCESS statement syntax by inserting an equal sign and resubmit the Process.
The submit within a Process failed with a reason code of 8. The following SCBI515E or SSUB268E error message is displayed: A parsing error occurred on the SECURE keyword when processing the SECURE keyword on the PROCESS statement.	You attempted to use the SECURE parameter in a PROCESS statement but the syntax was faulty.	Correct the PROCESS statement and resubmit the Process. For a complete description of the SECURE parameter and how to use it in the PROCESS statement, see the <i>IBM Connect:Direct Process Language Reference Guide</i> .
System initialization failed, and the following SITA196E error message is displayed: FIPS Mode Requested but SECURE.DSN parameter is not specified.	You specified the FIPS initialization parameter as YES, but you did not specify the SECURE.DSN parameter to enable Connect:Direct Secure Plus.	Update the initialization parameters and restart IBMConnect:Direct.
System initialization failed, and the following error message is displayed: Connect;Direct FIPS keyword requires z/OS release 1.11 or later.	Your current, active z/OS release level does not support FIPS mode for System SSL.	Either update the FIPS initialization parameter to NO or execute IBM Connect:Direct on the appropriate release level of z/OS.

Problem	Possible Cause	Solution
IBM Connect:Direct was terminated, and the following error message is displayed: Secure+ Severe FIPS Mode Error, &var1.	<p>During operation of a TLS FIPS mode request, a severe error occurred causing IBM Connect:Direct to terminate with a U4079 abend due to one of the following:</p> <ul style="list-style-type: none"> • KEY database (not FIPS-mode) • Random number generation failure • RSA or DSA keypair generation failure • gsk_perform_kat API failure 	Contact IBM Support or correct the error and restart IBM Connect:Direct.

Certificate Parameter Definitions

This topic describes the certificate parameter definitions for certificates created by the RACF application, GSKKYMANT utility, CA-ACF2 application, and CA-ACF2 application.

RACF Application Certificate Parameter Definitions

To avoid some problems associated with CA-signed and self-signed certificates, refer to the following information about certificate parameter definitions required to use Connect:Direct Secure Plus for z/OS. Minimum parameter definitions for certificates generated with the RACF, gskkyman, CA-ACF2, and CA-Top Secret security applications are provided.

If you plan to use FIPS mode, see [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#) for more information about System SSL and FIPS mode.

You may also want to record the parameter definitions you configure for certificates on the worksheets provided for the local and remote node records in [Configuration Worksheets](#).

This table describes the minimum parameter definitions required for Connect:Direct Secure Plus for z/OS. When two parameters are listed in the same row, the first parameter name is used when you create a certificate and the second parameter name is its equivalent, which is used when you display information about the certificate. Consult the RACF documentation for detailed information about all the certificate parameters and commands.

RACF Parameter	Description	Value Used for Connect:Direct Secure Plus
User ID	Security ID used to start the IBM Connect:Direct Job or Started Task.	RACF-defined ID
Label	Certificate label. LABEL keywords are case and blank sensitive; therefore, the values specified for these keywords must be exact.	<p>Information that identifies the certificate, for example, CD Secure Plus</p> <p>Note: Specify the exact value in the Certificate Label field in the Local Node record of the Connect:Direct Secure Plus parameter.</p> <p>Note: Use the default certificate defined in the key store. Leave the certificate label field in the local record as blank.</p>
Status	Status of the certificate.	<p>Status=TRUST</p> <p>All certificates used by Connect:Direct Secure Plus for z/OS must be Trusted.</p>

RACF Parameter	Description	Value Used for Connect:Direct Secure Plus
NOTBEFORE Start Date	Specifies the local date and time from which the certificate is valid.	Must be a valid date and time
NOTAFTER End Date	Specifies the local date and time after which the certificate is no longer valid. All certificates used in the SSL/TLS handshake, including issuer certificates, must not be expired.	Must be a valid date and time
Key Usage	Facilitates identification and key exchange during SSL/TLS security handshakes.	<p>HANDSHAKE (Required): Indicates that digital signature and key encipherment are enabled.</p> <p>DOCSIGN (Optional): Indicates that non-repudiation is enabled.</p> <p>DATAENCRYPT (Optional): Indicates that data encipherment is enabled.</p> <p>CERTSIGN: Indicates the certificate can sign other digital certificates and CRLs.</p> <p>Note: Do not specify CERTSIGN. Only Certificate Authority (Issuer) certificates should have keyCertSign and cRLSign indicators.</p>
X.509 Subject's Distinguished Name Issuer's Name	<p>Specifies the distinguished name of the issuer that issued or signed a certificate. The name identifies the trusted certificate of the issuer or CA that signed the server certificate. The name identifies the trusted certificate of the issuer or CA that signed the server certificate. The CA or entity certificate with that name must be available within the key database or Keyring. The Issuer Name keywords are case and blank sensitive.</p> <p>Note: Self-signed certificates display the same information in the Issuer Name and Subject Name parameters.</p>	<p>The following fields, which must be enclosed in single quotes, are attributes of the Issuer's Name parameter and the Subject's Name parameter:</p> <p>CN=Common Name of the certificate in single quotes, for example, 'RACF SELF SIGN COMMON'</p> <p>T='Title of person creating certificate'</p> <p>OU='Organizational Unit associated with the person creating the certificate'</p> <p>O='Organization for which the certificate is being created'</p> <p>L='Locality (city) of the entity for which the certificate is created'</p> <p>SP='State/Province of the locality'</p> <p>C='Country of the locality'</p>
X.509 Subject's Distinguished Name Subject's Name	Specifies the certificate's subject distinguished name. It identifies the certificate. This name can identify certificates that may have issued or signed other certificates and can match to other certificates Issuer's Name.	

RACF Parameter	Description	Value Used for Connect:Direct Secure Plus
Private Key Size	Specifies the size of the private key expressed in decimal bits. Key size of 1024 provides a secure encryption. A larger size provides a more secure encryption but requires more CPU to encrypt.	
Private Key Type	Specifies how the private key should be stored for future use. Type can be none, non-ICSF, or ICSF. If Type= none, the certificate does not have a private key.	If ICSF is specified, see IBM Connect:Direct Access to System Resources for SSL or TLS for requirements.
Ring Name	Specifies the name of the keyring that a certificate is connected with.	If you use a key ring, the exact value in this field must be specified in the Certificate Pathname field for the Local Node record in the Connect:Direct Secure Plus parameter file.
Usage	Specifies how this certificate should be used in a keyring for the USERID of the person submitting a batch job or signed on to TSO.	PERSONAL
Default	Specifies that the certificate is the default certificate. Only one certificate can be the default certificate. Define the end-user server certificate of the local IBM Connect:Direct node as the default.	YES

GSSKYPAN Utility Certificate Parameter Definitions

This table describes the minimum parameter definitions required for Connect:Direct Secure Plus for z/OS. Consult the GSSKYPAN documentation for detailed information about all the certificate parameters and commands. If you plan to use FIPS mode, see [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#) for more information about System SSL and FIPS mode.

GSSKYPAN Parameter	Description	Value Required for Connect:Direct Secure Plus Option
Label	Certificate label. LABEL keywords are case and blank sensitive; therefore, the values specified for these keywords must be exact.	Information to identify the certificate, for example, CD Secure Plus Note: Specify the exact Label value in the Certificate Label field in the local node record of the Connect:Direct Secure Plus parameter file. Note: Use the default certificate defined in the key store. Leave the certificate label field in the local record as blank. Ensure that a default certificate exist in the keystore.
Version	X.509 certificates with version number 3 are supported.	3
Trusted	Specifies the certificate status.	Yes

GSKKMAN Parameter	Description	Value Required for Connect:Direct Secure Plus Option
Effective Date	Specifies the local date and time from which the certificate is valid.	Must be a valid date and time
Expiration Date	Specifies the local date and time after which the certificate is no longer valid. All certificates used in the SSL/TLS handshake, including issuer certificates, must not be expired.	Must be a valid date and time
keyUsage	Facilitates identification and key exchange during SSL/TLS security handshakes.	Digital Signature (Required) Non-repudiation Key encipherment Data encipherment
Issuer Name	Specifies the distinguished name of the Issuer that issued or signed a certificate. The name identifies the trusted certificate of the issuer or CA that signed the server certificate. The CA or entity certificate with that name must be available within the key database or keyring. The Issuer Name keywords are case and blank sensitive. Self-signed certificates have the same Issuer name and Subject name.	
Certificate Subject Name	Specifies the certificate's subject distinguished name. It identifies the certificate. This name can identify certificates that may have issued or signed other certificates and can match to other certificates Issuer's Name.	The following fields are attributes of the Certificate Subject Name parameter: CN=Common Name of the certificate in single quotes, for example, 'RACF SELF SIGN COMMON' T='Title of person creating certificate' OU='Organizational Unit associated with the person creating the certificate' O='Organization for which the certificate is being created' L='Locality (city) of the entity for which the certificate is created' SP='State/Province of the locality' C='Country of the locality'
Public Key Algorithm	Specifies the algorithm used to encrypt data.	

GSKKYMANT Parameter	Description	Value Required for Connect:Direct Secure Plus Option
Public Key Size	Specifies the size of the public key expressed in decimal bits. Key size of 1024 provides a secure encryption. A larger size provides a more secure encryption but requires more CPU to encrypt.	
Key database password	Specifies the password used when you created a key database file.	When you specify a gskkyman key database file name in the Certificate Pathname field for the local node record, you must specify the key database password in the Certificate Pathname Pass Phrase field.
Default	Specifies that the certificate is the default certificate. Only one certificate can be the default certificate. Define the end-user server certificate of the local IBM Connect:Direct node as the default.	YES

CA-ACF2 Application Certificate Parameter Definitions

This table describes the minimum parameter definitions required for Connect:Direct Secure Plus for z/OS. Consult the CA-ACF2 documentation for detailed information about all the certificate parameters and commands.

CA-ACF2 Parameter	Description	Value Used by Connect:Direct Secure Plus Option
ACID	Security ID used to start the IBM Connect:Direct Job or Started Task.	CA-ACF2 defined ID
Label	Certificate label. LABEL keywords are case and blank sensitive; therefore, the values specified for these keywords must be exact.	Information that identifies the certificate, for example, CD Secure Plus Note: Specify the exact value in the Certificate Label field in the Local Node record of the Connect:Direct Secure Plus parameter file.

