

# **Connect:Direct® for i5/OS**

## **User's Guide**

**Version 3.6**

**Connect:Direct for i5/OS User's Guide**  
**Version 3.6**  
**First Edition**

(c) Copyright 1999-2006 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 for i5/OS 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

614/793-7000

---

# Contents

<b>Chapter 1</b>	<b>About Connect:Direct for i5/OS</b>	
	Issuing Connect:Direct CL Commands .....	7
	Issuing CL Commands Through the Main Menu .....	9
	Accessing the Administration Menu.....	11
<b>Chapter 2</b>	<b>Using Control Language</b>	
	Embedding CL Commands.....	15
	Entering CL Commands in Free Format.....	15
	Command Prompting.....	15
	Sample CL Program.....	17
	Accessing Command Level Help .....	17
	Prompting for a Command .....	18
	Accessing Parameter Level Help.....	18
	Using Parameter Level Prompt.....	18
<b>Chapter 3</b>	<b>Sending a File</b>	
	Native File System.....	19
	Integrated File System .....	19
	Integrated File System Types .....	20
	Specifying a Path Name .....	20
	Controlling the Character Set .....	20
	CDSND Command Example .....	21
	CDSND Command Format.....	21
	Required Parameters .....	23
	Optional Parameters.....	30
	Simulating Blocksizes.....	41
	Block Mode Transfer Examples .....	41
	Sending a File to a Remote Node .....	43
	Connect:Direct Send File Screen 1.....	43
	Connect:Direct Send File Screen 2.....	44
	Connect:Direct Send File Screen 3.....	44
	Connect:Direct Send File Screen 4.....	45
	Connect:Direct Send File Screen 5.....	45
<b>Chapter 4</b>	<b>Receiving a File</b>	
	Native File System.....	47
	Integrated File System .....	47
	Specifying a Path Name .....	48
	CDRCV Command Example.....	48

	CDRCV Command Format.....	49
	Required Parameters .....	50
	Optional Parameters .....	57
	Simulating Blocksizes .....	68
	Block Mode Transfer Examples.....	68
	Receiving a File From a Remote Node .....	70
	Connect:Direct Receive File (CDRCV) Screen 1.....	70
	Connect:Direct Receive File (CDRCV) Screen 2.....	71
	Connect:Direct Receive File (CDRCV) Screen 3.....	71
	Connect:Direct Receive File (CDRCV) Screen 4.....	72
	Connect:Direct Receive File (CDRCV) Screen 5.....	72
	Connect:Direct Receive File (CDRCV) Screen 6.....	73
<b>Chapter 5</b>	<b>Sending a Spool File</b>	
	CDSNDSPL Command Example .....	75
	CDSNDSPL Command Format .....	75
	Required Parameters .....	77
	Optional Parameters .....	83
<b>Chapter 6</b>	<b>Running a Remote Task</b>	
	CDRUNTASK Command Example .....	95
	CDRUNTASK Example With New TraceOpt Parameters.....	96
	CDRUNTASK Command Format .....	96
	Required Parameters .....	97
	Optional Parameters .....	98
	Initiating a Task on a Remote Node.....	101
<b>Chapter 7</b>	<b>Submitting a Job on a Remote Node</b>	
	CDRUNJOB Command Example.....	103
	CDRUNJOB Command Format .....	104
	Required Parameters .....	104
	Optional Parameters.....	105
	Submitting a Job to a Remote Node.....	108
<b>Chapter 8</b>	<b>Submitting a Connect:Direct Process on a Remote Node</b>	
	CDSUBMIT Command Example .....	111
	CDSUBMIT Command Format.....	111
	Required Parameters .....	112
	Optional Parameters.....	113
	Submitting a Process on a Remote Node .....	117
	Connect:Direct Submit Primary Screen.....	118
<b>Chapter 9</b>	<b>Compressing Files</b>	
	Required CDCOMP Tasks.....	119
	CDCOMP Command Example.....	119
	CDCOMP Command Format.....	120
	Required Parameters .....	120
	Optional Parameters.....	120
	Compressing a File Prior to Sending .....	122

<b>Chapter 10</b>	<b>Controlling Processes</b>	
	CDSUBPROC Command Example.....	126
	CDSUBPROC Command Format .....	126
	Required Parameter .....	127
	Optional Parameters .....	127
	Submitting a Process for Asynchronous Execution.....	129
	Selecting Specific Processes .....	130
	Viewing All Processes .....	130
	WRKCDPROC Command Example .....	130
	WRKCDPROC Command Format .....	131
	Required Parameters.....	131
	Optional Parameters .....	131
	Monitoring and Controlling Processes .....	132
	WRKCDPROC Option Descriptions.....	134
	Changing Process Characteristics.....	134
	Moving an Nonexecuting Asynchronous Process .....	134
	Removing an Nonexecuting Asynchronous Process .....	134
	Displaying Process Details .....	134
	Releasing a Process in the HOLD Queue .....	135
	Terminating and Flushing an Executing PNODE Process .....	135
	Interrupting an Executing Asynchronous Process.....	135
	Releasing or Holding Asynchronous Processes .....	135
	RLSCDPROC Screen .....	136
	hldcdproc Screen .....	136
<b>Chapter 11</b>	<b>Using Connect:Direct Statistics</b>	
	Specifying Selection Criteria .....	137
	CDSSELSTAT Command Example .....	137
	CDSSELSTAT Command Format.....	137
	Required Parameters.....	138
	Optional Parameters .....	138
	Selecting and Displaying Summary Statistics .....	140
	Selecting and Printing Detail Statistics.....	142
<b>Chapter 12</b>	<b>Decompressing Files</b>	
	CDDECOMP Command Example .....	145
	CDDECOMP Command Format .....	145
	Required Parameters.....	146
	Optional Parameters .....	146
	Decompressing a File.....	146
<b>Chapter 13</b>	<b>Displaying Connect:Direct Message Descriptions</b>	
	DSPCDMSGD Command Example .....	150
	DSPCDMSGD Command Format .....	150
	Displaying Connect:Direct Message Text .....	150
	Sample Message Description.....	151
<b>Appendix A</b>	<b>CL Program Examples</b>	
	i5/OS Compress/Decompress Examples .....	153
	i5/OS Block Mode Transfer Examples .....	153
	i5/OS Run Task Example .....	154
	i5/OS Send Example Using Send File Field Description .....	154

i5/OS Send and Receive Examples Using Native File System.....	154
i5/OS Send and Receive Examples Using Integrated File System .....	155
i5/OS–z/OS Send and Receive Examples .....	156
i5/OS–NT Send and Receive Examples.....	157
i5/OS–Windows Run Job and Run Task Examples .....	158
i5/OS–UNIX Send and Receive Examples .....	158
i5/OS–UNIX Run Task and Run Job Examples .....	160
i5/OS–UTIL Test Examples.....	160

---

# About Connect:Direct for i5/OS

Connect:Direct for i5/OS is an enterprise solution for secure, automated file delivery between business applications. Its session management and extended recovery features help assure data delivery in support of 24 hours a day, 7 days a week business operations. It links technologies, moves all types of information, and manages high-performance transfers by providing easy-to-use automation, checkpoint/restart, and error recovery. Connect:Direct offers choices in communication protocols, hardware platforms, and operating systems. It provides the flexibility to move information from mainframes to midrange systems to desktop and LAN-based workstations.

---

## Interfacing with Connect:Direct for i5/OS

When you install Connect:Direct for i5/OS, you create Control Language (CL) commands that enable you to use Connect:Direct. These CL commands enable you to send or receive data from other nodes running Connect:Direct. You can also use the CL commands to administer Connect:Direct. For example, you can use a CL command to display Connect:Direct message descriptions.

### Issuing Connect:Direct CL Commands

Issue Connect:Direct CL commands interactively from the command line or use them in a CL program. Results of CL commands that interact with a remote node are recorded in three places:

- ❖ Connect:Direct log
- ❖ User message queue of the local i5/OS node
- ❖ Statistics facility of the local and remote Connect:Direct node

Chapter 2, *Using Control Language*, provides information about CL commands, CL programming, and Connect:Direct CL commands.

---

## Understanding Connect:Direct for i5/OS Components

The Connect:Direct for i5/OS has several components that enhance its compatibility with other Connect:Direct platforms and run under the Connect:Direct subsystem:

- ❖ **Process Manager (PMGR)**

The PMGR initializes and manages the Connect:Direct environment. PMGR is always active when Connect:Direct for i5/OS is active.
- ❖ **Queue Manager (QMGR)**

The QMGR manages the Connect:Direct Process queue. The QMGR is always active when Connect:Direct for i5/OS is active.
- ❖ **Timer (TIMER)**

The TIMER component provides timer services to the session manager, the process queue manager, and other Connect:Direct for i5/OS components. The TIMER runs as a separate i5/OS job, submitted and controlled by the Process Manager component. The TIMER is always present when Connect:Direct for i5/OS is active.
- ❖ **Statistics Manager (STATMGR)**

Connect:Direct for i5/OS logs statistics information for all Processes. Each component of Connect:Direct for i5/OS generates statistics data to the STATMGR job. The long-running STATMGR job formats the statistics information and writes it to the statistics file. The Statistics Manager supports CL commands that provide the ability to schedule the creation, archival, and deletion of statistics file members. The STATMGR is always active when Connect:Direct for i5/OS is active.
- ❖ **Spool Monitor (SPLMON)**

The Connect:Direct SPLMON is a variable component of Connect:Direct for i5/OS. CDSPLMON uses Spool Monitor entries to associate output queues with destination Connect:Direct nodes.
- ❖ **TCP Listener (TCPLIST)**

The TCPLIST is a long-running job started and controlled by the Process Manager component. CDTCPPLIST processes incoming TCP/IP connection requests and requests session managers from the PMGR.
- ❖ **Session Manager (SMGR)**

The SMGR component executes the Connect:Direct Processes. The Process Manager starts all session managers except for SMGRs started by the LU6.2 evoke function. Evoked SMGRs request permission to run from the PMGR and continue processing.

