

Connect:Direct® for z/OS

Quick Reference

Version 4.8

Connect:Direct for z/OS Quick Reference

Version 4.8

First Edition

(c) Copyright 1998-2009 Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of the release notes.

STERLING COMMERCE SOFTWARE

*****TRADE SECRET NOTICE*****

THE CONNECT:DIRECT SOFTWARE ("STERLING COMMERCE SOFTWARE") IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice. As and when provided to any governmental entity, government contractor or subcontractor subject to the FARs, this documentation is provided with RESTRICTED RIGHTS under Title 48 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARS, this documentation and the Sterling Commerce Software it describes are provided pursuant to the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation.

These terms of use shall be governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14, and FAR 52.227-19(c)(1-2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation including DFAR 252.227-7013, DFAR 252,227-7014, DFAR 252.227-7015 and DFAR 252.227-7018, all as applicable.

The Sterling Commerce Software and the related documentation are licensed either "AS IS" or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

Connect:Direct is a registered trademark of Sterling Commerce. Connect:Enterprise is a registered trademark of Sterling Commerce, U.S. Patent Number 5,734,820. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

Sterling Commerce, Inc.

4600 Lakehurst Court Dublin, OH 43016-2000 *
614/793-7000

Contents

Introduction	5
Notational Conventions	6
Connect:Direct Process Statements	7
Process Statement	7
Intrinsic Variables	8
COPY Statement	9
RUN JOB Statement	11
RUN TASK Statement	11
Run Task Utility Programs and Parameters	11
SUBMIT Statement	12
SYMBOL Statement	12
Modal Statements	13
Connect:Direct Commands	14
Modify Command	14
Network Map Commands	15
Process Commands	16
Signon, Signoff, Stop, and Swap Commands	17
Task Commands	18
Type Commands	18
User Commands	19
Inquire Commands	21
Statistics Commands	21
Submit Command	22

Network Map Parameters	24
Initialization Parameters	25
Status Codes	30
Connect:Direct Trace Types	31

Introduction

The *Connect:Direct for z/OS Quick Reference* provides a condensed listing of the commands, statements, and parameters for Connect:Direct for z/OS.

The contents of this manual are extracted from the following sources:

- ◆ Connect:Direct Processes Web site
- ◆ *Connect:Direct for z/OS User's Guide*
- ◆ *Connect:Direct for z/OS Installation Guide*
- ◆ *Connect:Direct for z/OS Administration Guide*

Please consult these books for detailed descriptions of the commands, statements, and parameters.

Notational Conventions

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 (<i>nnn*</i>).
' '	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.

Connect:Direct Process Statements

Process Statement

Label	Statement	Parameters
Process name (required)	PROcEss	S NO D E= secondary node name S NO D E=TCPNAME=tcpsvalue;port S NO D E=UDT33NAM = udtvalue;port PNO D E = primary node name %PNO D E PNO D EID = (id [,pswd] [,newpswd]) SNO D EID = (id [,pswd] [,newpswd]) PACCT = 'pnode accounting data' SACCT = 'snode accounting data' CLASS = n CRC = (OFF ON) DEBUG = trace bits 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 STS SSL TLS STARTT = ([date day][,hh:mm:ssxm]) &symbolicName 1 = variable string 1 &symbolicName 2 = variable string 2 . . . &symbolicNamen = 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	<p>Specifies an allocated DD statement, which references a DSN to be passed to a Process being submitted.</p> <p>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.</p>
%JDATE	<p>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.</p> <p>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.</p>
%JOBID	Specifies the job identifier.
%JOBNM	Specifies the job name.
%JUSER	Specifies a variable that resolves to the USERID of the submitted job.
%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.
%SUBDATE	<p>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.</p> <p>The value returned can be used to create a file name on the node receiving the file.</p>
%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	<p>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.</p> <p>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</p>
%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
		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-rcd-key] [,TRTCH = C E T ET COMP NOCOMP])
		DISP = ([OLD <u>S</u> HR], [KEEP DELETE], [KEEP DELETE])
		RESGDG = <u>S</u> ub Run
		LABEL = ([file sequence number] [, <u>S</u> L 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>R</u>EPLACE 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,...])
		SYSOPTS = "DBCS = (tablename, so, si, PAD PAD = pc,LOGIC=A B)" "codepage =(from code set, to Unicode code set)" "parameter1 [parameter2,...]" "datatype = <u>t</u> ext binary" "xlate = <u>n</u> o yes" "strip.blanks = yes <u>n</u> o" "precomp = yes <u>n</u> o" DATATYPE = text binary XLATE = <u>n</u> o yes STRIP.BLANKS = <u>n</u> o yes
)