CA-ACF2 Parameter	Description	Value Used by Connect:Direct Secure Plus Option
Subjsdsn	Specifies the subject's distinguished name. It identifies the certificate. This name can identify certificates that may have issued or signed other certificates and can match to other certificates Issuer's Name.	<p>The following fields, which must be enclosed in single quotes, are attributes of the Issuer's Name parameter and the Subject's Name parameter:</p> <p>CN=Common Name of the certificate in single quotes, for example, 'RACF SELF SIGN COMMON'</p> <p>T='Title of person creating certificate'</p> <p>OU='Organizational Unit associated with the person creating the certificate'</p> <p>O='Organization for which the certificate is being created'</p> <p>L='Locality (city) of the entity for which the certificate is created'</p> <p>SP='State/Province of the locality'</p> <p>C='Country of the locality'</p>
Size	Specifies the size of the private encryption key in bits.	
Active	Specifies the local date and time from which the certificate is valid.	Must be a valid date and time
Expire	Specifies the local date and time after which the certificate is no longer valid. All certificates used in the SSL/TLS handshake, including issuer certificates, must not be expired.	Must be a valid date and time
Keyusage	KeyUsage certificate extension, of which one or more of the following values might be coded.	<p>HANDSHAKE (Required): Indicates that digital signature and key encipherment are enabled.</p> <p>DOCSIGN (Optional): Indicates that non-repudiation is enabled.</p> <p>DATAENCRYPT (Optional): Enables the certificate to be used to.</p> <p>CERTSIGN: Indicates the certificate can sign other digital certificates and CRLs.</p> <p>Note: Do not specify CERTSIGN. Only Certificate Authority (Issuer) certificates should have keyCertSign and cRLSign indicators.</p>
KEYRING	Specifies the record key of a KEYRING record to which the certificate is associated.	<p>If you use a keyring, the value in this field must be specified in the Certificate Label field for the Local Node record in the Connect:Direct Secure Plus parameter file.</p>

CA-ACF2 Parameter	Description	Value Used by Connect:Direct Secure Plus Option
RINGNAME	Specifies the ring name of a KEYRING record to which the certificate information is associated.	If you use a keyring, the value in this field must be specified in the Certificate Pathname field for the Local Node record in the Connect:Direct Secure Plus parameter file.
USAGE	Specifies how this certificate should be used in a keyring for the USERID of the person submitting a batch job or signed on to TSO.	PERSONAL
DEFAULT	Specifies that the certificate is the default certificate. Only one certificate can be the default certificate. Define the end-user server certificate of the local IBM Connect:Direct node as the default.	YES

CA-Top Secret Application Certificate Parameter Definitions

This table describes the minimum parameter definitions required for Connect:Direct Secure Plus for z/OS. Consult the CA-ACF2 documentation for detailed information about all the certificate parameters and commands.

CA-Top Secret Parameter	Description	Value Used for Connect:Direct Secure Plus Option
SUBJECTDSN	Specifies the subject's distinguished name. It identifies the certificate. This name can identify certificates that may have issued or signed other certificates and can match to other certificates Issuer's Name.	<p>The following fields, which must be enclosed in single quotes, are attributes of the Issuer's Name parameter and the Subject's Name parameter:</p> <p>CN='Common Name of the certificate in single quotes,' for example, 'RACF SELF SIGN COMMON'</p> <p>T='Title of person creating certificate'</p> <p>OU='Organizational Unit associated with the person creating the certificate'</p> <p>O='Organization for which the certificate is being created'</p> <p>L='Locality (city) of the entity for which the certificate is created'</p> <p>SP='State/Province of the locality'</p> <p>C='Country of the locality'</p> <p>UID='userid'</p>
UID	Security ID used to start the IBM Connect:Direct Job or Started Task.	CA-Top Secret defined ID
NBDATE/NBTIME	Specifies the local date and time from which the certificate is valid.	Must be a valid date and time

CA-Top Secret Parameter	Description	Value Used for Connect:Direct Secure Plus Option
NADATE/NATIME	Specifies the local date and time after which the certificate is no longer valid. All certificates used in the SSL/TLS handshake, including issuer certificates, must not be expired.	Must be a valid date and time
KEYSIZE	Specifies the size of the private encryption key in bits.	
LABLCERT	Certificate label. LABEL keywords are case and blank sensitive; therefore, the values specified for these keywords must be exact. This parameter is specified when you associate a certificate with an ACID.	Information to identify the certificate, for example, CD Secure Plus Note: Specify the exact value in the Certificate Label field in the Local Node record of the Connect:Direct Secure Plus parameter file.
ICSF	If Private Key type is ICSF, the private key is stored in the ICSF PKDS (public key data set). Access to the private key then requires that the ICSF application be executing and IBM Connect:Direct have access authority to the ICSF application	If ICSF is specified, see IBM Connect:Direct Access to System Resources for SSL or TLS for requirements.
TRUST NOTRUST	Specifies the status of the certificate when you associate a certificate with an ACID.	TRUST
KEYRING	Specifies the key ring being added to the user's ACID.	If you use a keyring, the value in this field must be specified in the Certificate Label field for the Local Node record in the Connect:Direct Secure Plus parameter file.
LABLRING	Specifies the label to be associated with the keyring being added to the user, which is used as the identifier of the digital certificate.	If you use a keyring, the value in this field must be specified in the Certificate Pathname field for the Local Node record in the Connect:Direct Secure Plus parameter file.
DEFAULT	Specifies how this certificate should be used in a keyring for the USERID of the person submitting a batch job or signed on to TSO.	PERSONAL
USAGE	Specifies that the certificate is the default certificate. Only one certificate can be the default certificate. Define the end-user server certificate of the local IBM Connect:Direct node as the default.	YES

Configuration Worksheets

Use the worksheets in this topic to record the configuration information for Connect:Direct Secure Plus for z/OS.

Local Node Security Feature Definition Worksheet

Record the security feature definitions for the Connect:Direct Secure Plus local node record on this worksheet. Refer to this worksheet as you configure the local node record. If you plan to use FIPS mode, see [“Planning for System SSL in FIPS Mode” on page 751](#) and [z/OS V1R11.0 Cryptographic Services System Sockets Layer Programming SC24-5901-08](#).

Local Node Name: _____	
TLS protocol enabled: Note: If System SSL is in FIPS mode, TLS is the only supported protocol. See “Planning for System SSL in FIPS Mode” on page 751 .	Yes _____ No _____ TLS1.0 ____ TLS1.1 ____ TLS1.2 ____
SSL protocol enabled:	Yes _____ No _____
Configured Security Functions	
Override enabled:	Yes _____
Encryption enabled:	Yes _____ No _____
Authorization Timeout: Set the value equal to or greater than the value set for the IBM Connect:Direct TCP.TIMER initialization parameter.	_____ (Numeric value equal to or greater than 0 seconds)
Certificate Label (label specified when the certificate was generated using one of the security applications; may be called LABLCERT): Note: If System SSL is in FIPS Mode, the Certificate Label has FIPS requirements. See “Planning for System SSL in FIPS Mode” on page 751 .	_____ _____ Valid only for SSL or TLS
Certificate Pathname: key database or key ring Note: If System SSL is in FIPS Mode, the Certificate Pathname has FIPS requirements. See “Planning for System SSL in FIPS Mode” on page 751 .	_____ _____ Valid only for SSL or TLS
Password: Valid only for certificates created in a gskkyman database; leave blank for key rings.	_____

Local Node Name: _____	
Cipher Suite(s) to Enable: Applies only to SSL and TLS Note: If System SSL is in FIPS mode, only certain ciphers are valid. See the <i>IBM Connect:Direct for z/OS Release Notes</i> for a list of valid FIPS-mode ciphers.	_____ ---
Enable External Authentication: Applies only to SSL and TLS	Yes _____ No _____

Remote Node Security Feature Definition Worksheet

Record the security feature definitions for a remote node record on this worksheet. Make a copy of this worksheet for each remote node defined in the Connect:Direct Secure Plus parameter file that you are configuring for Connect:Direct Secure Plus operations. Refer to this worksheet when you configure a remote node record.

Remote Node Name:	_____ ---
Security Options	
TLS protocol enabled:	Yes _____ No _____ TLS1.0 ____ TLS1.1 ____ TLS1.2 ____
SSL protocol enabled:	Yes _____ No _____
Enable Override: When override is enabled in a remote node record, values in the PROCESS statement override values in the remote node record that uses either protocol.	Yes _____ No _____ Default to local node _____
Encryption enabled:	Yes _____ No _____
Enable External Authentication: Valid only for SSL or TLS	Yes _____ No _____ Default to local node _____
TLS or SSL Protocol Functions	
If you enabled the TLS or SSL protocol and you did not define this information in the local node record, set one or more of the following functions:	
Certificate Label: Label specified when the certificate was generated using one of the security applications; may be called LABLCERT.	_____ --- You can type an asterisk (*) to default to the local node record.
Cipher Suite(s) Enabled:	_____ ---
Ask the trading partner which cipher suites are enabled. Circle all applicable cipher suites.	

<ul style="list-style-type: none"> • SSL_RSA_WITH_AES_128_SHA • SSL_RSA_WITH_AES_256_SHA • SSL_RSA_WITH_3DES_EDE_CBC_SHA • SSL_RSA_WITH_DES_CBC_SHA • SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5 • SSL_RSA_WITH_RC4_40_SHA • SSL_RSA_WITH_RC4_128_MD5 • SSL_RSA_EXPORT_WITH_RC4_40_MD5 • SSL_RSA_WITH_NULL_SHA • SSL_RSA_WITH_NULL_MD5 	<ul style="list-style-type: none"> • TLS_RSA_WITH_AES_128_SHA • TLS_RSA_WITH_AES_256_SHA • TLS_(or TLS_) RSA_WITH_3DES_EDE_CBC_SHA • TLS_RSA_WITH_DES_CBC_SHA • TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 • TLS_RSA_WITH_RC4_40_SHA • TLS_RSA_WITH_RC4_128_MD5 • TLS_RSA_EXPORT_WITH_RC4_40_MD5 • TLS_RSA_WITH_NULL_SHA • TLS_RSA_WITH_NULL_MD5 <p>Note: For a complete list of ciphers see, “Secure Cipher Suite” on page 753.</p>
Certificate Pathname key database or key ring	<p>----- --- You can type an asterisk (*) to default to the local node record.</p>
To add a second level of security by enabling Client Authentication, set the following two options:	
Enable Client Authentication:	Yes _____ No _____
If client authentication is enabled, specify the certificate common name of the local node certificate in the Client Auth. Compare field.	<p>----- ---</p>

.EASERVER Node Security Feature Definition Worksheet

Use the following worksheet to record information to configure the remote node record for .EASERVER node. Refer to this worksheet when you configure the .EASERVER remote node record.