---

## Connecting Connect:Direct for i5/OS to Connect:Direct/Plex

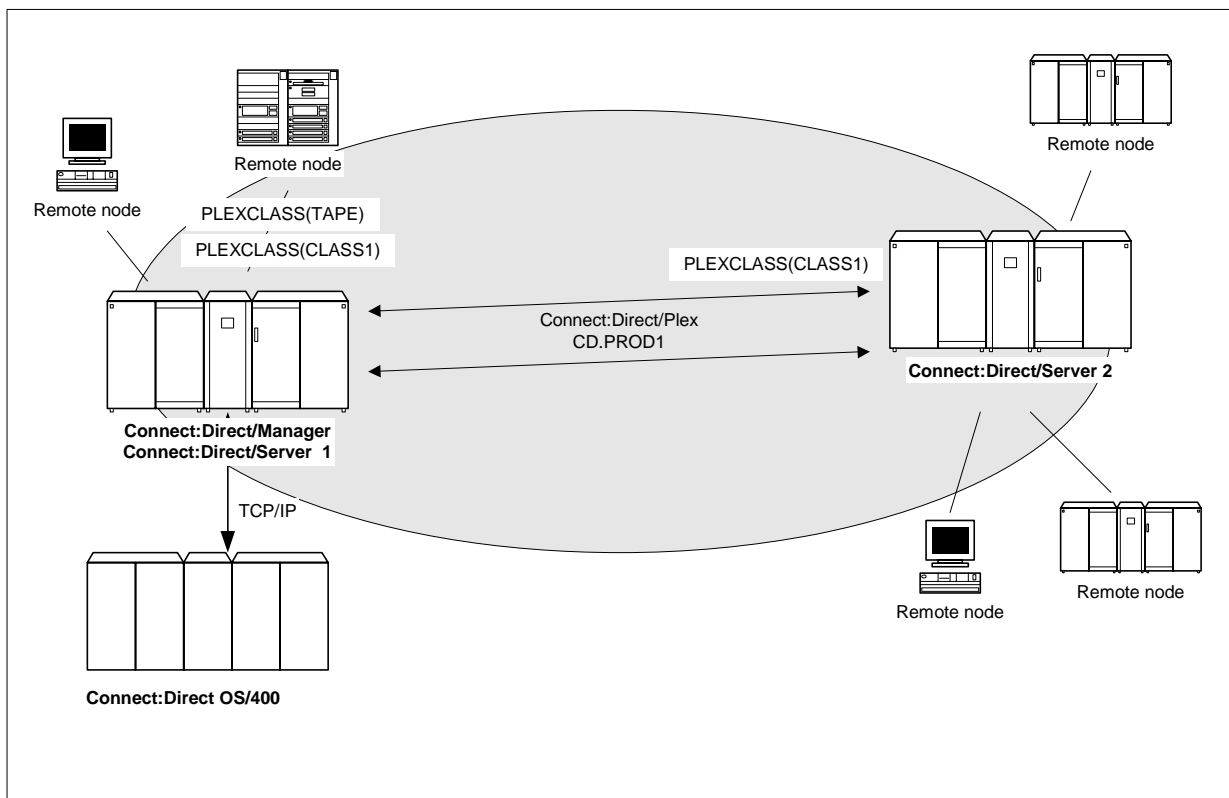
Connect:Direct for i5/OS can be connected to Connect:Direct/Plex, which is a Connect:Direct for z/OS system consisting of a Connect:Direct/Manager and one or more Connect:Direct/Servers in a TCP/IP environment. Once the connection between Connect:Direct for i5/OS and Connect:Direct/Plex has been defined, each system can initiate sessions to the other. All requests to initiate sessions to the Connect:Direct/Plex system go through the Connect:Direct/Manager, which controls the traffic and determines which Connect:Direct/Server can run a particular Process based on the PLEXCLASS of the Connect:Direct/Server and the Process coming from the Connect:Direct for i5/OS system. One advantage of routing the Processes through the Connect:Direct/Plex Manager is that you only have to define a single network map entry for the Connect:Direct/Plex Manager rather than creating separate node entries for each Connect:Direct/Plex Server.



The PLEXCLASS parameter represents a type of server in the Connect:Direct/Plex system. In the example illustrated in the graphic, there is one Connect:Direct for i5/OS system connected to a Connect:Direct/Plex system called CD.PROD1. CD.PROD1 consists of a Connect:Direct/Manager and two Connect:Direct/Servers. Both Connect:Direct/Servers support the CLASS1 PLEXCLASS but only Connect:Direct/Server 1 can process the TAPE PLEXCLASS.

When the Connect:Direct/Manager receives a Process that requires PLEXCLASS=CLASS1 and determines that both Connect:Direct/Servers support CLASS1, it routes the Process to the Connect:Direct/Server with the lowest workload.

For more information on the PLEXCLASS parameter, see the specific CL command you wish to execute in the Connect:Direct for z/OS environment in the *Connect:Direct for i5/OS User's Guide*.



## Accessing Connect:Direct Through the Main Menu

Connect:Direct for i5/OS includes an i5/OS system-style menu that allows you to execute Connect:Direct Command Language commands and related i5/OS CL commands from a menu. This menu consists of two screens.

### Issuing CL Commands Through the Main Menu

The first screen allows you to issue basic Connect:Direct commands. The second screen allows you to issue administration Connect:Direct commands. Each screen contains a full-function i5/OS command line and function key descriptions near the bottom of the screen.

To access the Main Menu:

1. Type **STRCDMNU** and press **Enter**. The Connect:Direct for i5/OS copyright screen is displayed.
2. Press **Enter**. The **Connect:Direct Main Menu** displays the Connect:Direct for i5/OS functions that you can perform from this screen.

```

CMDCD                      Connect:Direct (tm)                      System:  FRAN
Select one of the following:

    1. Send file to remote node                      CDSND
    2. Receive file from remote node                 CDRCV
    3. Send spooled file to remote node              CDSNDSPL
    4. RUNTASK on remote node                        CDRUNTASK
    5. RUNJOB on remote node                          CDRUNJOB
    6. SUBMIT process on remote node                  CDSUBMIT
    7. Submit local process                          CDSUBPROC
    8. Compress file                                  CDCOMP
    9. Decompress file                                CDDECOMP
    10. Display C:D Version and PTF level             CDVER

    11. Display Connect:Direct message description   DSPCDMSGD
    12. Work with Connect:Direct processes           WRKCDPROC
    13. Select Connect:Direct statistics             CDSELSTAT
                                                    More...

Selection or command

===> _____
_____
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT 1991, 2002 Sterling Commerce, Inc.

```

## Accessing the Administration Menu

Option 20 on the **Connect:Direct Main Menu** provides access to the following Administration Menu.

**Note:** To use the Connect:Direct administration commands, you must have administrative authority on the system.

```

CDADMIN                Connect:Direct Administration                System:  FRAN

Select one of the following:

Connect:Direct Configuration
  1. Work with Connect:Direct parameters                WRKCDPARM
  2. Work with Connect:Direct Netmap                    WRKCDNTMP
  3. Work with Connect:Direct users                    WRKCDUSR
  4. Work with Spool Monitor entries                    WRKSPLMONE
  5. Create Connect:Direct XTRAN table                  CRTCDXTTBL
  6. Secure+ Admin Utility                              SPADMIN

Connect:Direct Administration
  11. Start Connect:Direct subsystem                    STRCD
  12. End Connect:Direct subsystem                      ENDCD
  13. Work with Connect:Direct status                  WRKCDSTS
  14. Work with Connect:Direct processes                WRKCDPROC

                                                                More...

Selection or command

====> _____

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT 1991, 2002 Sterling Commerce, Inc.

```

The **Connect:Direct Administration** menu consists of two screens. The second screen is accessed by pressing the page down key.

```

CDADMIN                Connect:Direct (tm) Administration            System:  FRAN

Select one of the following:

Connect:Direct Statistics.
  21. Select Connect:Direct statistics                  CDSELSTAT
  22. Archive Connect:Direct stat member                CDARCSTATM
  23. Create Connect:Direct stat member                CDCRTSTATM
  24. Delete Connect:Direct stat member                CDDLSTATM
  25. Restore Connect:Direct stat member                CDRSTSTATM
  26. Display Connect:Direct stat information            CDDSPSTATI

Communications
  31. Vary communications device                        VRYCFG
  32. Configure TCP/IP                                 CFGTCP
  33. Start TCP/IP                                     STRTCP
  34. Work with TCP/IP status                          WRKTCPSTS

                                                                Bottom

Selection or command

====> _____

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

```



---

# Using Control Language

Connect:Direct for i5/OS uses Connect:Direct CL commands to perform Connect:Direct functions. This chapter provides the following information to help you use CL commands and programming:

- ❖ Understand the CL command format
- ❖ Reviewing Connect:Direct CL command functions
- ❖ Entering CL commands
- ❖ Monitoring CL program messages
- ❖ Using CL Programming and Connect:Direct CL commands
- ❖ Command prompting for the Connect:Direct CL commands

Refer to Appendix A, *CL Program Examples*, for member names of sample CL programs.

---

## Understanding the CL Command Format

Connect:Direct CL commands have the following general i5/OS CL command format:

- ❖ Each command can have required and optional parameters and subparameters that provide additional information needed to run the command.
- ❖ Subparameters are always positional.
- ❖ Each parameter provides a specific type of information or instruction.
- ❖ Each command is followed by one or more keywords and parameter pairs in the following format:

```
COMMAND keyword(parameter) keyword(parameter) ...
```

The following example shows a sample Connect:Direct CL command:

```
CDRCV  SNODE(DALLAS.OS390) SNODENVIRN(OS390)
       FDSN(DALLAS.ACCT.DATA)
       TDSN('ACCOUNT/DATA')
       TOSYSOPTS('TYPE(FILE)')
```

## Reviewing Connect:Direct CL Command Functions

The following table lists the Connect:Direct for i5/OS command names and their functions.

Command	Function
‡CDARCSTATM	Compress and delete a statistics file member other than the active one.
CDCOMP	Compress a file on the i5/OS node before sending it to a remote node.
‡CDCRTSTATM	Create a new statistics file member and log statistics to that new member.
CDDECOMP	Decompress files received from another node and archived on the i5/OS node as compressed files.
CDDEVACTG	Turns VARY on and off on devices used by Connect:Direct.
CDDLARCST	Deletes members from the archive statistics database file.
‡CDDLSTATM	Delete a specified statistics file member other than the active one.
CDDSPSTATI	Display information about where statistics are being logged.
CDRCV	Receive data from a remote node.
CDRSTSTATM	Decompress selected statistics file members to a user-designated file.
CDESELSTAT	Select statistics records and retrieve information about the active statistics file.
CDRUNJOB	Submit a job in batch mode on a remote node.
CDRUNTASK	Execute a Connect:Direct Run Task function on a remote node.
CDSND	Send data to a remote node.
CDSNDSPL	Send a spooled file to a remote node.
CDSUBMIT	Submit a Process to a remote node to initiate Connect:Direct work on the remote node.
CDSUBPROC	Submit Connect:Direct Processes to the process queue for asynchronous execution.
‡CDVER	Display the currently installed Connect:Direct version and Connect:Direct PTF level.
‡CNVNTMPF	Convert Connect:Direct network map file.
‡CNVUSRF	Convert Connect:Direct user file.
‡CRTCDOBJ	Create Connect:Direct objects.
‡CRTCDXTTBL	Create and update the translation table.
DSPCDMSGD	Display the text for the specified message ID.
‡ENDCD	End a Connect:Direct for i5/OS node and components.
HLDCDPROC	Identifies Processes to place in the Hold queue.
RLSCDPROC	Releases identified Processes from the Hold queue.
STRCDTCPL	Starts the Connect:Direct TCP listener.
UNINSTALLM	Uninstalls the latest Connect:Direct maintenance installation.
UPDATECD	Installs Connect:Direct maintenance.