† HFS File Only

Label	Statement	Parameters
	TO	(DSN = data set name/password FILE = file name
		PNODE <u>S</u> NODE
		TYPE = typekey
		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 = ([<u>N</u> EW OLD MOD RPL SHR] [,KEEP <u>C</u> ATLG] [,KEEP <u>C</u> ATLG 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
		RECORG = KS ES RR LS
		SECMODEL = (profile-name [,GENERIC])
		STORCLAS = storage-class-name
		LABEL = ([file sequence number] [, <u>S</u> AL BLP LTM NL] [,PASSWORD NOPWREAD] [,IN OUT] [,RETPD = nnnn EXPDT = [yyddd yyyy/ddd]])
		MSVGP = MS group name
		SPACE = (CYL TRK <u>b</u> lk, (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] [,SER = (serial-no [,serial-no,...])]) ([SER = (serial-no [,serial-no,...]) ,REF = dsn])
		BUFND = number
		IOEXIT = exit-name (exit-name [,parameter,...])
		DATAEXIT = exit-name (exit-name [,parameter,...])

† HFS File Only

Label	Statement	Parameters
		SYSOPTS = "UNIQUE=YES" "DBCS = (tablename, so, si, PAD PAD = pc, LOGIC=A B)" "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
)
		CKPT = nK nM
		COMPRESS [[PRIMEchar = X'40' X'xx' C'c'] EXTended]
		SECURE = (ENCRYPT.DATA=Y N algorithm name,SIGNATURE=Y or ENC=Y N algorithm name,SIG=Y N)

† HFS File Only

RUN JOB Statement

Label	Statement	Parameters
[optional]	RUN JOB	(DSN = dsn[(member)])
		<u>P</u> NODE <u>S</u> NODE

RUN TASK Statement

Label	Statement	Parameters
[optional]	RUN TASK	(PGM = program name, PARM = (parameter, [,parameter, ...]))
		SYSOPTS = "parameter [,parameter, ...]"
		<u>P</u> NODE <u>S</u> NODE

Run Task Utility Programs and Parameters

Program	Parameters
DMNOTIFY	CL4'GOOD' CL4'FAIL' file name
DMNOTFY2	CL4'GOOD' CL4'FAIL' file name user ID user ID list
DMRTDYN	ALLOC UNALLOC CONCAT DECONCAT LOCATE
DMRTAMS	sysprint output parameters control statement parameters
DMRTSUB	JCL source subsequent parameters

Program	Parameters
RTGDGSUB	%SRCDSN %DSTDSN %PROCNAME %PROCNUM

SUBMIT Statement

Label	Statement	Parameters
[optional]	SUBmit	DSN = dsn [(member)]
		CASE = Yes <u>No</u>
		CLASS = n
		DEBUG = trace bits
		HOLD = Yes <u>No</u> 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]
[optional]	GOTO	statement label
[no label]	EXIT	[none]

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.ON = (node name, debug bits)
	†	NODE.TRACE.OFF = node name
		SESSIONS = Quiesce Resume
		WHERE(SERVER = server name)

† The BITS.ON, BITS.OFF, DEBUG, and NODE.TRACE parameters affect debug traces. See *Connect:Direct Trace Types* on page 31 for a list of trace bits.

Network Map Commands

Command	Parameters
\$\$INSERT	
ADJACENT.NODE =	<pre> ((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) SOURCEIP=IP address SESS.SNODE.MAX = (1-255) LDNS=hostname ENVIRONMENT=operating environment LOGMODE=logmode entry name APPLIDS=(vtam applid1 [,vtam 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)) CONTACT.NAME="name" CONTACT.PHONE="phone information" DESCRIPTION="description information")) </pre>
\$\$UPDATE	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$DELETE	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
\$\$BLKDELETE	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
.	
.	
\$\$ENDDELETE	
\$\$BLKINSERT	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
.	
.	
\$\$ENDINSERT	

Command	Parameters
\$\$BLKUPDATE	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
.	
\$\$ENDUPDATE	
\$\$SYNTAX	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
.	
\$\$ENDSYNTAX	
\$\$VERIFY	
ADJACENT.NODE =	Same parameters as \$\$INSERT ADJACENT.NODE
.	
\$\$ENDVERIFY	

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	WHERE (NODE = nodename (list) PNAME = name (list) PNUMBER = number (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u> CLASS = n DEBUG = trace bits DEST= destination node HOLD = YES <u>NO</u> CALL NETMAP.REFRESH PLEXCLASS = (pnode plexclass snode plexclass) PRTY = 0 to 15 RELEASE RESTART = [NO FIRST = vol.seq.number FIRST = SER = vol.ser.number LAST = vol.seq.number LAST = SER = vol.ser.number VOLCNT = n] RETAIN = YES <u>NO</u> INITIAL STARTT = ([date day][,hh:mm:ssXM])