Remote Node Name:	.EASERVER (Required)
TLS protocol enabled:	Yes _____ No _____
SSL protocol enabled	Yes _____ No _____
Note: You must enable either SSL or TLS to communicate with the External Authentication server.	
External Auth Server Def	<p>----- - Name of the certificate validation definition configured on the External Authentication Server that defines how to validate certificates. This parameter is case sensitive.</p>
External Auth Server Address	<p>----- - IP address of server for the External Authentication Server application</p>

Remote Node Name:	.EASERVER (Required)
TLS protocol enabled:	Yes _____ No _____
External Auth Server Port	<p>-----</p> <p>--</p> <p>Number of the port to use to connect to the External Authentication Server</p>
Client Authentication enabled:	Yes _____
Client Authentication Common Name: If client authentication is enabled, specify the certificate common name of the local node certificate in the Client Auth. Compare field.	<p>-----</p> <p>–</p>
Certificate Label:	<p>-----</p> <p>--</p> <p>You can type an asterisk (*) in the Certificate Label field to default to the local node record.</p>
Certificate Pathname key database or key ring	<p>-----</p> <p>–</p> <p>You can type an asterisk (*) in the Certificate Pathname field to default to the local node record.</p>

.CLIENT Node Security Feature Definition Worksheet

Record the security feature definitions for a remote node record named .CLIENT that you create to allow secure connections. Refer to this worksheet when you configure the .CLIENT node record.

Remote Node Name:	.CLIENT (Required)
Note: The node name must be defined as .CLIENT to allow secure connections.	
Security Options	
Autoupd enabled:	Yes _____ No _____
TLS protocol enabled:	Yes _____ No _____
SSL protocol enabled:	Yes _____ No _____
Enable Override:	Yes _____ No _____
ISPF IUI and Batch Interface Options	
ISPF IUI protocol defined as SNA:	Yes _____ No _____
Batch interface protocol defined as SNA	Yes _____ No _____

Chapter 10. Quick Reference Guide

Notational Conventions

The following notational conventions are used throughout the IBM Connect:Direct for z/OS documentation.

Syntax or Symbol	Description
Bold letters	Boldface letters indicate required statements, labels, commands, or parameters.
Lowercase letters	Words in lowercase letters require substitution by the user.
UPPERCASE and lowercase letters	Uppercase letters in a statement, command, or parameter indicate an alternative to typing the entire command.
Underlined letters	Underlined letters indicate default values for parameters and subparameters.
	A vertical bar symbolizes “or.”
[]	Brackets indicate optional information.
,	Commas separate items within a list or set off the order of values specified as positional parameters. A comma must be used to indicate omission of a positional parameter.
=	Characters to the left of equal signs are keywords; characters to the right of equal signs are parameters.
()	Parentheses enclose lists and associate a group of values.
*	Asterisks indicate a generic specification of parameters. With generics, users can request information by specifying an asterisk (*) or a character string plus an asterisk (nnn*).
' '	Single quotation marks enclose special characters.
" "	Double quotation marks enclose symbolics within a quoted string.
\\	Bracketing backslashes indicate special processing of a character string. They are used to continue a string containing special characters across multiple records or to ensure that quotation marks within the string are maintained.
	A double bar is used to concatenate character strings. A blank space is required before and after the double bar.
-	A hyphen is used to continue a statement across multiple records. A blank space is required before and after the hyphen.

IBM Connect:Direct Process Statements

A Process is a series of statements and parameters that perform data movement and manipulation activities.

These include:

- moving files between different IBM Connect:Direct servers
- running jobs, programs, and commands on the IBM Connect:Direct server
- starting other Processes
- monitoring and controlling Processes

- handling processing errors

Process Statement

Label	Statement	Parameters
Process name (required)	PROCCess	SNODE= secondary node name SNODE=TCPNAME=tcpvalue;port SNODE=UDT33NAM = udtvalue;port
		PNODE = primary node name %PNODE
		PNODEID = (id [,pswd] [,newpswd])
		SNODEID = (id [,pswd] [,newpswd])
		PACCT = 'pnode accounting data'
		SACCT = 'snode accounting data'
		CLASS = n
		CRC = (OFF ON)
		DEBUG = trace bits
		FASP=NO SSP
		FASP.BANDWIDTH=nnn nM nG (Default <BANDWIDTH_FROM_LICENSE>)
		FASP.FILESIZE.THRESHOLD=nnn nM nG
		FASP.POLICY=FAIR HIGH LOW FIXED
		HOLD = Yes <u>No</u> Call
		MAXDELAY = [<u>Unlimited</u> Queued hh:mm:ss 0]
		NOTIFY = %USER userid
		PLEXCLASS = (pnode class, snode class)
		PRTY = n
		REQUEUE = Yes <u>No</u>
		RETAIN = Yes <u>No</u> Initial
		SECURE=OFF SSL TLS10 TLS11 TLS12 or SECURE=ENCRYPT.DATA=Y N or SECURE = (OFF SSL TLS10 TLS11 TLS12 ENCRYPT.DATA=Y N) or SECURE = (OFF SSL TLS10 TLS11 TLS12 <cipher_suite> (cipher_suite_list),ENCRYPT.DATA=Y N)
		STARTT = ([date day][,hh:mm:ssxm])
		&symbolicName 1 = variable string 1
		&symbolicName 2 = variable string 2
.		

Label	Statement	Parameters
.		
.		
&symbolicN amen = variable string n		

Note: If a variable contains blanks, you must enclose the string in quotation marks.

Intrinsic Variables

You can use the following intrinsic symbolic variables in a Process statement to substitute user-defined values when the Process is executed. This flexibility lets you use the same Process for multiple applications when these values change.

Value	Description
%DD2DSN	Specifies an allocated DD statement, which references a DSN to be passed to a Process being submitted. The format is %DD2DSN(ddname), where ddname refers to an allocated DD in the address space of the JCL submitting the Process. The DSN, including the member if specified, is substituted for this variable. The ddname may contain 1-8 characters.
%JDATE	Specifies the date the Process was submitted in Julian format. The variable is resolved as the submission date of the Process in the format yyyyddd. Among other uses, the value returned is suitable for constructing a file name on the node receiving the file. The value of the variable is resolved at Process submit time. The value will correspond to the date on which the Process was submitted, regardless of when or how many times the Process is actually executed.
%JOBID	Specifies the job number.
%JOBNM	Specifies the job name.
%JUSER	Specifies a variable that resolves to the USERID of the submitted job.
%MAXRC	Highest return code (CCOD) of all Process steps. Use to test the highest RC with the Modal logic.
%MSGID	The message of the previous step. Use to test for a specific message id produced by the previous step.
%NUM1	Specifies the submission time of the Process in minutes, seconds, and fraction of seconds in the format mmssth.
%NUM2	Specifies the submitted time of a Process as the low order 4 bits of the milliseconds of the time expressed as 1 hex digit (a value from 0 through 15 expressed as 0 through F).
%PNODE	Specifies the PNODE name where the submit occurs.
%PRAND	Use this parameter to generate a pseudo-random number (6 hex digits) that can be used to construct a unique data set name.

Value	Description
%SUBDATE	Specifies the date the Process was submitted in Gregorian format. The variable is resolved as the submission date of the Process in the format cyymmdd where c is the century indicator and is set to 0 for year 19yy or 1 for year 20yy. The value returned can be used to create a file name on the node receiving the file.
%SUBDATE1	Use this parameter to substitute the submitted date in the yyyyymmdd date format.
%SUBDATE2	Use this parameter to substitute the submitted date in the yyyyddmm date format.
%SUBDATE3	Use this parameter to substitute the submitted date in the mmdyyy date format.
%SUBDATE4	Use this parameter to substitute the submitted date in the ddmmyyyy date format.
%SUBTIME	Specifies the time the process was submitted. The variable is resolved as the submission time of the process in the format hhmmss. The return value can be used to create a file name on the node receiving the file. The value of the variable is resolved at Process submit time. The value will correspond to the time at which the Process was submitted, regardless of when or how many times the Process is actually executed
%USER	Specifies a variable that resolves to the user submitting the Process.

COPY Statement

Label	Statement	Parameters
[optional]	COPY	FROM (DSN = data set name/password FILE = file name
		<u>P</u> NODE SNODE
		SYSOPTS = "UNIQUE=YES" "DBCS = (tablename, so, si, PAD PAD = pc, LOGIC=A B (B,RC))" "codepage =(from code set, to Unicode code set)" "parameter1 [parameter2,...]" "datatype = <u>text</u> binary" † "xlate = <u>no</u> yes" † "strip.blanks = yes <u>no</u> " † "permiss = nnn" † DATATYPE = text binary XLATE = <u>no</u> yes STRIP.BLANKS = <u>no</u> yes PERMISS = nnn ZIIP=ON OFF ZEDC=OFF ON
		DISP = ([OLD <u>SHR</u>], [KEEP DELETE], [KEEP DELETE])
		RESGDG = <u>Sub</u> Run

Label	Statement	Parameters
		LABEL = ([file sequence number] ,[<u>SL</u> AL BLP LTM NL] ,[PASSWORD NOPWREAD] ,[IN OUT] ,[RETPD = nnnn EXPDT = [yyddd yyyy/ddd]])
		MSVGP = MS group name
		UNIT = ([unit address device-type group name],[unit-count P])
		VOL = ([PRIVATE],[RETAIN] ,[volume-sequence-no] ,[volume-count] ,[SER = (serial-no[,serial-no,...])]) ([SER = (serial-no, [serial-no,...]) REF = dsn])
		ALIAS = <u>Y</u> N
		EXCLUDE = (generic member (startrange/stoprange) list)
		PDS.DIRECTORY = <u>Y</u> N
		<u>REPLACE</u> NOREPLACE
		SELECT=(member generic (*) (member, [new-name],[NR R]) (generic,, [NR R]) (start-range/stop-range,, [NR R]) list)
		BUFND=number
		IOEXIT=exit-name (exit-name [,parameter,...])
		DATAEXIT=exit-name (exit-name [,parameter,...])
		“DBCS = (tablename, so, si, PAD PAD = pc, LOGIC=A B (B,RC))” “codepage =(from code set, to Unicode code set)” “parameter1 [parameter2,...]” “datatype = <u>text</u> binary” “xlate = <u>no</u> yes” † “strip.blanks = yes <u>no</u> ” † “precomp = yes <u>no</u> ” DATATYPE = text binary † XLATE = <u>no</u> yes STRIP.BLANKS = <u>no</u> yes
)
		TO (DSN = data set name/password FILE = file name
		<u>PNODE</u> <u>SNODE</u>
		TYPE = typekey