‡ Administrative commands

‡ Connect:Direct Secure+ Option for i5/OS commands

Command	Function
†WRKCDPARM	View and change Connect:Direct initialization parameter values.
†‡SPADMIN	Secure+ Option Administrative utility to provide security enhancements.
†STRCD	Start Connect:Direct for i5/OS.
STRCDMNU	Start Connect:Direct Menu.
†WRKCDNTMP	Select remote node records and display attributes. This screen provides the options of copy or change to attributes of the remote node records.
WRKCDPROC	Monitor and control active and queued Processes.
†WRKCDSTS	View and control Connect:Direct job status.
†WRKCDUSR	Browse existing user records and invoke other user maintenance commands.
†WRKSPLMONE	View, change, or delete destination configuration file entries of the Spool Support for Connect:Direct for i5/OS.

† Administrative commands

†‡ Connect:Direct Secure+ Option for i5/OS commands

---

## Entering CL Commands

You can type CL commands on an i5/OS node using any of the following methods:

### Embedding CL Commands

You can embed CL commands in an application (user-written CL program), which allows you to combine functions supplied by Connect:Direct for i5/OS with i5/OS CL commands within a user-written application.

### Entering CL Commands in Free Format

You can type CL commands in a free format, which allows you to type a command and its parameters as a string of characters on a command line. This method is useful when typing familiar commands.

### Command Prompting

Command prompting guides you through typing commands and parameters.

---

**Note:** Connect:Direct for i5/OS commands are case sensitive if the information passed to the remote node is enclosed in single quotation marks.

---

---

## Monitoring CL Program Messages

After issuing Connect:Direct for i5/OS commands such as, CDSND, CDRCV, CDRUNTASK, and CDRUNJOB, use the MONMSG command to monitor the following categories of error messages using a CL program:

Message Category	Message ID	Severity	Other
Warning	CDE0107	10	*ESCAPE
Unrecoverable error	CDE0108	30	*ESCAPE
Unexpected error	CDE0000 (for example, CDE0301)		

---

**Note:** You cannot use MONMSG to monitor CDS0108 (Process execution ended) because it is not sent as an error message; it is an informational message.

---

To capture the Connect:Direct error message after the MONMSG code, add the following after the MONMSG command:

```
RCVMSG MSGTYPE(*LAST) WAIT(1) RMV(*NO) MSGID(&MSGID)MSGDATA(&MSGDTA) SNDMSG
MSG(&MSGID || ' ' || &MSGDATA) TOUSR(*SYSOPR)
```

&MSGID is the CDE\* error message. The first 8 characters of the &MSGDATA are the application error message that you can look up interactively with the following command:

```
DSPCDMSGD <msgid>
```

---

**Note:** The &MSGID and &MSGDATA variables must be defined in the CL program. Specify the &MSGID character length as 7 characters and &MSGDATA as 8 characters to capture the error message.

---



---

## Using CL Programming and Connect:Direct CL Commands

An i5/OS CL program includes Connect:Direct CL commands as source statements. You can synchronize the processing of one or more Connect:Direct operations by including Connect:Direct CL commands in a CL program.

---

**Note:** If you are using the Source Edit Utility (SEU) to edit a CL program, the prompt facility can help you type all CL commands, including Connect:Direct CL commands.

---



## Sample CL Program

The following figure shows a sample CL program with multiple CL commands:

```

PGM          PARM(&LIBNAME &FILENAME &MBRNAME)

DCL          VAR(&LIBNAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&FILENAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&MBRNAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&FDSN) TYPE(*CHAR) LEN(32)

/*  BUILD THE FDSN VARIABLE                                     */
      CHGVAR  VAR(&FDSN) VALUE(&LIBNAME *TCAT '/' *CAT +
                                &FILENAME *TCAT '(' *CAT &MBRNAME *TCAT +
                                ')')

/*  SEND THE FILE MEMBER  */
      CDSND   SNODE('CDSERVER.V4100') +
              SNODENVIRN(WINDOWS) FDSN(&FDSN) +
              TDSN('Send_Test.TXT') +
              FMSYSOPTS('TYPE(MBR)') +
              TOSYSOPTS('DATATYPE(TEXT)')
      MONMSG  MSGID(CDE0000) EXEC(GOTO CMDLBL(ERROR))

      GOTO END

ERROR:      SNDMSG  MSG('The process erred') TOUSR(*SYSOPR)

END:       ENDPGM

```

---

## Command Prompting for the Connect:Direct CL Commands

Connect:Direct CL commands follow the format of the native i5/OS CL commands, which provide an extensive help facility describing their purpose, parameters, values, and any dependencies.

In addition, you can use the prompt/help feature to select and type commands and appropriate parameters. Prompt/help is offered at the command level and at the parameter level. Both levels allow you to use **F4** for prompting and **F1** for help. You can access screen level help by moving the cursor off an entry or field and pressing **F1**.

### Accessing Command Level Help

Complete the following steps to access the Help text:

1. For help with commands, type the command on the command line and press **F1**. Help text for the command displays and describes the function of the command with a description of each parameter and possible values.
2. After viewing help information, press **F3** to exit the help function or press **F12** to cancel and return to the command line.

## Prompting for a Command

Complete the following steps to prompt for a command:

1. Type the command on the command line and press **F4** to display an entry screen for the command. This screen lists valid parameters and provides a brief description of each parameter.
2. Press **F11** to display the keyword names.
3. Complete the command by filling in the parameter values on the screen.
4. After typing in all applicable parameters, press **Enter** to execute the command.

## Accessing Parameter Level Help

Complete the following steps to access help with parameters:

1. Place the cursor by the parameter for which you want further description.
2. Press **F1** to display a description of the parameter and valid values.
3. After viewing help information, press **F3** to exit the help function or press **F12** to cancel and return to the command line.

## Using Parameter Level Prompt

Complete the following steps to prompt for a parameter:

1. Place the cursor by the parameter for which you want to prompt.
2. Press **F4** for a list of valid values for the parameter. At this screen, type in the appropriate value for the parameter.
3. Press **Enter** to set the parameter and return to the original screen. You can also press **F3** or **F12** to exit the help function.

If a parameter is required to run the command successfully, the field is highlighted. Prompt does not provide the default value. A greater than (>) symbol indicates that the parameter field contains a user-supplied value.

---

# Sending a File

Sending a file on Connect:Direct for i5/OS requires the Connect:Direct Send (CDSND) command. Understanding the Connect:Direct for i5/OS file systems, the CDSND command parameters, block mode transfer, and the CDSND screens enables you to use this command more effectively. This chapter provides the following information to help you use this command:

- ❖ Using the file systems
- ❖ Using the CDSND command
- ❖ Using block mode transfer
- ❖ Using the CDSND screens

---

## Using the File Systems

Connect:Direct for i5/OS supports two types of file systems to perform file transfers: native file and integrated file. The native file system and the integrated file system can both be used to send files to and receive files from any supported file system.

### Native File System

The native file system used in previous releases of Connect:Direct for i5/OS remains functional. This file system accepts the file name in the following formats:

**library-name/file-name**

specifies the library and file name to be copied.

**library-name/file-name(member-name)**

specifies the library, file, and member name of the member to be copied. You cannot use the EXCLUDE and SELECT parameters if you specify a member name.

---

**Note:** \*LIBL, \*CURLIB and QTEMP are not valid library names for a Connect:Direct Process.

---

### Integrated File System

As part of i5/OS, the integrated file system provides an integrated structure over all data stored in the iSeries system. This file system supports stream input/output and storage management like personal computers and UNIX operating systems.

## Integrated File System Types

The Connect:Direct for i5/OS integrated file system provides application and data compatibility with other platforms. The integrated file system consists of five file system types:

- ❖ /QLANSrv is a LAN server file system
- ❖ /QDLS is a document library file system
- ❖ /QSYS.LIB is the iSeries DB2 file system
- ❖ /QOpenSys is a UNIX-style file system
- ❖ All others are part of the root file system

## Specifying a Path Name

The integrated file system recognizes the objects by the path specification. You must specify the fully qualified path name to transfer a file. Connect:Direct for i5/OS identifies a path that starts with a slash (/) or a backslash (\) as a request for the integrated file system.

The integrated file system does not support file field description. You cannot specify the SNDFFD(\*YES) under this file system. Similarly, you cannot specify SELECT or EXCLUDE with the integrated file system.

To use this system, you must specify the fully qualified path name in one of the following formats:

**/QSYS.LIB/library-name.LIB/file-name.FILE/member-name.MBR**

specifies the library, file name, and member name to be copied using the integrated file system.

**/directory/file-name|/QLANSrv/file-name|/QDLS/folder-name|/QOpenSys/file-name**

specifies the directory, folder, and file names to be copied using the integrated file system.

Directories like /QOpenSys and /root are case-sensitive. The /QSYS.LIB notation is case-sensitive only if you enclose the name in single quotation marks.

## Controlling the Character Set

You can control the coded character set ID used by the integrated file system. To specify a coded character set ID other than the job-coded or the system-coded character set, set the SYSOPTS parameter called CCSID to the character set you want to use, for example, CCSID(819).

If this parameter is not set, the system default for CCSID is used.

---

## Using the CDSND Command

Use the CDSND command to send a file to a remote node. This command sends files to any file systems supported by the i5/OS V3R1M0 or later. You can also specify a user-exit command (either **EXITCMD** or **FAILCMD**) to be executed after the completion of a copy process in either FROMSYSOP or TOSYSOP. Using EXITCMD, you can specify a command to be executed only if the copy process is successful. You can also specify a command to be executed only if the copy process is not successful by using the FAILCMD command. The EXITCMD and FAILCMD commands must be executed on an i5/OS using Connect:Direct release 3.3.00 and above. The FAILCMD and EXITCMD are executable iSeries commands as well as Connect:Direct as long as the size of the command does not exceed 127 bytes.

## CDSND Command Example

The command shown in the following figure is an example of the CDSND command.

```
CDSND  PNAME ( CDSEND )  SNODE ( DALLAS . OS3901 )
      SNODENVIRN ( OS390 )
      FDSN ( ' DALLIB / ACCTDATA ' )
      TDSN ( DALLAS . ACCTDATA )
      FMSYSOPTS ( ' TYPE ( FILE ) ' )
```

## CDSND Command Format

The following table shows the parameters for the CDSND command. Required parameters are in bold.