Label	Command	Parameters
[optional]	DELeTe PROCess	WHERE (PNAME = name (list) PNUMBER = number (list) SUBmitter = (nodename,userid) (list)) CASE = Yes <u>No</u>
[optional]	FLUSH PROCess	WHERE (PNAME = name (list) PNUMBER = number (list) SUBmitter = (nodename,userid) (list)) FORCE CASE = Yes <u>No</u>
[optional]	SELeCt 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 PRint <u>TABLE</u>
[optional]	SUSpend PROCess	WHERE (DEST = node (list) PNAME = name (list) PNUMBER = number (list) QUEUE = <u>All</u> Exec Hold Timer Wait 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 No FILE PRint <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) ESF = <u>Yes</u> No

Label	Command	Parameters
		FOLD = Yes <u>No</u>
		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]))
		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 IMMEDIATE <u>QUIESCE</u> 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])

Label	Command	Parameters
		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
		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>

† Valid in the Connect:Direct IUI only.

‡ Valid in the batch interface only.

Label	Command	Parameters
		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>
		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'
		PTICDATA = (APPL profile name, secured signon key)
		RESETSA
		RUNJOB = Y <u>N</u>
		RUNTASK = Y <u>N</u>
		SECURITY = (security id, security pswd)
		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))

† Valid in the Connect:Direct IUI only.

‡ Valid in the batch interface only.

Label	Command	Parameters
		PRint <u>T</u> ABle
		CASE = Yes <u>N</u> o

† Valid in the Connect:Direct GUI only.

‡ Valid in the batch interface only.

Inquire Commands

Label	Command	Parameters
[optional]	INQUIRE CDPLEX	
[optional]	INQUIRE DEBUG	WHERE (SERVER = server name)
[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

Label	Command	Parameters
[optional]	SElect STATistics	WHERE (CCODE= (condition, completion code) EXCLUDE = (MEMBer MCR) (WTO) (NOTWTO) (list) MSGID = ID (list) PNAME = name (list) PNUMber = number (list) SERVER = server name 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 ARCHDSN = dsname (list))
		FILE PRint <u>TABLE</u> SUMmary
[optional]	SElect MeSsaGe	WHERE (IDMSGID=message ID) [Short Long](

Submit Command

Label	Command	Parameters
[optional]	SUBmit	PROC = member or DSN = dsn dsn(member) CASE = YES <u>NO</u> CLASS = n DEBUG = trace bits 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

Label	Command	Parameters
		...
		&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) - SOURCEIP=IP address - SESS.SNODE.MAX = (1- <u>255</u>) - LDNS=hostname - ENVIRONMENT=operating environment - LOGMODE=logmode entry name - APPLIDS=(vtam applid1 [,vtam 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 LU62 UDT33, - ALternate.LOGmode entry name=, - ALternate.USE.OUTbound= <u>Yes</u> No)) - CONTACT.NAME="name" - CONTACT.PHONE="phone information" - DESCRIPTION="description information" -))

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.TIMER = <u>5</u> nnnnn	5
CDPLEX.WLM.GOAL = (<u>NO</u> YES, exitname)	NO
CKPT = nK nM	none
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
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
CTCA.TIMER = number of seconds to wait when establishing a CTCA connection	180
DATEFORM = (<u>MDY</u> DMY YMD YDM)	MDY
DEBUG = nnnnnnnn	00000000
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
ESTAE = <u>YES</u> NO	YES
EXPDT = TT DD TD DT ALL <u>NONE</u> (TT,DD,TD,DT)	NONE

Global Initialization Parameters	Default Values
EXTENDED.RECOVERY = <u>NO</u> YES	NO
GDGALLOC = GENERATION <u>DSNAME</u>	DSNAME
GDGENQ = YES <u>NO</u>	NO
IMMEDIATE.SHUTDOWN = <u>I</u> R	I
INVOKE.ALLOC.EXIT = SEND <u>RECV</u> BOTH	RECV
INVOKE.ALLOC.EXIT.ON.RESTART = <u>NO</u> YES	NO
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 RA RH)	number of days = 0, HE
MAX.AGE.TOD = time	midnight and initialization
MAXBATCH = number of users	MAXUSERS value
MAXPRIMARY = number of primary sessions	6
MAXPROCESS = number of executing PNODE plus SNODE Processes	MAX-PRIMARY plus MAX-SECONDARY
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
NON.SWAPABLE = YES NO	none
PDSE.SHARING = YES <u>NO</u>	NO
PDSENQ = 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 = YES <u>NO</u>	NO
RESET.ORIGIN.ON.SUBMIT = YES <u>NO</u>	NO
REUSE.SESIONS = <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
SECURE.DSN = filename	none