Label	Statement	Parameters
		DCB = ([model file name] [,BLKSIZE = number of bytes] [,DEN = 0 <u>1</u> 2 3 4] [,NCP = 0-255] [,DSORG = DA PO PS VSAM] [,KEYLEN = number of bytes] [,LIMCT = number of blocks or tracks] [,LRECL = number of bytes] [,OPTCD = [W Q Z] [,RECFM = record format] [,RKP = first-byte-of-record-key] [,TRTCH = C E T ET COMP NOCOMP])
		DISP = ([NEW OLD MOD RPL SHR] [,KEEP <u>CATLG</u> [,KEEP <u>CATLG</u> DELETE])
		AVGREC = U K M
		DATACLAS = data-class-name
		DSNTYPE = PDS LIBRARY BASIC LARGE EXTPREF EXTREQ
		KEYLEN = bytes
		KEYOFF = offset-to-key
		LIKE = model-data-set-name
		LRECL = bytes
		MGMTCLAS = management-class-name
		RECORD = KS ES RR LS
		SECMODEL = (profile-name [,GENERIC])
		STORCLAS = storage-class-name
		LABEL = ([file sequence number] [, <u>SL</u> AL BLP LTM NL] [,PASSWORD NOPWREAD] [,IN OUT] [,RETPD = nnnn EXPDT = [yyddd yyyy/ddd])
		MSVGP = MS group name
		SPACE = (CYL TRK <u>blk</u> , (prim, [sec], [dir]) [RLSE], [CONTIG],[ROUND]) (avg- rec-len,(primary-rcds, [secondary-rcds] , [dir]))
		UNIT = [(unit address device-type group name],[unit-count P])
		VOL = ([PRIVATE], [RETAIN], [volume-sequence-no], [volume-count] [,volume-sequence-no] [,volume-count] [,SER = (serial-no[,serial-no,...])]) ([SER = (serial-no, [serial-no,...]) REF = dsn])
		BUFND = number

Label	Statement	Parameters
		IOEXIT = exit-name (exit-name [,parameter,...])
		DATAEXIT = exit-name (exit-name [,parameter,...])
		SYSOPTS = “UNIQUE=YES” “DBCS = (tablename, so, si, PAD PAD = pc, LOGIC=A B (B,RC))” “codepage =(from code set, to Unicode code set)” “parameter1 [parameter2,...]” “datatype = <u>text</u> binary” † “xlite = <u>no</u> yes” † “strip.blanks = yes <u>no</u> ” † “permiss = nnn” † DATATYPE = text binary XLATE = <u>no</u> yes STRIP.BLANKS = <u>no</u> yes PERMISS = nnn ZIIP=ON OFF ZEDC=OFF ON
)
		CKPT = nK nM
		COMPRESS [[PRIMEchar = X'40' X'xx' C'c'] EXTended EXTended=(TYPE=(NOZIIP ZIIP , NOZEDC ZEDC))]
		ZFBA = 2
		FASP=NO SSP
		FASP.BANDWIDTH=nnn nM nG (Default <BANDWIDTH_FROM_LICENSE>)
		FASP.FILESIZE.THRESHOLD=nnn nM nG
		FASP.POLICY=FAIR HIGH LOW FIXED
† = HFS File Only		

RUN JOB Statement

Label	Statement	Parameters
[optional]	RUN JOB	(DSN = dsn[(member)])
		<u>PNODE</u> SNODE

RUN TASK Statement

Label	Statement	Parameters
[optional]	RUN TASK	(PGM = program name, PARM = (parameter, [,parameter, ...]))
		SYSOPTS = “parameter [,parameter, ...]”
		<u>PNODE</u> SNODE

Run Task Utility Programs and Parameters

Program	Parameters
DGADTIFY	CL4'GOOD' CL4'FAIL' file name
DGADTFY2	CL4'GOOD' CL4'FAIL' file name user ID user ID list
DGADTDYN	ALLOC UNALLOC CONCAT DECONCAT LOCATE
DGADTAMS	sysprint output parameters control statement parameters
DGADTSUB	JCL source subsequent parameters
DGADGSUB	%SRCDSN %DSTDSN %PROCNAME %PROCNUM

SUBMIT Statement

Label	Statement	Parameters
[optional]	SUBmit	DSN = dsn[(member)]
		CASE = Yes <u>No</u>
		CLASS = n
		DEBUG = trace bits
		FASP=NO SSP
		FASP.BANDWIDTH=nnn nM nG
		FASP.FILESIZE.THRESHOLD=nnn nM nG
		FASP.POLICY=FAIR HIGH LOW FIXED
		HOLD = Yes No Call
		NEWNAME = newname
		NOTIFY = %USER userid
		PACCT = 'pnode accounting data'
		PLEXCLASS = (pnode class, snode class)
		PNODEID = (id[,pswd][,newpswd])
		PRTY = n
		REQUEUE = Yes <u>No</u>
		RETAIN = Yes <u>No</u> Initial
		SACCT = 'snode accounting data'
		SNODE = secondary node SNODE=TCPNAME = tcpvalue;port
		SNODEID = (id [,pswd] [,newpswd])
		STARTT = ([date day] [,hh:mm:ssxm])
		SUBNODE = <u>PNODE</u> SNODE

Note: If a variable string contains blanks, you must enclose the string in quotation marks.

SYMBOL Statement

Label	Statement	Parameter
[optional]	SYMBOL	&symbolic_name = variable-string

Modal Statements

Label	Statement	Parameters
[optional]	IF	(label condition nn)
	THEN	(process steps)
[no label]	ELSE	(alternative process steps)
[no label]	EIF	[none]

Label	Statement	Parameters
[optional]	GOTO	statement label
[no label]	EXIT	[none]

IBM Connect:Direct Commands

Use IBM Connect:Direct commands to submit and manipulate Processes stored in the Transmission Control Queue (TCQ). For example, you can select, delete, and suspend Processes by using IBM Connect:Direct commands.

Modify Command

Label	Command	Parameters
[optional]	MODIFY †	BITS.OFF = X'nnnnnnnn'
	†	BITS.ON = X'nnnnnnnn'
		CLOSE = ddname
		DDNAME = (ddname,nn)
	†	DEBUG = trace bits
		DYN = 'dynamic allocation string'
		INITPARMS
		MODDIR.TRACE = YES
		NODE.TRACE.BITSOFF=(node name, X'nnnnnnnn')
		NODE.TRACE.BITSON=(node name, X'nnnnnnnn')
	†	NODE.TRACE.ON = (node name, debug bits)
	†	NODE.TRACE.OFF = node name
		SESSIONS = Quiesce Resume (WHERE(NODE=node name))
		WHERE(SERVER = server name)
		ZIIP = NONE EXTCOMP SSLTLS ALL PROJECT
		ZEDC = ON OFF PREFERRED
		ZFBA = YES NO

† The BITS.ON, BITS.OFF, DEBUG, and NODE.TRACE parameters affect debug traces. See [Trace Types](#) for a list of trace bits.

Network Map Commands

Command	Parameters
\$\$INSERT ADJACENT.NODE =	((nodename, communications name channel-range-start-addr, remote library name IP address or Alias addr-count, session type security node type, data direction restriction) PARSESS = (max,default) ZFBA=(addr1,addr2) - COMPRESS.EXT=ALLOW DISALLOW FORCE COMPRESS.STD=ALLOW DISALLOW FORCE COMPRESS.STD.PRIMECHAR=C'x' X'xx' SOURCEIP=IP address - SESS.SNODE.MAX = (1–255) LDNS=hostname LOGMODE=logmode entry name APPLIDS=(vtam applid1 [,vtam applid2,...]) BATCHAPPLIDS=(batch.applid1 [,batch.applid2,...]) TSO.APPLIDS=(tso.applid1 [,tso.applid2,...]) INTERACTIVE.APPLIDS=(interactive.applid1 [,interactive.applid2,...]) CICS.APPLIDS=(cics.applid1 [,cics.applid2,...]) NETID=networkid CTCA server name PNODE.LUS=(luname1 [,luname2,...]) SNODE.LUS=(luname1 [,luname2,...]) USE.SERVER.NODE = NO YES TCPAPI= (port number, IP address) CRC = (OFF ON DEFAULT) PLEXCLASS= (* plexclass, * plexclass) BUFFER.SIZE= (3072–262144 3K-256K) (ALTErnate.COMMinfo = (ALTErnate.RESTART= <u>No</u> Yes, ALTErnate.DIREction= <u>BALANCE</u> TOP, (ALTErnate.ADDRESS= ALTErnate.NODEDEF=, ALTErnate.PORT=, SOURCEIP=outbound IP address ALTErnate.TYPE=SNA TCP UDT33 LU62, ALTErnate.LOGmode entry name=, ALTErnate.USE.OUTbound= <u>Yes</u> No)) CDFTP.PLUGIN="name or location of the plugin" CDFTP.TEMPFILE="fully qualified file path and name" CONTACT.NAME="name (maximum of 40 characters)" CONTACT.PHONE="phone information" DESCRIPTION="description information")
\$\$UPDATE ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$REPLACE ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$DELETE ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE

Command	Parameters
\$\$BLKDELETE ADJACENT.NODE = . . \$\$ENDDELETE	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$BLKINSERT ADJACENT.NODE = . . \$\$ENDINSERT	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$BLKUPDATE ADJACENT.NODE = . . \$\$ENDUPDATE	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$SYNTAX . . \$\$ENDSYNTAX	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$VERIFY ADJACENT.NODE = . . \$\$ENDVERIFY	Same parameters as \$\$INSERT ADJACENT.NODE

Label	Command	Parameters
[optional]	SElect NETMAP	WHERE (NODE = (node generic (list)))
		PRINT TABLE <u>DISplay</u>
[optional]	SElect TCPXLAT	WHERE (TRTCPNAM = (tcpip.host.name (list))TRTCPADR = (tcp.net.adr (list)))
		PRINT <u>DISplay</u>

Process Commands

Label	Command	Parameters
[optional]	CHange PROCess	<p>WHERE (NODE = nodename (list) PNAME = name (list) PNUMber = number (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u></p> <p>CLASS = n</p> <p>DEBUG = trace bits</p> <p>DEST= destination node</p> <p>HOLD = YES <u>NO</u> CALL</p> <p>NETMAP.REFRESH</p> <p>PLEXCLASS = (pnode plexclass snode plexclass)</p> <p>PRTY = 0 to 15</p> <p>RELEASE</p> <p>RESTART = [NO FIRST = vol.seq.number FIRST = SER = vol.ser.number LAST = vol.seq.number LAST = SER = vol.ser.number VOLCNT = n]</p> <p>RETAIN = YES <u>NO</u> INITIAL</p> <p>STARTT = ([date day][,hh:mm:ssXM])</p>
[optional]	DELeTe PROCess	<p>WHERE (PNAME = name (list) PNUMBER = number (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u></p>
[optional]	FLUSH PROCess	<p>WHERE (PNAME = name (list) PNUMber = number (list) SUBmitter = (nodename,userid) (list)) FORCE CASE = Yes <u>No</u></p>
[optional]	SELEct PROCess	<p>WHERE (DEST = node (list) PNAME = name (list) PNUMber = number (list) QUEUE = <u>All</u> queue name SERVER=server name STATUS = process status (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u> FILE PPrint <u>TABLE</u></p>