Command	Parameters
CDSND	<b>SNODE(remote-node-name)</b>
	<b>SNODENVIRN(remote-environment-name)</b>
	<b>FDSN('from data-set-name')</b>
	<b>TDSN('to data-set-name')</b>
	<b>FMSYSOPTS('</b> PRECMPR(*YES *NO) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command) SNDDFFD(*YES *NO) TEXTFILE(*YES *NO) <b>TYPE(FILE MBR OBJ)</b> XTRAN(table-name)[XTRANLSO(so-code) XTRANPAD(pc)] XTRANLSI(si-code) XTRANLDATA(MIXED DBCS) CODEPAGE(nnnnn,nnnnn)]
	<b>TOSYSOPTS('</b> DECMRPR(*YES *NO) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command) <b>TYPE(FILE MBR OBJ SPLF)</b> CODEPAGE(nnnnn,nnnnn) CCSID(nnnnn) EORCHAR(xxxx) ALWNULL(Y) ')
	<b>BLKSIZE(n *CALC)</b>
	<b>CKPTINV(n)</b>
	<b>CLASS(n)</b>
	<b>COMPRESS(*NO *YES *EXT X xx C cc)</b>
<b>CRC(*INITPARMS *NETMAP *OFF *ON)</b>	
<b>ECTUNING(ecclevel,ecwsize,ecmlevel)</b>	
<b>EXCLUDE(generic member-name (start-range stop-range) 'list')</b>	

† Connect:Direct Secure+ Option parameter.

Command	Parameters
	FDISP(current, normal end, abnormal end)
	NOTIFY(userid)
	PLEXCLASS(user-defined class)
	PNAME(process-name)
	PNODEID(id, pswd, newpswd)
	REPLACE(*YES *NO)
	RMTWORK(*NO *YES)
	SACCT(accounting-data)
	†SECOPTIONS(* *)
	SELECT(member,new-name,R NR,'list')
	SNODEID(id, pswd, newpswd)
	TBUFND(n)
	TDBPARMS(dbid, dbsubid)
	TDCB(modeldcb, blksize, den, dsorg, keylen, limct, lrecl, optcd, recfm, rkp, trtch)
	TDISP(current, normal end, abnormal end)
	TIOEXIT(program-name)
	TIOPARM(parameter-names)
	TLABEL(seqno,labeltype,pswdread,inout,retpd expdt)
	TLINK(vmuserid,pwd,accmode,uu)
	TMSVGP(data-definition name)
	TPROTCT(*YES *NO)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS *CURRENT Command Trace Options: L,S,F,C,X,K,D,H,V,P *ALL *EXT *COMMON  *NONE)
	TSPACE(avgbk, type, primary, secondary, pdsbk, rlse, contig, round)
	TSQL('sql-statement' DSN=data-set-name')
	TTYPE(entry-name)
	TUNIT(file-address)
	TVOL(private, volcount, ser)
	TVOLREF(data-set-volume)
	TVSAMCAT(dsn,vmid,pwd,accmode,uu)

† Connect:Direct Secure+ Option parameter.

**Note:** The CLASS parameter is not used for Connect:Direct for i5/OS.

## Required Parameters

Unless otherwise noted, these required parameters apply to all remote node environments.

### **SNODE(remote-node-name)**

specifies a 1–16 character alphanumeric name for the receiving node (remote node). The node name must be defined in the network map.

---

**Note:** The name parameter is case sensitive.

---

### **SNODENVIRN(remote-environment-name)**

specifies the 1–10 character alphanumeric name for the receiving node environment: OS400, OS390, MVS, MSP, OPENVMS, TANDEM, UNIX, VM, VOS, VSE, Windows, or Windows\_NT.

### **FDSN('from data-set-name')**

specifies the file to be sent to the remote node. The file name can be up to 256 characters long. File names are verified based on the i5/OS standard file name conventions. The entire data set name must be in single quotes. If you are using the Connect:Direct for i5/OS **F4** prompting feature, the system supplies the single quotes if you do not type them with the data set name on the command line.

The length of the input field displayed is 132 characters, by default. To increase the length of the field to 256 characters, type an ampersand(&) followed by a blank in the first two positions of the field.

---

**Note:** The values of \*LIBL, \*CURLIB, and QTEMP cannot be used when sending to or from an iSeries system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the iSeries.

---

Connect:Direct for i5/OS supports two file systems: native file system and integrated file system. Refer to the *Using the File Systems* on page 19 for detailed information about the file systems.

### **TDSN('to data-set-name')**

specifies the receiving file name on the remote node. The file name can be up to 256 characters long. The length of the input field displayed is 132 characters, by default. To increase the length of the field to 256 characters, type an ampersand(&) followed by a blank in the first two positions of the field.

---

**Note:** Do not use member names longer than eight characters when sending to z/OS, VSE or MSP nodes. The results are unpredictable.  
For TDSN, the values of \*LIBL, \*CURLIB, QTEMP cannot be used when sending to or from an iSeries system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the iSeries.

---

TDSN specifies the destination file name. File names are verified based on the i5/OS standard file name conventions. The entire DSN must be in single quotes. When you are using the Connect:Direct for i5/OS **F4** prompting feature, the system supplies the single quotes if you do not type them with the file name on the command line.

Connect:Direct for i5/OS supports both the native file system and the integrated file system. The file system description begins on page 19. Refer to the FDSN parameter format on page 23 for TDSN format information.

---

**Note:** If the TDSN requires a password, you can specify the password in the TDSN after the data set name. A slash (/) must follow the data set name and precede the password. This password is used at data set allocation. If it is not correct, the system issues a Write to Operator Request (WTOR) which requests the password when Connect:Direct opens the data set.

---

The following table summarizes the data set name requirements for Connect:Direct platforms.

SNODENVIRN	TDSN	Format	Special Considerations
OS/400	Native File System 'library-name/file-name' 'library-name/file-name (member-name)'		Enclose in single quotation marks.
	Integrated File System '/QSYS.LIB/library-name.LIB /file-name.FILE /member-name.MBR' or '/directory/file-name /QLANSrv/file-name  /QDLS/folder-name /QOpenSys/file-name'		You cannot specify the *SNDDFD, *SELECT, and *EXCLUDE parameters with the integrated file system.
	Spool File Support spooled-file-name		Do not use member names longer than eight characters when sending to z/OS, VSE, and MSP. The results are unpredictable.
OS/390, VSE, MSP, MVS	Standard z/OS DSN naming conventions.		Single quotes allow special characters. Do not use member names longer than eight characters. The results are unpredictable.
VM	Fully qualified file-name		
Tandem	\system.\$volume.subvolume.filename		
Windows	drive:\directory\subdirectory\filename or \\servername\sharepointname\subdirectory\fil ename		
UNIX	/directory/subdirectory/filename		
OpenVMS	node::device:[directory]filename.type;version		When you are copying a file, the file specification must be enclosed in single quotation marks if it uses spaces or other special characters not recognized by the other node.
Stratus VOS	Fully qualified path and file name		Enclose the file name in double quotation marks when you use a reserved word (statement name or keyword) for the file name.

#### FMSYSOPTS(

PRECMR(\*YES | \*NO) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command)  
 SNDDFD(\*YES | \*NO) TEXTFILE(\*YES | \*NO) TYPE(FILE | MBR | OBJ)  
 XTRAN(table-name) CODEPAGE(nnnnn,nnnnn)[XTRANLSO(so-code) | XTRANLSI(si-code |  
 XTRANLDATA(MIXED | DBCS)|XTRANPAD(pc)] CCSID(nnnnn) EORCHAR(yyyy)  
 ALWNULL(Y) specifies sending system operations. If you omit this parameter, Connect:Direct  
 for i5/OS assumes 'TYPE(FILE)'. The maximum number of characters for FMSYSOPTS is 2K.

**Note:** All FMSYSOPTS keyword values must be enclosed in parentheses. The entire  
 FMSYSOPTS string must be enclosed in single quotes, for example,  
 FMSYSOPTS('TYPE(FILE) SNDDFD(\*YES)').



**PRECMPR(\*YES|\*NO)** specifies that the file being sent has previously been compressed using the CDCOMP command. This option PRECMPR(\*YES) cannot be specified if a secured copy function is being performed using Secure+ Option. To send a file that has been compressed, you must specify the PRECMPR(\*YES) parameter with the CDSND command. For information about how to compress files, refer to Chapter 9, *Compressing Files*.

---

**Note:** Ensure that the output file contains no members before you execute CDCOMP.

---

**EXITCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is successful.

**FAILCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is not successful.

**SNDFFD(\*YES|\*NO)** specifies whether the sending node transfers file field descriptions.

Use the following formula to determine whether a file is sent with the file field descriptions. If the bytes required value is greater than 12K, you cannot transfer the file field descriptions.

$12 \times \text{number of keys} + (45 \times \text{number of fields}) + \text{SYSOPTS} = \text{bytes required}$
--

---

**Note:** You cannot use this parameter with the integrated file system.

---

**TEXTFILE(\*YES|\*NO)** specifies that the file being sent is a text file. This keyword is for the i5/OS only.

**TYPE(FILE|MBR|OBJ)** specifies the type of data to be sent. You *must* define the type of data to be copied as one of the following:

**FILE** specifies that the file being copied is a physical database file.

**MBR** specifies that the object being copied is a member of a physical database file.

**OBJ** specifies that the data to be copied is in save file format.

**XTRAN(table-name)** specifies the translation table to use. The named table object must exist in a library that is in the library list of the session manager job. If the library name is not in the list, the CDSND command fails.

If the XTRAN keyword is present, then the following related optional keywords can also be used:

**XTRANLSO (so-code)** specifies extended translation local shift-out code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-out (SO) character and overrides the default value of 0E. You can specify any two valid hexadecimal digits.

**XTRANLSI (si-code)** specifies extended translation local shift-in code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-in (SI) character and overrides the default value of 0F. You can specify any two valid hexadecimal digits.

**XTRANLDATA (MIXED|DBCS)** specifies extended translation local data format. This keyword specifies the format of the data on the local (i5/OS side) of the COPY.

**MIXED** indicates that the data can contain both double-byte character set (DBCS) and single-byte character set (SBCS) characters and that SO/SI characters are used. MIXED is the default.

**XTRANPAD(pc)** specifies that padding characters are in use. When DBCS data is translated from EBCDIC to ASCII, PAD specifies that the SHIFT-OUT and SHIFT-IN characters are replaced by a pad character to allow the displacement of fields within a record to remain unchanged during translation.

When DBCS data is translated from ASCII to EBCDIC, PAD specifies that the input ASCII DBCS file is in a padded format. The character immediately preceding a DBCS character or string is overlaid by the SHIFT-OUT character. The character immediately following a DBCS character or string is overlaid with the SHIFT-IN character. This allows the displacement of fields within a record to remain unchanged during translation.