Global Initialization Parameters	Default Values
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
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
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 size	100
STAT.SNODEID = (NO YES,NO YES)	(NO,YES)
STAT.SWITCH.SUBMIT = dsn [member]	none
STAT.SWITCH.TIME = (hh:mm:ss, ...)	none
STAT.TPPEC = (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 5	5
SUBMIT.EXIT = modulename	none
SYSOUT = class	none
TAPE.PREMOUNT = YES <u>NO</u>	NO
TAPEIO = <u>EXCP</u> BSAM	EXCP
TAPEMOUNT.EXIT = modname	none
TCP = OES <u>NO</u>	NO
TCP.ADDR=nnn.nnn.nnn.nnn xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx <u>ANYADDR</u> ANYADDR6 HOSTNAME	ANYADDR
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.PORTNUM = port number	none
TCP.FMH.TIMER=hh:mm:ss	0
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.NAME = name of started task ANYNAME , FAIL	TCPIP
TCP.PORTNUM = port number	1364
TCP.RUNTASK.TIMER =hh:mm:ss	0

Global Initialization Parameters	Default Values
TCP.SRC.PORTS = (ip.address,port-ranges),(ip.address2,port1,port2), - (ip.address3,port-ranges)	none
TCP.SRC.PORTS = (ip.address/submask,port-ranges), ...	
TCP.SRC.PORTS = (ip.address/0xFFFFFFFFFFFFFFFF,ports,ranges)	
TCP.SRC.PORTS.LIST.ITERATIONS = number of scans	none
TCP.TIMER = wait time	0
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 2	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
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)	4096, double the first parameter
WTMESSAGE = <u>NO</u> YES (YES, nnn)	NO
WTRETRIES = hh:mm:ss	00:03:00
XCF.NAME = XCF group name	none

System File Initialization Parameters	Default Values
APDSN = dsn	none
AUTHDSN = dsn	none
CKPTDSN = dsn	none
MSGDSN = dsn	none
NETDSN = dsn	none
STAT.ARCH.DIR = archive directory dsn	none
STAT.DSN.BASE = dsn base	none
STAT.FILE.PAIRS = number	
TYPEDSN = dsn	none

Local Initialization Parameters	Default Values
CDPLEX.MANAGER = <u>NO</u> YES	NO
CDPLEX.MSGID = <u>NONE</u> xx	NONE
CDPLEX.PLEXCLASSES = (*,plexclass, ..., plexclass)	*
CDPLEX.REDIRECT = 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 and or the first IPV6 address for the server.
CDPLEX.SERVER = Connect:Direct/Server name	none

Local Initialization Parameters	Default Values
CDPLEX.SERVER.JOBDSN = data set name	none
CDPLEX.SERVER.JOBMEM = ((member name, server name), ...)	none
CDPLEX.SERVER.NODE = nodename	none
CDPLEX.TCPIP = TCP/IP address Hostname	none
CDPLEX.TCPNAME = name of TCP/IP started task NAME,FAIL	TCPIP
CDPLEX.TIMER = <u>5</u> nn	5
CDPLEX.VTAM = (VTAM-APPL, P/S-Node-APPL)	none
CTCA = YES <u>NO</u>	NO
CTCA.TIMER = number of seconds to wait when establishing a CTCA connection	180
DEBUG = nnnnnnnn	00000000
ESTAE = <u>YES</u> NO	YES
MAXBATCH = number of users	MAXUSERS value
MAXPRIMARY = number of primary sessions	6
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
QUIESCE.NODE = nodename	none
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. The default port is 1363.
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.NAME = name of TCP/IP started task ANYNAME , FAIL	TCPIP
TCP.PORTNUM = port number	1364
TCP.TIMER = wait time	0
TCQ = <u>WARM</u> COLD	WARM
TRACE.BUFFER = nnn 2	2
UDT = YES <u>NO</u>	NO
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)	4096, double the first parameter size)

Status Codes

Code	Description
EX	Process executing
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
HS	Held for suspension
RA	Held for restart due to allocation error
RH	Held for restart
RS	Restarting
WA	Waiting acknowledgement
WC	Waiting connection
WR	Waiting restart
WS	Waiting designated start time
WT	Waiting transport protocol
WX	Waiting eligible Connect:Direct/Server

Connect:Direct 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	DMGCSUB trace	RADBDD08
00400000	TCQSH from DMCOPYRT	DMCOPYRT
00200000	Make each SVC dump unique	N/A
00040000	GETMAIN/FREEMAIN trace	RADBDD16
00008000	I/O buffer trace	RADBDD21
00004000	WTO all dynamic allocation parameters	RADBDD22
00002000	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
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

† 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.

The Modify command requests the same output as the DEBUG initialization parameter.