Label	Command	Parameters
[optional]	SUSpend PROCess	WHERE (DEST = node (list) PNAME = name (list) PNUMber = number (list) QUEUE = <u>All</u> queue name STATUS = process status (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u>
[optional]	View PROCess	WHERE (DEST = node (list) PNAME = name (list) PNUMber = number (list) QUEUE = <u>All</u> queue name SERVER=server name STATUS = process status (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u> FILE PPrint <u>TABLE</u>

Signon, Signoff, Stop, and Swap Commands

Label	Command	Parameters
[optional]	SIGNON	NETMAP = network map data set name
		CASE = Yes <u>No</u>
		COMADDR = (port number, IP address alias name)
		DESCRiption = 'up to 30 character'
		ESF = <u>Yes</u> No
		FOLD = Yes <u>No</u>
		MAXRETRY = nn
		MAXWAIT = nn
		NODE = nodename
		PACCT = 'pnode accounting data'
		PRINT = destination of printed spool output
		RECONNECT
		SACCT = 'snode accounting data'
		SPACE = (<u>CYL</u> TRK blk,([prim],[sec]))
		TIMEOUT = YES <u>NO</u>
		TMPDD = preallocated data set dd name
		TMPDSN = preallocated data set name
		TRANSPORT = SNA TCP <u>NETMAP</u>
		UNIT = temporary dsn unit type
		USERID = (id,pswd,newpswd)
		VOLSER = volume serial
[optional]	SIGNOFF	
[none]	STOP CD	[FORCE <u>IMMEDIATE</u> QUIESCE STEP]
		[CDPLEX WHERE(SERVER = server name)]
		RECOVER
[optional]	SWAP NODE	nodename

Task Commands

Label	Command	Parameters
[optional]	FLUSH TASK	WHERE (TASK = taskno (list) , SERVER = server name)
		FORCE
[optional]	SElect TASK	PRInt Operator table DISplay
		WHERE (SERVER = server name)

Type Commands

Label	Command	Parameters
[optional]	INSert TYPE	TYPEKEY = typekey
		DCB = ([BLKSIZE = number of bytes ,DSORG =[DA PS PO VSAM] ,LRECL = number of bytes ,RECFM = record format])
		DISP = ([NEW OLD MOD RPL SHR] [,KEEP ,CATLG ,DELETE] [,KEEP ,CATLG ,DELETE])
		DSNTYPE = PDS LIBRARY BASIC LARGE EXTPREF EXTREQ
		AVGREC = (U K M)
		DATACLAS = data class name
		EATTR = OPT NO blanks
		KEYLEN = bytes
		KEYOFF = offset to key
		LIKE = model data set name
		LRECL = bytes
		MGMTCLAS = management class name
		RECORG = (KS ES RR LS)
		SECMODEL = (profile name [, GENERIC])
		STORCLAS = storage class name
		SPACE = (CYL TRK blk, (prim, sec, (dir)) (,RLSE , (CONTIG ,) (ROUND)) (ave_rec_len, (primary_rcds, secondary_rcds))
		UNIT = unit type
		VOL=SER = volume serial number
		IOEXIT = exitname (exitname{,parm,...})
[optional]	UPDate TYPE	Same parameters as INSert TYPE
[optional]	DELeTe TYPE	WHERE (TYPEKEY = typekey generic (list))
		PRint <u>TABLE</u>
[optional]	SELeCt TYPE	WHERE (TYPEKEY = typekey generic (list))
		PRint <u>TABLE</u>

User Commands

Label	Command	Parameters
[optional]	INSert User	USERID = (nodename,userid)
		NAME = 'user name'
		ADD TYPE = Y <u>N</u>
		ALTER TYPE= Y <u>N</u>
		READ TYPE = Y <u>N</u>
		REMOVE TYPE = Y <u>N</u>
		ADD USER = Y <u>N</u>
		ALTER USER = Y <u>N</u>
		READ USER = Y <u>N</u>
		REMOVE USER = Y <u>N</u>
		CASE = YES <u>NO</u>
		CHANGE = Y <u>N</u>
		COPY = Y <u>N</u>
		DELPR = Y <u>N</u>
		FLUSH = Y <u>N</u>
	†	GEN.CHG.PROCESS = Y <u>N</u>
	†	GEN.DEL.PROCESS = Y <u>N</u>
	†	GEN.FLS.PROCESS = Y <u>N</u>
	†	GEN.SEL.PROCESS = Y <u>N</u>
	†	GEN.SEL.STATISTICS = Y <u>N</u>
	†	CDEL = Y <u>N</u>
	†	CDELOFF = Y <u>N</u>

Label	Command	Parameters
		MAXSA = max. signon attempts
		MODALS = Y <u>N</u>
		MODIFY = Y <u>N</u>
		NSUBMIT = Y <u>N</u>
		OVCRC = Y <u>N</u>
		PASSword = initial password
		PHone = 'phone number'
		PTKTDATA = (APPL profile name, secured signon key)
		RESETSA
		RUNJOB = Y <u>N</u>
		RUNTASK = Y <u>N</u>
		SECURITY = (security id, security pswd)
		SECURERD = Y <u>N</u>
		SECUREWR = Y <u>N</u>
		SELNET = Y <u>N</u>
		SELPR = Y <u>N</u>
		SELSTAT = Y <u>N</u>
		STATCMD = Y <u>N</u>
		STOP CD = Y <u>N</u>
		SUBMIT = Y <u>N</u>
	‡	SUBMITTER.CMDS = (Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u> Y <u>N</u>)
		UPDNET = Y <u>N</u>
		VIEWPROCESS = Y <u>N</u>
[optional]	DELeTe USER	WHERE (USERID = (nodename,userid) (list))
		CASE = Yes <u>No</u>
[optional]	SELEct USER	WHERE (USERID = (nodename,userid) (generic (list)) EXCLUDE = (AUTH))
		PRint <u>TABLE</u>
		CASE = Yes <u>No</u>
‡ Valid in the IBM Connect:Direct IUI only. ‡ Valid in the batch interface only.		

Inquire Commands

Label	Command	Parameters
[optional]	INQuire CDPLEX	
[optional]	INQuire DEBUG	WHERE (SERVER = server name)

Label	Command	Parameters
[optional]	INQUIRE INITparm	WHERE (SERVER = server name)
[optional]	INQUIRE SNMP	
[optional]	INQUIRE STATDIR	STARTT = ([date day] [,hh:mm:ssXM])
[optional]	INQUIRE STATistics	
[optional]	INQUIRE TCP	

Statistics Commands

Label	Command	Parameters
[optional]	STATistics SWITCH	
[optional]	STATistics ON OFF	TYPE = (record type list)
[optional]	STATistics ARCHived	file pair number
[optional]	SElect MeSsaGe	WHERE (IDMSGID=message ID) [Short Long]

Label	Command	Parameters	subparameters
(optional)	SElect STATistics	WHERE (subparameters)	ARCHDSN = dsname (list) CASE = YES NO CCODE = (condition, completion code) EXCLUDE = (MEMber MCR) (WTO) (NOTWTO) (list) FNAME = dsname (list) * MSGID = ID (list) PNAME = name (list) PNUMBER = number (list) SERVER = server name (list) SNODE = name (list) STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) TYPE = ID (list) * USER = name (list)
		FILE PRINT TABLE SUMmary	Note: These parameters are mutually exclusive - only one of them must be specified.

Label	Command	Parameters	subparameters
(optional)	SElect STATistics	WHERE2 (subparameters)	ANODE = name (list) ARCHDSN = dsname (list) CASE=YES NO CCODE = (condition , completion code) EXCLUDE = (MEMBer) (MCR) (WTO) (NOTWTO) (list) FNAME = dsname (list) FNODE = name (list) LNODE = name (list) MSGID = ID (list) PNAME = name (list) PNODE = name (list) PNUMber = number (list) RNODE = name (list) SERVER = server name (list) SNODE = name (list) STARTT = ([date day][,hh:mm:ssXM]) STOPT = ([date day][,hh:mm:ssXM]) TNODE = name (list) TYPE = ID (list) UNODE = name (list) USER = name (list)
		FILE PRint TABLE SUMmary	Note: These parameters are mutually exclusive - only one of them must be specified.

Submit Command

Label	Command	Parameters
[optional]	SUBmit	PROC = member or DSN = dsn dsn(member)
		CASE = YES <u>NO</u>
		CLASS = n
		DEBUG = trace bits
		FASP= <u>NO</u> SSP
		FASP.BANDWIDTH= nnn nM nG
		FASP.FILESIZE.THRESHOLD=nnn nM nG
		FASP.POLICY= <u>FAIR</u> HIGH LOW FIXED
		HOLD = YES <u>NO</u> CALL
		MAXDELAY = <u>Unlimited</u> Queued hh:mm:ss 0
		NEWNAME = newname
		NOTIFY = %USER userid
		PACCT = 'Pnode accounting data'
		PLEXCLASS = (pnode class, snode class)
		PNODE = primary node name
		PNODEID = (id,pswd,newpswd)
		PRTY = 0 – 15
		REQUEUE = YES <u>NO</u>
		RETAIN = YES <u>NO</u> INITIAL
		SACCT = 'snode accounting data'
		SNODE = secondary node name SNODE=TCPNAME UDT33NAM = tcpvalue;port
		SNODEID = (id,pswd,newpswd)
		STARTT = ([date day] [,hh:mm:ssXM])
		SUBNODE = PNODE SNODE
		&symbolic name 1 = variable string1
		&symbolic name 2 = variable string2
		...
		&symbolic name n = variable string n

Note: If a variable string contains blanks, you must enclose the string in quotation marks.