**pc** is the pad character to be used during EBCDIC to ASCII translation. **pc** is ignored for ASCII to EBCDIC translations.

If XTRANPAD is not used, the SI/SO characters are stripped from the file when transferring from EBCDIC to ASCII, and added to the records when sending from ASCII to EBCDIC.

**DBCS** indicates that the data is pure DBCS characters and that no SO/SI characters are used.

The following rules apply to the use of the XTRAN keyword:

- ❖ You must specify the XTRAN keyword to use extended translation; all other keywords are optional.
- ❖ The default for local shift-out is the IBM standard x0E.
- ❖ The default for local shift-in is the IBM standard x0F.
- ❖ The default local data format is MIXED with SO/SI in use. XTRAN is *not* allowed with PRECMPR(\*YES) or DECMPR(\*NO)
- ❖ If you send a file to an ASCII platform using extended translation, you *must* specify in the TOSYSOPTS that the data type is binary. Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for platform-specific SYSOPTS parameters.

**CODEPAGE(nnnnn,nnnn)** converts data from one code page to another. The supported conversions are identified on the IBM Web site in the V5R1 Information Center, DB2 Universal Database for iSeries SQL Reference, Appendix E. CODEPAGE() on Connect:Direct for i5/OS relates to CODEPAGE() developed for the z/OS system, UNIX, and Windows.

**nnnnn** identifies the IBM CCSID codepage number to convert from and the IBM CCSID code page number to convert to. If the FROM code page or TO code page is locale, the code page for the locale where CONNECT:Direct is executing is used. For example, if `fmsysopts(codepage(locale,1208))`, then the FROM code page is the code page for the locale and the TO code page is UTF-8.

**ALWNULL(Y)** specifies if NULLS characters will be processed when sending DB2 files.

**CCSID(nnnnn)** specifies the coded character set ID to use when creating IFS files. If no character set ID is specified, the system default is used.

**EORCHAR(xxxx)** specifies the two-to four-digit hexadecimal symbol to identify the character to use in text files as the end-of-record character, for example, EORCHAR(0A).

**TOSYSOPTS(\*DECMPR(\*YES|\*NO) TYPE(FILE|MBR|OBJ|SPLF) EXITCMD(valid i5/OS Command) FAILCMD(valid i5/OS Command) CCSID(nnnnn) EORCHAR(yyyy) )**

specifies system operation parameters at the receiving node. The maximum number of characters permitted for TOSYSOPTS is 256. The format of CODEPAGE is based on the remote operating system default values for CDSND.

For additional information about valid platform-specific SYSOPTS parameters, see the appropriate COPY statement section of the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

**EXITCMD(valid i5/OS i5/OS command)** specifies a command to be executed only if the copy process is successful.

**FAILCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is not successful.

**DECMPR(\*YES|\*NO)** specifies whether the receiving Connect:Direct node is to decompress the received file. This parameter is valid only when the receiving system is a Connect:Direct for i5/OS node. The EXITCMD and FAILCMD are valid here if the receiving node is i5/OS 3.3.00 or higher. This option DECMPR(\*NO) cannot be specified if a secured copy function is being performed using Secure+ Option.

---

**Note:** Ensure that the destination file contains no members before you send a file with DECMPR(\*NO) specified.

---

**\*NO** instructs the receiving system to place the received data in a database file without decompressing it.

**CCSID(nnnnn)** specifies the coded character set ID to use when creating IFS files. If no character set ID is specified, the system default is used.

**EORCHAR(xxxx)** specifies the two-to four-digit hexadecimal symbol to identify the character to use in text files as the end-of-record character, for example, EORCHAR(0A).

**CODEPAGE(nnnnn,nnnnn)** converts data from one code page to another. The supported conversions are identified on the IBM Web site in the V5R1 Information Center, DB2 Universal Database for iSeries SQL Reference, Appendix E. CODEPAGE() on Connect:Direct for i5/OS relates to CODEPAGE() developed for the z/OS system, UNIX, and Windows.

**nnnnn** identifies the IBM CCSID code page number to convert from and the IBM CCSID code page number to convert to. If the FROM code page or TO code page is locale, the code page for the locale where CONNECT:Direct is executing is used. For example, if `tosysopt(codepage(1208,locale))`, then the FROM codepage is UTF-8 code page and the TO code page is the local code page.

**TYPE(FILE|MBR|OBJ|SPLF)** specifies the type of data to be sent. Define the type of data to be sent as one of the following:

**FILE** specifies that the file being copied is a physical database file.

**MBR** specifies that the object being copied is a member of a physical database file.

**OBJ** specifies that the data to be copied is in save file format.

**SPLF** specifies that the data to be copied is an i5/OS spooled output file.

For each type of file, you can specify certain additional TOSYSOPTS parameters.

If the receiving node is a Connect:Direct for i5/OS node, the following TOSYSOPTS parameters are valid for **TYPE(FILE)** and **TYPE(MBR)** files:

**RCDLEN(record-length)** specifies the data length for each record in the file.

**FILETYPE(\*SRC)\*DATA)** specifies the type of file to be created.

**TEXT('text description')** specifies a text description to be associated with this member (and file, if created). This description cannot exceed 50 characters and must be enclosed in single quotes.

**EXPDATE(expiration-date)** specifies the date after which the new or replaced member cannot be used. If EXPDATE is not specified, then the file does not have an expiration date.

**MAXMBRS(number)\*NOMAX)** specifies the maximum number of members a physical file can contain.

**SIZE(#-of-recs|incr-value #-of-incrs)\*NOMAX)** is used when a new file is created for the file received.

**AUT(\*CHANGE)\*ALL)\*USE)\*EXCLUDE)** specifies the authority to be given to a user who does not have specific authority to the file or member, is not on the authorization list, and whose user group does not have specific authority to the file or member.

**IGCDTA(\*YES)\*NO)** specifies that DBCS support is installed on the destination i5/OS node and the file being sent contains DBCS data. The following TOSYSOPTS parameters are valid for **TYPE(OBJ)** files:

**MAXRCDS(number)\*NOMAX)** specifies the maximum number of records the save file (which was created to hold the data received) can reach.

**ASP(auxiliary-storage-pool)** specifies the auxiliary storage pool from which the system allocates storage for the save file.

**TEXT('text description')** specifies a text description to be associated with this object. This description cannot exceed 50 characters and must be enclosed in single quotes.

**AUT(\*EXCLUDE)\*CHANGE)\*ALL)\*USE)** specifies the authority to give to a user who does not have specific authority to the object, is not on the authorization list, and whose user group does not have specific authority to the object.

The following TOSYSOPTS parameters are valid for **TYPE(SPLF)** files:

**DEV(\*JOB)\*SYSVAL|device-name)** specifies the name of the printer device description.

**DEVTYPE(\*IPDS)\*SCS)\*USERASCII)** specifies the type of data stream created for a printer file from the data received.

**PAGESIZE(page-length page-width)** specifies the length and width of the page used by the printer. This parameter overrides the FORMTYPE parameter.

**LPI(3|4|6|7.5|8|9)** specifies the line space setting (lines per inch) on the printer.

**CPI(5|10|12|13.3|15|16.7|18|20)** specifies the printer character density, in characters per inch, for the printer.

**OVRFLW(overflow-line-number)** specifies the line number on the page where overflow to a new page begins. The value specified must not exceed the form length specified for PAGESIZE.

**FOLD(\*NO)\*YES)** specifies whether entire records are printed when the record length exceeds the form width.

**RPLUNPRT(\*YES 'replacement-character')\*NO)** specifies whether unprintable characters are replaced with printable characters when printed. It also specifies the replacement character.

**ALIGN(\*NO|\*YES)** specifies whether the page must be aligned in the printer before printing is started.

**CTLCHAR(\*NONE|\*FCFC)** specifies whether the data contains printer control characters.

**CHLVAL(\*NORMAL| (channel#1 line#1) ... (channel#12 line#12))** specifies the list of channel numbers with their assigned line numbers. Note that CTLCHAR(\*FCFC) must be specified as a SYSOPTS parameter for this to be a valid parameter.

**FORMFEED(\*DEV|\*CONT|\*CUT|\*AUTOCUT)** specifies the form feed attachments used by the printer (4214, 5219, and 5553 printers only).

**PRTQTY(\*STD|\*DRAFT|\*DEV|\*NLQ)** specifies the quality of print produced.

**DRAWER(1|2|3|\*E1)** specifies the source drawer to be used when automatic cut-sheet feed mode is used. FORMFEED(\*AUTOCUT) must be specified as a SYSOPTS parameter for this to be a valid parameter.

**FONT(\*CPI|\*DEV|font-identifier)** specifies the font identifier to be used for the spooled output file.

**CHRID(\*DEV|\*SYSVAL| graphic-character-set code-page)** specifies the character identifier to be used for the spooled output file. This parameter allows you to print data with different character identifier coding.

**PAGRTT(\*DEV|\*COR|0|90|180|270)** specifies the degree of rotation (clockwise from the edge of the paper first loaded into the printer) of text on each page printed. 0|90|180|270 indicates the specific angle of rotation.

**PRTTXT('print-text')** specifies a line of text to be printed at the bottom of each page printed. Up to 30 characters enclosed in single quotes is specified.

**JUSTIFY(0|50|100)** controls the print positions of the characters on the page (in the spooled file) so the right margin is regular.

**DUPLEX(\*NO|\*YES|\*TUMBLE)** specifies whether the spooled output file is printed on one or both sides of the paper.

**SPOOL(\*YES|\*NO)** specifies whether the data is sent to a spooled file prior to printing.

**OUTQ(\*JOB|\*DEV|library-name/output-queue- name)** specifies the output queue for the spooled output file created. This parameter is valid only when SPOOL(\*YES).

**FORMTYPE(form-type)** specifies the type of form to use in the printer when the spooled file is printed. A form-type identifier is user-defined and is no longer than 10 characters. The PAGESIZE parameter overrides this parameter.

**COPIES(number-of-copies)** specifies the number of copies to be printed. Valid values range from 1–255. This parameter is valid only when SPOOL(\*YES).

**MAXRCDS(maximum-records)** specifies the maximum number of records that is placed in the spooled file. Valid values range from 1–500000. This parameter is valid only when SPOOL(\*YES).

**FILESEP(number-of-file-separators)** specifies the number of blank separator pages to be placed between each copy of the file printed.

**HOLD(\*YES|\*NO)** specifies whether the file is to be held on the output queue until released by the user. This parameter is valid only if SPOOL(\*YES).

**SAVE(\*YES|\*NO)** specifies whether the spooled output file is to be saved on the output queue once printed. This parameter is valid only if SPOOL(\*YES).

**OUTPTY(\*JOB|output-priority)** specifies the scheduling priority of the file on the output queue. This parameter is valid only if SPOOL(\*YES).

**USRDTA(user-data)** allows up to 10 characters of data to identify the spooled output file. This parameter is valid only if SPOOL(\*YES).

Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for general information on SMS support and TOSYSOPTS parameters for other platforms.

## Optional Parameters

The following parameters, listed alphabetically, are optional. Unless otherwise noted, the parameters apply to all remote node environments.

### **BLKSIZE(n)\*CALC**

specifies the blocksize value for the source file. This value must be a multiple of the file record size.

**n** specifies the block size of the file. This value must be a multiple of the record size. The valid range is 1–32720

**\*CALC** indicates that the blocksize is calculated.

### **CKPTINV(n)**

specifies the number of bytes, from 10 K to 2 gigabytes, to send before taking a checkpoint. The format is *nnnnnnn*, *nnnnnnK*, or *nnnnnnM*, where **K** specifies thousands of bytes and **M** specifies millions of bytes. A value of **0** specifies no checkpoint/restart. Connect:Direct for i5/OS uses the value you specify, rounded up to the nearest record boundary, to determine when to take a checkpoint.

A checkpoint interval specified here overrides the system-wide default value in the initialization parameters file.

Checkpointing cannot occur in the following cases, even if you specify a checkpoint interval:

- ❖ The file being sent is compressed during the send operation by the local node using the extended compression feature, and decompression is deferred on the receiving node. That is, the TOSYSOPTS parameter has DECMPR(\*NO).
- ❖ A precompressed file is sent and decompressed by the receiving node during the send operation. That is, the FMSYSOPTS has PRECMPR(\*YES) and TOSYSOPTS has DECMPR(\*YES).
- ❖ A file is sent to a single z/OS partitioned data set member.

If you request checkpointing when you are transferring multiple members of a file, checkpoints are taken only at member boundaries regardless of the interval specified in the initialization parameters or in the CDSND or CDRCV command.

### **CLASS(n)**

determines the node-to-node session on which an operation can execute. The CLASS is a number in the range of 1–255. If you do not specify the CLASS parameter, it defaults to the number of maximum sessions defined in the mode description for the communications environment.

---

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

---

### **COMPRESS (\*NO|\*YES)\*EXT|X|xx|C|cc)**

specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file is automatically decompressed at the destination.

---

**Note:** Compression is CPU-intensive, and its effectiveness depends on the data and line speed. Only use it if it benefits the installation.

---

You should specify the primary compression character for text data or single-character repetitive data. Extended compression generally offers improved compression for all types of data.

**\*NO** specifies no compression.

**\*YES** enables compression. If **\*YES** is specified, the default primary compression character is `PRIMEchar=X\40\` (hex 40=" ").

If you specify **\*YES**, `Connect:Direct` compresses the data transmitted based on the following rules:

- ❖ Repetitive occurrences (ranging from 2–63) of the primary compression character are compressed to one byte.
- ❖ Repetitive occurrences (ranging from 3–63) of any other character are compressed to two bytes.

**\*EXT** specifies that extended compression is used to search for repetitive strings of characters in data and compress them to codes that are transmitted and converted back to the original string during decompression. Use this parameter when the line transmission speeds are limited, the CPU is available, and the data is repetitive.

`X\xx\` enables compression, where `xx` is the hexadecimal representation of the value to be used for the primary compression character.

`C\cc\` enables compression, where `cc` is the character representation of the value to be used for the primary compression character.

#### **CRC(\*INITPARMS|\*NETMAP|\*OFF|\*ON)**

Determines whether node and Process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to **\*OFF**, setting overrides for CRC checking will be ignored. The default value is **\*INITPARMS**.

**\*INITPARMS** specifies that the CRC setting is determined by the value for the initialization parameter `TCP.CRC`.

**\*NETMAP** specifies that the CRC setting is determined by the value for the `NETMAP` parameter, Cyclic Redundancy Check. The value for the initialization parameter `TCP.CRC` must be set to **\*ON** and the value for the initialization parameter `TCP.CRC.OVERRIDE` must be set to **\*YES**.

**\*OFF** specifies that CRC is disabled.

**\*ON** specifies that CRC is enabled.

#### **ECTUNING(eccllevel,ecwinsize,ecmlevel)**

**ECCLEVEL** specifies the extended compression level, which affects how much CPU the extended compression routines use. Higher compression levels use more CPU but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is **\*INITPARMS**.

**ECWINSIZE** specifies the extended compression window size, which is specifically for the history buffer that is filled from the user's input buffer (both compressing and decompressing). The window specifies the amount of storage designated to maintain data previously read.

This data can be scanned for string matches. The extended compression window size affects how much virtual memory the extended compression routines use. Higher window size values use more memory but achieve greater compression. The valid values for this subparameter are 8–15, inclusive. The default value is **\*INITPARMS**.

**ECMLEVEL** specifies the extended compression memory level parameter, which determines how much memory should be allocated for other internal data structures like the hash table and the previous table (pointers to previous strings starting with the same 3 characters). The extended memory level affects how much memory the extended compression routines use. Higher memory levels use more virtual memory but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is **\*INITPARMS**.

**EXCLUDE (generic|member-name|(start-range/stop-range)|'list')**

specifies criteria that identifies the file members that are not to be copied. EXCLUDE allows you to make exceptions to members specified generically or by range in the SELECT option.

**generic** specifies a generic member name. For example, if you specify CD\*, all member names beginning with CD are excluded.

**member-name** specifies an individual member name.

**start-range/stop-range** specifies first and last names in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). A slash (/) separates the first (start-range) and last (stop-range) member names. The two members specified, as well as all members in the range, are excluded.

**'list'** specifies any combination of the subparameters.FDISP(current, normal end, abnormal end) specifies the status of the sending file and what is to be done with the file after notification of successful or unsuccessful transmission.

**current** specifies the status of the file. Valid values for this subparameter are:

**SHR** specifies that the member is read simultaneously by another operation. The default is SHR.

**OLD** specifies that the operation is to be given exclusive control of the file.

**normal end** specifies the disposition of the file following a normal operation termination, resulting in a zero completion code. Valid dispositions are:

**KEEP** specifies that the system is to keep the member after the operation ends.

**DELETE** specifies that the system is to delete the member after the operation ends.

**abnormal end** specifies the disposition of the file after an abnormal operation termination, resulting in a nonzero completion code. Valid source file dispositions are:

**KEEP** specifies that the system is to keep the member after the operation terminates abnormally.

**DELETE** specifies the system is to delete the member after the operation terminates abnormally.

---

**Note:** Currently, normal end disposition and abnormal end disposition are not applicable for Connect:Direct for i5/OS.

---

**NOTIFY(userid)**

specifies the user ID on the sending node to receive status messages that are directed to the user's message queue.

**PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to run a task, for example TAPE.

---

**Note:** This parameter is for use only in Connect:Direct for z/OS environments.

---

**PNAME(process-name)**

identifies the Process in any messages or statistics. The name is from 1 to 12 characters long. The first character must be alphabetic. The default name is CDSND.

---

**Note:** Only Connect:Direct for i5/OS and Connect:Direct UNIX support Process names with more than 8 characters.

---



**PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local i5/OS node. Use this parameter only to validate security with a different ID than the one you used to sign on to the system.

**id** specifies the security ID passed to the security system on the sending node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. The security system on the sending node can use this parameter to validate the current security password (1–10 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password. The security system can use this parameter to change the current security password to the new security password (1–10 alphanumeric characters).

---

**Note:** Although you can specify a value for newpswd, Connect:Direct for i5/OS does not support this parameter.

---

**REPLACE (\*YES|\*NO)**

specifies that the sending file replaces members of the same name at the receiving file or PDS.

**RMTWORK(\*NO|\*YES)**

specifies whether a remote process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function.

**\*YES** allows a remote process to run on the session after Connect:Direct for i5/OS completes its Process.

**SACCT(accounting-data)**

specifies the accounting data for the receiving node. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, the string must be enclosed in single quotation marks.

**SECOPTIONS(\* \*)**

Encryption and Digital Signature values.

Encryption specifies if data encryption is to be performed and what algorithm to use.

Digital Signature specifies if a digital signature is created while a file is being transmitted.

---

**Note:** The SECOPTIONS parameter is valid only if the Connect:Direct Secure+ Option is installed. For information about these parameters, refer to the *Connect:Direct Secure+ Option Implementation Guide*.

---

**SELECT(member, new-name, R/NR, 'list')**

specifies selection criteria by which file members are to be copied. Use this parameter when you specify TYPE(FILE) in the FMSYSOPTS parameter.

---

**Note:** \*FIRST is not a valid subparameter.

---

The SELECT parameter consists of three positional subparameters separated by commas.

**member** specifies the member(s) to be copied. You can also specify a list of individual member names or generic member names separated by spaces. However, you cannot use the new-name and replace-member (R) subparameters when specifying a list. Member selections are:

\* represents a global generic. A global generic indicates that all members of the file are to be included.

**generic** specifies a generic member name. For example, if you specify CD\*, all member names beginning with CD are selected for copying.

**member-name** specifies an individual member name.

**start-range/stop-range** specifies first and last names in an alphanumeric range of members. Although member names in a range are treated as generics, you cannot use them with an asterisk (\*). A slash (/) separates the first (start-range) and last (stop-range) member names. The two members specified, as well as all members in the range, are copied.

**new-name** specifies a new name for a member. The new-name subparameter cannot be used if you use a generic or range to select the members to be copied or if you specify a list of member names.

**R/NR** specifies whether a copied member replaces an existing member of the same name in the receiving file. When used with a generic or a range, this subparameter applies to all members selected.

---

**Note:** When using a generic and specifying **NR** or **R**, the second positional subparameter (new-name) must be null, for example, **generic\*,NR**.

---

**R** specifies that a member replaces an existing member of the same name in the receiving file. When used with the new-name subparameter, **R** applies to the specified new-name and not the original member name.

**NR** specifies that a member does not replace an existing member of the same name in the receiving file. NR overrides the REPLACE parameter.

**'list'** specifies any combination of the subparameters.

#### **SNODEID (id, pswd, newpswd)**

specifies security user IDs and passwords at the remote node. If nothing is specified for this parameter, the user profile of the user who submits the command is sent to the remote node with a blank password.

**id** specifies the security ID passed to the security system on the receiving node (1–64 alphanumeric characters).

**pswd** specifies the current security password. The security system on the SNODE can use this parameter to validate the current security password (1–64 alphanumeric characters). This is optional unless you have security set to require a password.

**newpswd** specifies the new security password and is used by the security system to change the current security password to the new security password (1–64 alphanumeric characters).

---

**Note:** newpswd is not supported if the remote node is an i5/OS node.  
Parameter values are case sensitive.

---

#### **TBUFND(n)**

specifies the number of I/O buffers VSAM uses for transmitting data between virtual and auxiliary storage. Valid values range from 1–510. The default is 2. Increasing this number generally improves the I/O performance on the file but requires more memory.