Network Map Parameters

Node Type	Parameters
LOCAL.NODE =	((nodename, communications name, , superuserpassword) TCQ = (tcxdsn,tcqdsn))
ADJACENT.NODE =	((nodename, - communications name channel-range-start-addr, - remote library name IP address or Alias addr-count, - session type - security node type, - data direction restriction) - PARSESS = (max,default) - ZFBA = (Addr1, Addr2) - FASP=NO SSP - FASP.BANDWIDTH= nnn nM nG - FASP.FILESIZE.THRESHOLD=nnn nM nG - FASP.POLICY=FAIR HIGH LOW FIXED - COMPRESS.EXT=ALLOW DISALLOW FORCE - COMPRESS.STD=ALLOW DISALLOW FORCE - COMPRESS.STD.PRIMECHAR=C'x' X'xx' - SOURCEIP=IP address - SESS.SNODE.MAX = (1-255) - LDNS=hostname - LOGMODE=logmode entry name - APPLIDS=(vtam applid1 [,vtam applid2,...]) - BATCHAPPLIDS=(batch.applid1 [,batch.applid2,...]) - TSO.APPLIDS=(tso.applid1 [,tso.applid2,...]) - INTERACTIVE.APPLIDS= (interactive.applid1 [, interactive.applid2,...]) - CICS.APPLIDS=(cics.applid1 [,cics.applid2,...]) - NETID=networkid CTCA server name - PNODE.LUS=(luname1 [,luname2,...]) - SNODE.LUS=(luname1 [,luname2,...]) - USE.SERVER.NODE = NO YES - TCPAPI= (port number, IP address) - CRC = (OFF ON DEFAULT) - PLEXCLASS= (* plexclass, * plexclass) - BUFFER.SIZE= (3072-262144 3K-256K) (ALternate.COMMinfo = (ALternate.RESTART= <u>No</u> Yes, ALternate.DIRction= <u>BALANCE</u> TOP, - (ALternate.ADDRess= ALternate.NODEDEF=, ALternate.PORT=, SOURCEIP=outbound IP address - ALternate.TYPE=SNA TCP UDT33 LU62, ALternate.LOGmode entry name=, - ALternate.USE.OUTbound= <u>Yes</u> No)) - CDFTP.PLUGIN="name or location of the plugin" - CDFTP.TEMPFILE="fully qualified file path and name" - CONTACT.NAME="name (maximum of 40 characters)" - CONTACT.PHONE="phone information" - DESCRIPTION="description information" -)

Initialization Parameter Overview

Initialization parameters supply values for various IBM Connect:Direct functions. IBM Connect:Direct processes these parameters during initialization.

Global Initialization Parameters

Global Initialization Parameters	Default Values
ABEND.CODES.NODUMP = (abend code list)	no abend codes
ABEND.RUNTASK = (DUMP ABEND.CODES.NODUMP)	DUMP
ALLOC.CODES = (allocation errors)	(020C 0210 0218 0220 0234 0068 0069 006A)
ALLOC.MSG.LEVEL = INFO WARN SEVERE	INFO
ALLOC.RETRIES = no. retries	20
ALLOC.WAIT = hh:mm:ss	00:03:00
ALLOCATION.EXIT = modname	No allocation exit
CDPLEX = <u>NO</u> YES	NO
CDPLEX.MAXSERVER = maximum number of servers <u>4</u>	4
CDPLEX.TIMER = <u>5</u> number of minutes	5
CDPLEX.WLM.GOAL = (<u>NO</u> YES, exitname)	NO
CHECK.CERT.EXPIRE = NO <u>YES</u>	YES
CHECK.CERT.EXPIRE.TIME = <u>00:00:00</u> HH:MM:SS	<u>00:00:00</u> (midnight)
CHECK.CERT.EXPIRE.WARN.DAYS = <u>30</u> nnn	30 days
CKPT = nK nM	10M
CKPT.DAYS = number of days	4
CKPT.MODE = (<u>RECORD</u> BLOCK <u>BLOCK</u> RECORD <u>PDS</u> NOPDS <u>NOPDS</u> PDS <u>VSAM</u> NOVSAM <u>VSAM</u> NOVSAM)	RECORD BLOCK PDS NOPDS VSAM VSAM
COMPRESS.EXT = <u>ALLOW</u> DISALLOW	ALLOW
COMPRESS.NEGO.FAIL = <u>STEP</u> PROCESS	STEP
COMPRESS.NETMAP.OVERRIDE = <u>ALLOW</u> DISALLOW	ALLOW
COMPRESS.STD = <u>ALLOW</u> DISALLOW	ALLOW
CONFIRM.COLD.START = YES <u>NO</u>	NO
CRC = (<u>OFF</u> ON, <u>Yes</u> No)	OFF, Yes
CTCA = YES <u>NO</u>	NO

Global Initialization Parameters	Default Values
CTCA.TIMER = number of seconds to wait when establishing a CTCA connection	180
DATEFORM = (MDY DMY YMD YDM)	MDY
DEBUG = nnnnnnnnnn	00200000
DESC.CRIT = (descriptor code)	2
DESC.NORM = (n, n, ...)	none
DESC.TAPE = (n, n, ...)	2
DSNTYPE = YES <u>NO</u>	NO
ECZ.COMPRESSION.LEVEL = <u>1</u> 2 3 4 5 6 7 8 9	1
ECZ.MEMORY.LEVEL = 1 2 3 <u>4</u> 5 6 7 8 9	4
ECZ.WINDOWSIZE = 8 9 10 11 12 <u>13</u> 14 15	13
ESF.WAIT = hh:mm:ss	00:03:00
EXPDT = TT DD TD DT ALL <u>NONE</u> (TT,DD,TD,DT)	NONE
EXTENDED.RECOVERY = <u>NO</u> YES	NO
FIPS = YES <u>NO</u>	NO. If FIPS = YES, you must specify a filename for the SECURE.DSN initialization parameter. The Local parameter file record should enable FIPS as Y.
FASP= <u>NO</u> SSP	NO
FASP.BANDWIDTH=nnn nM nG	FASP bandwidth
FASP.FILESIZE.THRESHOLD=nnn nM nG	FASP file threshold size
FASP.POLICY= <u>FAIR</u> HIGH LOW FIXED	FAIR
GDGALLOC = <u>GENERATION</u> DSNAME	GENERATION
GDGENQ = YES NO	YES, when GDGALLOC = GENERATION or NO, when GDGALLOC = DSNAME
IMMEDIATE.SHUTDOWN = <u>I</u> R	I
INITPARM.BACKUP = member	none
INVOKE.ALLOC.EXIT = SEND RECV <u>BOTH</u>	BOTH
INVOKE.ALLOC.EXIT.ON.RESTART = NO <u>YES</u>	YES
INVOKE.SPOE.ON.SNODEID = <u>NO</u> YES	NO
MAX.AGE = (number of days the entries are left in TCQ, ALL <u>HE</u> HO HP HS PR RA RH)	number of days = 0, HE
MAX.AGE.TOD = time	midnight and initialization
MAXPRIMARY = number of primary sessions	6
MAXPROCESS = number of executing PNODE plus SNODE Processes	MAX-PRIMARY plus MAX-SECONDARY

Global Initialization Parameters	Default Values
MAXRETRIES = number of retries	7
MAXSECONDARY = number of secondary sessions	6
MAXSTGIO = max. storage in bytes for nontape sequential data set transfers	61440
MAX.TAPE = number of tape Processes NONE	10
MAXUSERS = number of users	6
MCS.CLIST = Operator interface CLIST library name	none
MCS.SIGNON = [SIGNON USERID = (userid, password) NETMAP = network map]	none
MULTI.COPY.STAT.RCD = not set CT MC M2	not set
NETMAP.CHECK = NO (<u>ALL</u> TCP, <u>ALL</u> BOTH NODENAME, <u>FAIL</u> WARN PASS)	ALL ALL FAIL
NETMAP.CHECK.ON.CALL= YES <u>NO</u>	NO
NODE.QUIESCE.OFF = NODENAME	none
NODE.QUIESCE.ON = NODENAME	none
NODE.TRACE.OFF = NODENAME	none
NODE.TRACE.ON = NODENAME	none
NON.SWAPABLE = <u>YES</u> NO	YES
PDSE.SHARING = YES <u>NO</u>	NO
PDSENQ = YES <u>NO</u>	NO
PROCESS.RETENTION = YES <u>NO</u>	NO
PRTYDEF = Process default priority	10
QUIESCE = YES <u>NO</u>	NO
QUIESCE.NODE = nodename	none
REMOTE.DUMMY.PASSWORD = [<u>YES</u> INTERNAL]	YES
REQUEUE = <u>YES</u> NO	YES
RESET.ORIGIN.ON.SUBMIT = YES <u>NO</u>	NO
REUSE.SSESSIONS = <u>YES</u> NO	YES
ROUTCDE.CRIT = (route code)	(8,11)
ROUTCDE.NORM = (route code)	11
ROUTCDE.TAPE = (route code)	(5, 11)
RUN.JOB.EXIT = modname	none
RUNJOBID = <u>USER</u> CD	USER
RUN.TASK.EXIT = modname	none
RUNTASK.RESTART = YES <u>NO</u>	NO

Global Initialization Parameters	Default Values
S+CMD.ENFORCE.SECURE.CONNECTION = <u>YES</u> NO	YES
SECURE.DSN = filename	none. If the FIPS initialization parameter is set to YES, a filename must be specified for the SECURE.DSN parameter.
SECURE.SSL.PATH.PREFIX = prefix	none
SECURITY.EXIT = (modname, DATASET ALL, PSTKT) OFF SECURITY = (modname, DATASET ALL, PSTKT) OFF	C:D Authorization Facility
SECURITY.NOTIFY = YES <u>NO</u> HOLD	NO
SESSION.HIGHWATER.SMF = (<u>133</u> ,60)	133,60
SERVER.MSU = <u>0</u>	0
SERVER.TYPE = <u>PROD</u> TEST	PROD
SNA = <u>YES</u> NO	YES
SNMP = YES <u>NO</u>	NO
SNMP.DSN = data set name data set name (member)	none
SNMP.MANAGER.ADDR = hostname IP address	IP address or hostname
SNMP.MANAGER.PORTNUM = port-number	162
SNODE.ARCH.RECALL.WAIT Yes <u>NO</u>	NO
STAT.ARCH.CONFIRM = YES <u>NO</u>	NO
STAT.BUFFER.ESDSDATA = number of ESDS data buffers STAT.BUFFER.KSDSDATA = number of KSDS data buffers STAT.BUFFER.KSDSINDX = number of KSDS index buffers	6
STAT.ERROR = <u>ABEND</u> DISABLE	ABEND
STAT.EXCLUDE = (record type list)	none
STAT.INIT = <u>WARM</u> COLD	WARM
STAT.QUEUE.ELEMENTS = statistics record queue	1500
STAT.SNODEID = (NO YES, NO YES)	(NO, YES)
STAT.SWITCH.SUBMIT = dsn [member]	none
STAT.SWITCH.TIME = (hh:mm:ss, ...)	none
STAT.TPREC = (start_time, end_time, snaps_per_hour)	none
STAT.USER = (userid, [password])	Security ID of DTF region
STATISTICS.EXIT = module name (modname[, <u>MANAGER</u>] [,SERVER] [,BOTH])	none
STRNO.MSG = number <u>10</u>	10
SUBMIT.EXIT = modulename	none
SYSOUT = class	none