**TDBPARMS (dbid, dbsubid)**

indicates that the Connect:Direct SQL SERVER is to process the I/O request against the database identified by the subparameters of this keyword and defined by the SQL= parameter. The DBPARMS parameter is used with the optional Connect:Direct for SQL feature. This parameter is valid for z/OS nodes only. Only DB2 is supported.

To determine the values to code to access a given DB2 table or set of DB2 tables, consult your database administrator.

**dbid** specifies the DB2 database ID of one to four bytes.

**dbsubid** specifies the DB2 database sub-ID of one to four bytes.

**TDCB (modeldcb, blksize, den, dsorg, keylen, linct, lrecl, optcd, recfm, rkp, trtch)**

overrides attributes of the source file. For existing source and destination files, DCB attributes are determined from the operating system unless specified. For a new destination file, use the DCB attributes of the source file to allocate the destination file unless you provide DCB information in the operation.

**modeldcb** specifies a model data set control block (DSCB).

**blksize** specifies the length in bytes of the block. The minimum length is 18 bytes, and the maximum length is 32,760 bytes.

**den** specifies the magnetic tape mode setting. The following table shows the values for the DEN parameter for 7- and 9-track tapes. When coded together, the DEN and TRTCH values are used to select a tape device for allocation by the remote node.

DEN	7-Track Tape	9-Track Tape
1	556 bpi	--
2	800 bpi	800 bpi
3	--	1600 bpi
4	--	6250 bpi

**dsorg** specifies the file organization. Valid file organizations are BDAM, ISAM, PO, PS, and VSAM.

**keylen** specifies the length of the keys used in a file. Maximum length is 255 bytes.

**linct** specifies the blocks or tracks searched to find a free block or available space.

**lrecl** specifies the record length in bytes.

**optcd** specifies optional processing associated with this file. This specification only applies to this file and is not automatically applied to the other files involved in the RECEIVE operation. Valid options are:

**W** performs write validity checks on direct access storage devices.

**Q** performs ASCII-to-EBCDIC conversion for input files and EBCDIC-to-ASCII conversion for output files. **Q** is the default and only used for AL-labeled tape files.

**Z** performs reduced error recovery for tape files.

**refm** specifies the format of the records in the file. You can specify any valid record format, such as:

**F** specifies fixed format.

**FA** specifies fixed format with ASA printer control.

**FB** specifies fixed block format.

**FBA** specifies fixed block format with ANSI carriage control.

**FM** specifies fixed format with machine code control character.

**U** specifies undefined format.

**V** specifies variable format.

**VB** specifies variable block format.

**VBA** specifies variable block format with ASA printer control.

**VBM** specifies variable block format with machine code control character.

**VS** specifies variable spanned format.

**VBS** specifies variable block spanned format.

For FDR volumes, you must specify RECFM=U.

**rkp** specifies the position of the first byte of the record key within each logical record. The beginning byte of a record is addressed as **0**.

**trtch** specifies the magnetic tape mode setting. When coded together, the TRTCH and DEN values select a tape device for allocation by Connect:Direct for z/OS. Valid options are:

**C** specifies data conversion, odd parity, and no translation.

**E** specifies no data conversion, even parity, and no translation.

**T** specifies no data conversion, odd parity, and BCD or EBCDIC translation.

**ET** specifies no data conversion, even parity, and BCD or EBCDIC translation.

**COMP** is a feature for 3480X tape drives only. It enables Improved Data Recording Capability (IDRC), which compresses the data. This subparameter overrides the system-wide IDRC setting for no compression. If you are specifying COMP, you must also include a UNIT= parameter that specifies either 3480X or a systems programmer-defined name equivalent to a 3480X tape drive.

**NOCOMP** overrides the system-wide IDRC setting for compression. It applies to 3480X tape drives only.

#### **TDISP (current, normal end, abnormal end)**

specifies the status of the file on the receiving node.

**current** specifies the status of the file before the operation executes. Valid choices for this subparameter are:

**NEW** indicates that the current Connect:Direct operation creates the destination file or member.

**OLD** indicates that the destination member already exists. The operation has exclusive control of the member. If the destination member does not exist, the operation fails.

**MOD** indicates that the operation modifies the member by adding data to the end of the member or, if none exists, allocates a new member. The file is also created if it does not exist.

**RPL** indicates that the destination file or member is replaced. The file is created if it does not exist.

**SHR** indicates that another operation can read the file simultaneously. The file is created if it does not exist on i5/OS.

---

**Note:** Normal end disposition and abnormal end disposition are not applicable when sending to Connect:Direct for i5/OS.

---

For information on subparameters for other Connect:Direct platforms, refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

**TIOEXIT(program-name)**

specifies the name of the user-written program to be given control for I/O-related requests for the associated data.

**TIOPARM(parameter-names)**

specifies a parameter or list of parameters to be passed to the specified exit. The TIOPARM syntax is dependent on the parameter list required by the program specified in TIOEXIT.

**TLABEL(seqno, labeltype, pswdread, inout, retpd, expdt)**

specifies label information for the tape.

**seqno** specifies the relative file position on the tape.

**labeltype** is designated as:

**SL** specifies IBM standard label.

**AL** specifies American National Standard label.

**BLP** specifies bypass label processing.

**LTM** specifies bypass leading tape marks.

**NL** specifies no label.

**pswdread** specifies that a password must be supplied by the operator or user before the file is accessed.

**inout** specifies that a BSAM data set opened for INOUT or a BDAM data set opened for UPDAT is to be read only.

**retpd** specifies the retention period for the data set in days, where *nnnn* is one to four digits.

**expdt** specifies the expiration date for the data set, where *yyddd* or *yyyyddd* is a valid Julian date.

**TLINK(vuserid pwd accmode cuu)**

specifies the disk where the CMS file is located. This parameter allows the user to access the CMS file. This parameter is for a VM remote node.

**vuserid** specifies the owner ID for the CMS minidisk where the file is located. The valid length ranges from 1–8 characters.

**pwd** specifies the appropriate password for the CMS minidisk where the file is located. The maximum length is 256 characters. The default password is ALL.

**accmode** specifies the link access mode.

When used with the FROM parameter, the access modes are **W** (primary read/write access), **M** (primary multiple access), **R** (primary read only), **RR** (primary and remote read only access), **WR** (primary read/write access; alternate read only access), **MR** (primary multiple access; alternate read only access), and **MW** (primary multiple access; alternate read/write only access).

When used with the TO parameter, the access modes are **W**, **M**, and **MW**.

**cuu** specifies the virtual address of the disk where the CMS file is located. Any three-digit number is valid.

**TMSVGP(data-definition-name)**

specifies the receiving file mass storage volume group that resides on a mass storage system (MSS) device. This must be a valid DD (data definition) name.

**TPROTCT(\*YES|\*NO)**

specifies whether an IBM RACF profile is created for a new file on a remote VM system.

**TRACEOPT**

**Trace Instructions**

(L, S, F, C, X, K, D, H, V, W, P|\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)
- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

**Command Trace Options**

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.

- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

**TSPACE(avgblk, type, primary, secondary, pdsblk, rlse, contig, round)**

specifies the amount of storage to be allocated for new files on the destination node.

If you specify SPACE, the DISP of the destination file must be NEW. If you do not specify SPACE and the DISP is NEW, space allocation defaults to the value obtained from the source file (unless a TTYPE has been specified). The default allocation is in blocks and, if supplied with the ROUND option, provides device-independent space allocation.

This parameter is required for all new non-VSAM files unless you have specified a type key that includes a SPACE parameter.

**avgblk** defines the average blocksize.

**type** defines the space allocation. Valid options are:

**CYL** specifies that space is to be allocated by cylinder.

**TRK** specifies that space is to be allocated by track.

**blk** specifies that space is to be allocated by the average block length of the data and is the default. The system computes the number of tracks to be allocated. 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. If no space information is coded, allocation is in blocks, due to device dependencies.

**primary** specifies the primary allocation of storage in number of units.

**secondary** specifies the secondary allocation of storage in number of units.

**pdsblk** specifies the number of PDS directory blocks to be created in the file.

**rlse** specifies that the unused storage allocated to the output file be released. Valid values are \*YES and \*NO.

**contig** specifies that the storage for the local allocation must be contiguous. Valid values are \*YES and \*NO.

**round** specifies that the storage allocated by average block length is rounded to an integral number of cylinders. Valid values are \*YES and \*NO.

**TSQL ('sql-statement'|'DSN=data-set-name')**

defines the SQL statement to be processed against the database identified by the DBPARMS parameter or the DSN containing the SQL statements. The SQL parameter is used with the optional Connect:Direct for z/OS SQL feature. This parameter is valid for Connect:Direct for z/OS nodes only.

**'sql-statement'** defines the actual SQL statement to be processed.

**'DSN=data-set-name'** specifies the data set containing the SQL statement(s) to be processed.

**TTYPE(entry-name)**

specifies the entry name of the type defaults file on the remote node that contains the default file attributes used to allocate the destination file.

**TUNIT(file-address)**

specifies the unit address, device type, or user-assigned group name where the file resides or will reside. For SAM-to-SAM copies where the destination file is new and the UNIT parameter is not coded, the device type from the source file is used. For VSE nodes, you must specify either this parameter or TVOL.

Specify a unit-count to allow additional units to be allocated if required, or specify **P** to allocate the same number of units as volumes and then parallel mount the volumes.

**TVOL(private, volcount, ser)**

specifies the volume serial number or numbers containing the file and optional processing associated with the file. If TVOL is not specified, the file must be cataloged. For VSE nodes, you must specify either this parameter or TUNIT.

**private** specifies allocation of an output file only if the volume is specifically requested and is used for direct access storage devices only. Valid values are \*YES and \*NO.

**volcount** specifies the maximum number of volumes required by an output file.

**ser** identifies by serial number the volumes on which the output file resides or will reside.

**TVOLREF(data-set-volume)**

allows you to place a data set on the same volume as the referenced data set. It must be cataloged on the system where it is referenced.

**TVSAMCAT = (dsn, vmid, pwd, accmode, cuu)**

specifies the catalog for the VSAM file to be copied. This parameter is required only if using a catalog other than the master catalog. This parameter is for a VSE or VM remote node.

**dsn** specifies the filename of the VSAM catalog containing the file to be copied. The maximum length is 44 characters.

**vmid** specifies the owner ID for the VSAM minidisk where the file is located. The maximum length is eight characters.

**pwd** specifies the appropriate password for the VSAM minidisk where the file is located. The maximum length is eight characters.

**acmode** specifies the link access mode. Valid access modes are NULL, **W** (primary read/write access), **M** (primary multiple access), and **MW** (primary multiple access; alternate read/write only access).