Global Initialization Parameters	Default Values
TAPE.PREMOUNT = YES <u>NO</u> LIST	NO
TAPEIO = EXCP <u>BSAM</u>	BSAM
TAPEMOUNT.EXIT = modname	none
TCP = OES <u>NO</u>	NO
TCP.API.DISCONNECT.ERROR.MESSAGE YES NO	
TCP.API.LISTEN = ((addr1 , port1) , (addrn , portn))	If the parameter is specified with an address only, the port can default. The default port is 1363.
TCP.API.TIMER = <u>00:00:00</u> hh:mm:ss	<u>00:00:00</u> meaning that no timer is used.
TCP.CONNECT.TIMEOUT = <u>30</u> number of seconds	30
TCP.FMH.TIMER = hh:mm:ss	<u>00:10:00</u>
TCP.LISTEN = ((addr1 , port1) , (addrn , portn))	If the parameter is specified with an address only, the port can default. The default port is 1364.
TCP.RUNTASK.TIMER = hh:mm:ss	<u>00:00:00</u>
TCP.SOURCEIP (IPv4 addr , IPv6 Addr)	Source IP address type
TCP.SRC.PORTS = (ip.address,port-ranges),(ip.address2,port1,port2), - (ip.address3,port-ranges) TCP.SRC.PORTS = (ip.address/submask,port-ranges), ... TCP.SRC.PORTS = (ip.address/0xFFFFFFFFFFFFFFFF,ports,ranges)	none
TCP.SRC.PORTS.LIST.ITERATIONS = number of scans	none
TCP.TIMER = wait time in seconds	300
TCQ = <u>WARM</u> COLD	WARM
TCQ.THRESHOLD = <u>NO</u> YES nn	NO
THIRD.DISP.DELETE = <u>YES</u> NO	YES
TRACE.BUFFER = nnn <u>2</u>	2 MB
TRANS.SUBPAS = <u>YES</u> NO	YES
UDP.SRC.PORTS = (ip.address,port-ranges),(ip.address2,port1,port2), - (ip.address3,port-ranges)	none
UDP.SRC.PORTS.LIST.ITERATIONS = number of scans	none
UDT = YES <u>NO</u>	NO
UDT.MAXPROCESS = <u>2</u> number of executing UDT Processes	2
UDT33.LISTEN = ((ipaddr,port)[,(ipaddr,port)...])	The default port is 1366.
UPPER.CASE = YES <u>NO</u>	NO
V2.BUFSIZE = (maximum transmission buffer size, TCP/IP send/receive buffer size)	32K, 128K

Global Initialization Parameters	Default Values
WTMESSAGE = <u>NO</u> YES (YES, nnn)	NO
WTRETRIES = hh:mm:ss <u>00:3:00</u>	00:03:00
XCF.NAME = XCF group name	NONE
ZEDC = ON <u>OFF</u> PREFERRED	OFF
ZFBA = YES or <u>NO</u>	NO
ZIIP = <u>NONE</u> EXTCOMP SSLTLS ALL PROJECT	NONE

System File Initialization Parameters

System File Initialization Parameters
AUTHDSN = dsn
CKPTDSN = dsn
MSGDSN = dsn
NETDSN = dsn
STAT.ARCH.DIR = archive directory dsn
STAT.DSN.BASE = dsn base STAT.FILE.PAIRS = number
TYPEDSN = dsn

Local Initialization Parameters

Local Initialization Parameters	Default Values
CDPLEX.INITPARM.BACKUP = member	none
CDPLEX.MANAGER = <u>NO</u> YES	NO
CDPLEX.MSGID = <u>NONE</u> xx	NONE
CDPLEX.PLEXCLASSES = (*,plexclass, ..., plexclass)	*
CDPLEX.REDIRECT = ((INT_IPv4,EXT_IPv4),(INT_IPv6,EXT_IPv6), (INT_UDT_IPv4,EXT_UDT_IPv4),(INT_UDT_IPv6,EXT_UDT_IPv6))	Defaults to the first IPV4 or IPV6 address for the server.
CDPLEX.REDIRECT.EXCEPTION = ((Mgr-IP, Svr-IP, Svr-port, Exception-IP, Exception-port),...)	none
CDPLEX.SERVER = IBM Connect:Direct/Server name	none
CDPLEX.SERVER.JOBDSN = data set name	none
CDPLEX.SERVER.JOBMEM = ((member name, server name), ...)	none
CDPLEX.SERVER.NODE = nodename	none
CDPLEX.TIMER = <u>5</u> number of minutes	5
CDPLEX.VTAM = (VTAM-APPL, P/S-Node-APPL)	none
CTCA = YES <u>NO</u>	NO

Local Initialization Parameters	Default Values
CTCA.TIMER = number of seconds to wait when establishing a CTCA connection	180
DEBUG = nnnnnnnnnn	00200000
MAXUSERS value	6
MAXPRIMARY = number of primary sessions	
MAXPROCESS = number of executing PNODE plus SNODE Processes	MAX-PRIMARY plus MAXSECONDARY
MAXSECONDARY = number of secondary sessions	6
MAX.TAPE = number of tape Processes	10
QUIESCE = YES <u>NO</u>	NO
SECURITY.EXIT = (modname,DATASET ALL,PSTKT) OFF SECURITY = (modname,DATASET ALL,PSTKT) OFF	C:D Authorization Facility
SNA = <u>YES</u> NO	YES
STAT.INIT = <u>WARM</u> COLD	WARM
STATISTICS.EXIT = modulename (modname[, <u>MANAGER</u>] [,SERVER] [,BOTH])	none
TCP = OES <u>NO</u>	NO
TCP.API.LISTEN = ((addr1 , port1) , (addrn , portn))	If the parameter is specified with an address only, the port can default to 1363.
TCP.LISTEN = ((addr1 , port1) , (addrn , portn))TCP.TIMER = wait time in seconds	If the parameter is specified with an address only, the port can default. The default port is 1364.
	300
TCQ = <u>WARM</u> COLD	WARM
TRACE.BUFFER = nnn <u>2</u>	2
UDT = YES <u>NO</u>	NO
UDT.MAXPROCESS = <u>2</u> number of executing UDT Processes	2
UDT33.LISTEN = ((ipaddr,port)[,(ipaddr,port)...])	The default port is 1366.
UPPER.CASE = YES <u>NO</u>	NO
V2.BUFSIZE = (maximum buffer size for this transmission, TCP/IP send/receive buffer size)	32K, 128K
ZIIP = <u>NONE</u> EXTCOMP SSLTLS ALL PROJECT	NONE
ZFBA = YES <u>NO</u>	NO

Status Codes

Code	Description
EX	Process executing

Code	Description
HC	Held for call
HE	Held in error
HI	Held initially
HO	Held by operator
HP	Held due to Process error
HR	Submitted with RETAIN=YES or held for process retry (intelligent retry)
HS	Held for suspension
PR	Process being retained
RA	Held for restart due to allocation error
RH	Held for restart
RS	Restarting
SS	Session with other node is being started.
WA	Waiting acknowledgement
WC	Waiting connection
WR	Waiting restart
WS	Waiting designated start time
WT	Waiting transport protocol
WX	Waiting eligible IBM Connect:Direct/Server

Trace Types

DEBUG Setting†	Trace Type	DDNAME Where Output is Directed
80000000	COPY Routine and RUN TASK trace	RADBDD01
10000000	Full TPCB/SYMBOLICS from DMCBSUBM	DMCBSUBM
08000000	Session manager trace	RADBDD05
04000000	Separate trace per task (Example: "R0000005" to trace TASK 5)	Rnnnnnnn
02000000	API session trace	RADBDD07
01000000	DGADBSUB trace	RADBDD08
00400000	TCQSH from DGADCPYR	DMCOPYRT
00200000	Make each SVC dump unique	N/A
00100000	SECURITY Trace Control	SECURITY
00040000	GETMAIN/FREEMAIN trace	RADBDD16
00008000	I/O buffer trace	RADBDD21
00004000	WTO all dynamic allocation parameters	RADBDD22

DEBUG Setting†	Trace Type	DDNAME Where Output is Directed
00002000	IBM Connect:Direct/Plex traces	
	ACTION queue manager trace	CDPLXACT
	CKPT queue manager trace	CDPLXCKP
	TCQ queue manager trace	CDPLXTCQ
	STATS queue manager trace	CDPLXSTA
	First REQUEST queue manager trace	CDPLXREQ
	Second and subsequent REQUEST queue manager trace (Example: "CDPLXR03" to trace third queue manager)	CDPLXRnn
	JOIN queue manager trace	CDPLXJOI
00001000	Workload balancing trace	CDPLXWLB
00000800	zIIP-related trace	CDZIIP
00000100	In-storage tracing only Note: The size of this in-storage table is controlled by TRACE.BUFFER	N/A
00000080	RPL trace – long	RPLOUT
00000040	RPL trace – short	RPLOUT
00000020	Version 2 session trace	RADBDD33
00000008	Logon exit trace	RADBDD35
00000004	Logon processor trace	RADBDD36
00000002	SCIP exit trace	RADBDD37
00000001	SNMP Trap trace	SCTRAPDD
00000400	ESTA* trace	CDESTAE
† The DEBUG=X'nnnnnnnn' initialization parameter turns on a specific trace option or any combination of options, where nnnnnnnn represents the debug setting in hexadecimal.		

Note: The Modify command requests the same output as the DEBUG initialization parameter.

Chapter 11. Connect:Direct for z/OS EAV support

IBM Connect:Direct for z/OS supports Extended Address Volume (EAV) features at the z/OS 1.12 level.

An EAV is a volume with more than 65,520 cylinders. An EAV increases the amount of addressable direct access storage device (DASD) storage per volume beyond 65,520 cylinders by changing how tracks on extended count key data (ECKD) volumes are addressed. In z/OS 1.12, the largest supported EAV volume is 262,668 cylinders. In z/OS 1.13, the largest supported EAV volume is 1,182,006 cylinders.

To utilize the Connect:Direct for z/OS support for EAV, you need to be familiar with EAV concepts and terminology, such as Track Managed Space (TMS), Cylinder Managed Space (CMS), Base Addressing Space (BAS), Extended Addressing Space (EAS), breakpoint value (BPV) and the default behavior when EATTR is not specified.

For more information on EAV concepts and terminology, refer to [DFSMS Extended Address Volumes in the z/OS Introduction and Release Guide](#), or IBM Redbook [DFSMS V1.10 and EAV Technical Guide](#). For the most recent updates to EAV, see [z/OS DFSMS Using the New Functions](#).

Connect:Direct for z/OS for EAV provides support for the following features:

- Support for EATTR in COPY statements, TYPE records, and DGADTDYN.
- Support for 28-bit cylinder numbers in format-3, format-4, format-8 and format-9 DSCBs.
- Support for any data set type supported by z/OS in CMS.

EATTR parameter

Connect:Direct for z/OS support for EAV includes support for the EATTR data set attribute.

EATTR has two valid values:

- OPT - (optional) the data set can have extended attributes. If it is allocated on an EAV, it is biased towards being allocated in CMS.
- NO – the data set cannot have extended attributes. If it is allocated on an EAV, it is allocated in TMS.

In Connect:Direct for z/OS, the EATTR value can come from a number of places. When it is specified in more than one place, this order of precedence is used:

1. EATTR (in COPY statement) or SYSOPTS EATTR.
2. TYPE when the TYPE record contains an EATTR.
3. LIKE or SYSOPTS LIKE (dataset attribute).
4. DATACLAS or SYSOPTS DATACLAS.
5. TYPE when the TYPE record contains a DATACLAS.
6. The COPY FROM dataset's EATTR setting (propagation).
7. Otherwise, if an EATTR value is not specified when a data set is created, the system uses a default behavior equivalent to OPT for VSAM, and NO for non-VSAM.

COPY statement support

EATTR can be added to the PROCESS COPY statement TO side.

To add EATTR to a PROCESS COPY statement TO side, use this syntax:

```
COPY FROM (...) TO (DSN=... DISP=(NEW,...) EATTR=[OPT|NO|<blank>])
```

Note: A blank or null value for EATTR is valid in COPY statements and will ensure the created data set has no EATTR value.

The EATTR parameter is on the COPYFILE Receiving File panel:

```

DMI@CF20          COPYFILE - RECEIVING FILE (z/OS or OS/390)      TIME: 14:35
CMD ==>
NODE NAME: CD.ART
RECEIVING DSNAME:  EPETEL.O.CKPT.TEST_____
_____
_____
DISPOSITION:      ( RPL , CATLG_ , _____ )
UNIT PARAMETER:   ( _____ )
VOLUME SERIAL(S): ( _____ ) )
    RETAIN: _ (Y OR N)    COUNT: ____ (1-255)
DCB PARAMETER:    ( _____ )
LABEL PARAMETER:  ( _____ )
SPACE:            ( _____ )
TYPE KEY:         _____    MSVGP NAME: _____
SYSOPTS:          _____
SMS: DATA CLASS: _____    STORAGE CLASS: _____    MGMT CLASS: _____
    AVERAGE RECORDS: _    DSNTYPE: _____    EATTR: _____
    VSAM ORGANIZATION: ____    KEY LENGTH: ____    KEY OFFSET: _____
    LIKE DSNAME: _____
    SECURITY MODEL: _____
    GENERIC MODEL: ____    (YES OR NO)

```

Figure 5. COPYFILE Receiving File panel



Attention: Do not attempt to deliver a file with EATTR present in a COPY statement to a node that is running a version of Connect:Direct for z/OS without EAV support. It will cause these errors on the receiving node:

```

SDAA098I - UNKNOWN 5 BYTE KEYWORD FOUND IN SYSOPTS
SDAA001I - SYSOPTS=EATTR
SDAA004I - SYSOPTS=" EATTR=OPT"

```

VIEW PROCESS support

When EATTR is specified in the process, it appears in the output for the VIEW PROCESS command.

VIEW PROCESS example:

```
VIEW PROC WHERE(CASE=YES,PNUM=000002)
```

```

=====
                        VIEW PROCESS
=====
                        PROCESS NAME: EAV1      PROCESS NUMBER: 2
=====

EAV1      PROCESS  SNODE=CD.BOB                -
                        PNODE=CD.ART            -
                        HOLD=YES                -
                        RETAIN=NO               -
                        CLASS=1                 -
                        PRTY=10

COPY      COPY FROM                                -
            (PNODE                                -
            DSN='EPETE1.F80.DATA1'              -
            DISP=(SHR)                          -
            )                                    -
            TO                                    -
            (SNODE                                -
            DSN='EPETE1.F80.DATA2'              -
            SPACE=(CYL,(000000021,000000021),,,ROUND) -
            DISP=(NEW,CATLG,DELETE)             -
            UNIT=(SYSDA,,)                      -
            SYSOPTS=\ " EATTR=OPT" \            -
            )

```

Figure 6. EAV1 View Process

TYPE file support

EATTR can be added to INSERT TYPE and UPDATE TYPE commands. EATTR appears in the output for the SELECT TYPE. The EATTR parameter appears on the TYPE Record SMS/VTAM Attributes panel.

EATTR has two valid values: OPT (optional) and NO. Blanking out the parameter will remove it from the TYPE record.

INSERT TYPE and UPDATE TYPE commands:

- `INS TYPE TYPEKEY=EATTR1 EATTR=OPT`

The data set can have extended attributes. If it is allocated on an EAV, it is biased towards being allocated in the CMS. (optional)

- `UPD TYPE TYPEKEY=EATTR1 EATTR=NO`

The data set cannot have extended attributes. If it is allocated on an EAV, it is allocated in the TMS.

- `UPD TYPE TYPEKEY=EATTR1 EATTR=`

This setting erases the value set in the type file.

SELECT TYPE example:

```
SEL TYPE WHERE (TYPEKEY=(EATTR1))
```

```

=====
                        SELECT TYPE DEFAULTS
=====

TYPE KEY => EATTR1          DATE CREATED => 01.09.2012

DISP=(NEW,CATLG,DELETE)
SPACE=(CYL, (21,21,45))
UNIT=EAVTEST
EATTR=OPT
DCB=(DSORG=PO,LRECL=80,BLKSIZE=3120,RECFM=FB)

```

Figure 7. EATTR1 Select Type Defaults

TYPE Record SMS/VTAM Attributes panel example:

```

DMIIUTPV                      Type Record SMS/VSAM Attributes                      10:28
CMD ==>

Operation  ==> UPDATE
Type Key   ==> T

(SMS=)     Data Class          ==> _____
           Management Class    ==> _____
           Storage Class       ==> _____

(Other=)   EATTR              ==> _____ (OPT or NO)
           DSNTYPE            ==> _____ (PDS, LIBRARY, LARGE, BASIC, EXTREQ,
                                           EXTPREF)

(MODEL=)   Like Data Set Name ==> _____
           Security Profile    ==> _____
           Generic Profile     ==> _____ (YES or NO)

(SPACE=)   Average Record Value ==> _ (U, K or M)

(VSAM=)    Organization ==> _____ (ES - ESDS, KS - KSDS, RR - RRDS, LS - LDS)
           Key Offset      ==> _____ (0 - 32760)
           Key Length      ==> _____ (1 - 255)

```

Figure 8. TYPE Record SMS/VTAM Attributes panel

DGADTDYN support

The EATTR parameter can be used with DGADTDYN ALLOCATE.

To add EATTR to DGADTDYN ALLOCATE, use this syntax:

ALLOC DSN=... DISP=(NEW,...) ... EATTR=[OPT|NO] ...

Note: A blank or null value for EATTR is invalid with DGADTDYN and will cause a syntax error. If a value is not specified, the EATTR text unit will not be included in the SVC 99.



Attention: Do not attempt to deliver a file with EATTR present in ALLOC parameters for DGADTDYN (DMRTDYN) to a node that is running a version of Connect:Direct for z/OS without EAV support. It will cause these errors on the receiving node:

```

SDAA048I - SYNTAX ERROR
SDAA001I - EATTR=EATTR

```


z/OS operating system support of EAV

There are several things to consider before deciding to put datasets on EAV, and deciding between CMS and TMS.

Datasets on EAV

z/OS support for EAV was introduced in release 1.10 (DS8000 Release 4.0 Licensed Internal Code is also required). A DS8700 can define CKD volumes with up to 262,668 cylinders (~223 GB) and a DS8800 can define 1,182,006 cylinders (~1 TB) per volume. An EAV is a 3390A with more than 65520 cylinders. It is divided into the base addressing space (BAS) containing cylinders 0-65535 and the extended address space (EAS) containing the rest. An EAV is also divided into the track managed space (TMS) containing cylinders 0-65519, and cylinder managed space (CMS) containing the rest.

EATTR provides a way to directly set the allocation behavior for an EAS eligible data set. This gives the user a way to override the default allocation behavior for any data set, regardless of whether it's EAS eligible at that OS level. EATTR=NO means it cannot have extended attributes, and so if allocated on an EAV, it is always allocated in the TMS (track managed space). EATTR=OPT means it can have extended attributes, and so if allocated on an EAV, it is biased towards being allocated in the CMS (cylinder managed space). Not specifying EATTR means to use the default allocation behavior for the data set type. For VSAM data sets, the default behavior is the equivalent of EATTR=OPT. For non-VSAM data sets, the default behavior is the equivalent of EATTR=NO. The EATTR setting is saved in DS1EATTR in the F1 or F8 DSCB (data set control block).

CMS and TMS

The system uses this logic to decide between CMS and TMS:

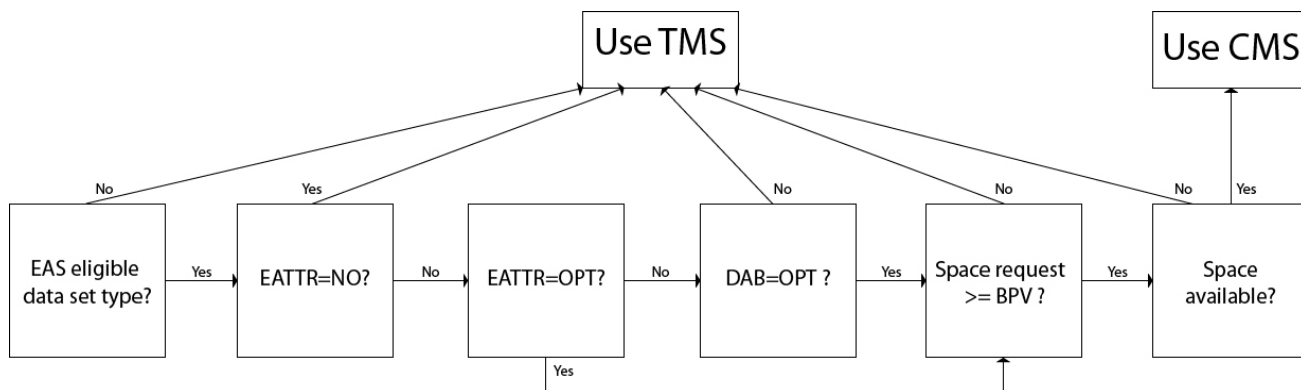


Figure 9. EAV Logic

Note: For more information about defining the Break Point Value (BPV), see page 314 in the [IBM Redbook DFSMS V1.10 and EAV Technical Guide](#)

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown in the next column.

© 2015.

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2015.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Commerce™, Information Broker®, and Integrator® are trademarks, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,

INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Part Number:
Product Number: 5655-X01