**cuu** specifies the virtual address of the disk where the VSAM file is located. Any three-digit number is valid.

---

## Using Block Mode Transfer

Connect:Direct for i5/OS supports block mode transfer for send operations. In this mode, Connect:Direct uses the file block as the data unit. Using the block mode transfer improves performance because the sending and receiving nodes handle fewer data units. This type of transfer reduces instruction path length and decreases execution time. Larger units move between Connect:Direct and the file access routines, requiring fewer calls to the access routines and further improving performance.

Connect:Direct for i5/OS simulates block reads and writes. Using logical blocks instead of physical blocks, Connect:Direct for i5/OS allows you to modify the blocksize to match the other node in some cases.



Connect:Direct for i5/OS supports block mode transfer when the following conditions exist:

- ❖ Remote node supports block mode
- ❖ Source and destination files have like characteristics
- ❖ Files are supported file types

Connect:Direct for i5/OS supports block mode transfers for the following i5/OS file types:

- ❖ DB2 for i5/OS files (without staged compression)
- ❖ Spool files
- ❖ Save files
- ❖ Stream files (using the integrated file system)

## Simulating Blocksizes

The simulated blocksize determines whether a block mode transfer occurs. If the block mode transfer does occur, the simulated blocksize is used for the transfer. The following rules determine what simulated blocksize Connect:Direct for i5/OS uses.

As the receiving node, Connect:Direct for i5/OS determines the simulated blocksize by these conditions in order of precedence:

- ❖ Destination file blocksize specified on the request
- ❖ Source file blocksize sent by the sending node in the COPY control block

If neither condition exists, Connect:Direct uses the record mode.

As the sending node, Connect:Direct for i5/OS determines the simulated blocksize by these conditions in order of precedence:

- ❖ Source file blocksize specified on the request
- ❖ Destination file blocksize specified on the request
- ❖ Calculated value\*

\*Largest multiple of record size less than some arbitrary value, such as 4096.

## Block Mode Transfer Examples

The following block mode transfer examples include both sending and receiving examples.

### CDRCV Example—Not Specifying Source File Blocksize

In the first example, the request does not specify the source file blocksize. In this scenario, the following occurs:

- ❖ The Connect:Direct for i5/OS node sets the source file simulated blocksize to 3200 by using the destination file value.
- ❖ The receiving Connect:Direct for i5/OS sets the destination file simulated blocksize to the value specified on the request.
- ❖ Because the blocksizes match, the block mode copy occurs.

```
C:D i5:    receiving
request:   source blksize=not specified; dest blksize = 3200
srce file: lrecl=80, (on another C:D i5 3.3 node)
dest file: NEW
```

## CDRCV Example—Specifying Source File Blocksize

In the second example, the following occurs:

- ❖ Connect:Direct for i5/OS calculates the simulated blocksize at 3200 because the destination file is new and the source file specifies a blocksize.
- ❖ The receiving node uses the existing destination file blocksize of 3200.
- ❖ A block mode transfer occurs because the blocksizes match.

```
C:D i5: receiving
request: source blksize=not specified; dest blksize = not specified;
srce file: lrecl=80, blksize=3200 (on z/OS node)
dest file: NEW
```

## CDSND Examples—Simulating Blocksize

In the third example, the following events occur:

- ❖ Connect:Direct for i5/OS sets the simulated blocksize at 6400 from the request.
- ❖ The receiving node creates the destination file using source file attributes of blksize=6400.
- ❖ Because the file attributes match, the block mode transfer occurs.

```
C:D i5: sending
request: source blksize=6400; dest blksize = not specified;
srce file: lrecl=80
dest file: NEW (on z/OS)
```

## CDSND Examples—Setting Actual Blocksize

In the fourth example, the following events occur:

- ❖ The remote node sets the destination file blocksize to 3200, the actual blocksize of the destination file.
- ❖ Connect:Direct for i5/OS calculates the simulated blocksize at 3200.
- ❖ The blocksizes are identical, so block mode transfer occurs.

```
C:D i5: sending
request: source blksize=not specified; dest blksize = not specified;
srce file: lrecl=80
dest file: existing: lrecl=80, blksize=3200 (on z/OS)
```

## CDSND and CDRCV Example

In the fifth example, Connect:Direct for i5/OS calculates the simulated blocksize at 32720 because the copy is between two Connect:Direct for i5/OS nodes.

```
C:D i5 sending and receiving
request: source blksize=not specified; dest blksize=not specified
srce file: lrecl=32720
dest file: new
```

## Using the CDSND Screens

Use the **Connect:Direct Send File (CDSND)** screens to send a file to a remote node.

### Sending a File to a Remote Node

Complete the following steps to send a file to a remote node.

1. Access the CDSND screens by typing **CDSND** at the command line and pressing **F4** or by choosing the send file option from the Connect:Direct menu.
2. Type in the remote node name at the screen prompt and press **Enter**. The command retrieves the remote node environment and displays fields for required parameters.
3. Type values for required and optional parameters as appropriate and press **Enter**.

Following is a sample of the first screen in the **Connect:Direct Send File (CDSND)** screen flow. The sample shows the keyword display option.

#### Connect:Direct Send File Screen 1

```

Connect:Direct Send File (CDSND)

Type choices, press Enter.

Receiving Node Name . . . . . > DALLAS
Receiving Node Environment . . . OS400 AS400, MSP, MVS, OS400...
Sending File Name . . . . . _____

Receiving File Name . . . . . _____

Sending System Operations . . . 'TYPE(FILE)'
_____
_____
_____
_____

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

The content of this screen and the following screens depends on the receiving node environment.

### Connect:Direct Send File Screen 2

```

Connect:Direct Send File (CDSND)

Type choices, press Enter.

Receiving System Operations . . 'TYPE(FILE)'
_____
_____
_____
_____
_____

Trace Options:
Trace Instructions . . . . . *CURRENT_____
Command trace options . . . . *NONE_____
Process Name . . . . . CDSND_____ Name
Sending Node User Info.:
User Id . . . . . *CURRENT_____ Character value, *CURRENT
Password . . . . . _____ Character value
New Password . . . . . _____ Character value
User to Notify . . . . . *CURRENT_____ Name, *CURRENT, *NONE
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
    
```

### Connect:Direct Send File Screen 3

```

Connect:Direct Send File (CDSND)

Type choices, press Enter.

Receiving Node User Info.:
User Id . . . . . _____
_____
Password . . . . . _____
New Password . . . . . _____

Class . . . . . *MODED 1-255, *MODED
Checkpoint Interval . . . . . *INITPARMS 0-9999999, *INITPARMS
Allow remote process . . . . . *NO Character value, *YES, *NO
Sending File Disposition:
Current Disposition . . . . . SHR SHR, OLD
Normal End Disposition . . . . _____ KEEP, DELETE
Abnormal End Disposition . . . _____ KEEP, DELETE
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
    
```

### Connect:Direct Send File Screen 4

```

Connect:Direct Send File (CDSND)

Type choices, press Enter.

File Members Selection List . . _____
_____
_____

File Members Exclusion List . . _____
_____
_____

Sending block size . . . . . *CALC      Character value, *CALC
Same Member Replaced . . . . . *YES      *YES, *NO
CRC . . . . . *INITPARMS    *INITPARMS, *NETMAP, *ON, *OFF
Secure+ Options:
  Encryption . . . . . *          */Y/N/Algorithm
  Digital Signature . . . . . _        *, N, Y
  Compress Data . . . . . *NO       Character value, *NO, *YES...

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

### Connect:Direct Send File Screen 5

```

Connect:Direct Send File (CDSND)

Type choices, press Enter.

Extended compression tuning:
  Extended compression level . . *INITPARMS  1-9, *INITPARMS
  Extended compression window . *INITPARMS  8-15, *INITPARMS
  Extended compression memory . *INITPARMS  1-9, *INITPARMS
Receiving File Type Parameter .
Receiving File Disposition:
  Current Disposition . . . . . NEW        NEW, OLD, MOD, RPL, SHR
  Normal End Disposition . . . . CATLG      CATLG, KEEP
  Abnormal End Disposition . . . _____  KEEP, CATLG, DELETE
Receiving Unit:
  Address/Type/Group/Unit-ID . . _____  Character value
  Receiving Unit Count/P . . . . _____  Character value, P

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```



---

# Receiving a File

Receiving a file on Connect:Direct for i5/OS requires the Connect:Direct Receive (CDRCV) command. Understanding the Connect:Direct for i5/OS file systems, the CDRCV command parameters, and the **CDRCV** screens enables you to use this command more effectively. This chapter provides the following information to help you use this command:

- ❖ Using the file systems
- ❖ Using the CDRCV command
- ❖ Using block mode transfer
- ❖ Using **CDRCV** screens

Additional information about the use of command parameters for receiving remote files is available on the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

---

## Using the File Systems

Connect:Direct for i5/OS supports two types of file systems for the file transfers: native file and integrated file. The native file system and the integrated file system can both be used to send files to and receive files from any supported file systems.

### Native File System

The native file system used in previous releases of Connect:Direct for i5/OS remains functional. This file system accepts the file name in the following formats:

**library-name/file-name**  
specifies the library and file name to be copied.

**library-name/file-name(member-name)**  
specifies the library, file, and member name of the member to be copied. You cannot use the EXCLUDE and SELECT parameters if you specify a member name.

### Integrated File System

As part of the i5/OS, the integrated file system provides an integrated structure over all data stored in the iSeries. This file system supports stream input/output and storage management like personal computers and UNIX operating systems.

The Connect:Direct for i5/OS integrated file system provides application and data compatibility with other platforms. The integrated file system consists of five file system types:

- ❖ /QLANSrv is a LAN server file system
- ❖ /QDLS is a document library file system
- ❖ /QSYS.LIB is the iSeries DB2 file system
- ❖ /QOpenSys is a UNIX-style file system
- ❖ All others are part of the root file system

## Specifying a Path Name

The integrated file system recognizes the objects by the path specification. You must specify the fully qualified path name to transfer a file. Connect:Direct for i5/OS identifies a path that starts with a slash (/) or a backslash (\) as a request for the integrated file system.

The integrated file system does not support file field description. You cannot specify the SNDFFD(\*YES) under this file system. Similarly, you cannot specify SELECT or EXCLUDE with the integrated file system.

To use this system, you must specify the fully qualified path name in one of the following formats:

**/QSYS.LIB/library-name.LIB/file-name.FILE/member-name.MBR**  
specifies the library, file name, and member name to be copied using the integrated file system.

**/directory/file-name/QLANSrv/file-name/QDLS/folder-name/QOpenSys/file-name**  
specifies the directory, folder, and file names to be copied using the integrated file system.

Directories like /QOpenSys and /root are case-sensitive. The /QSYS.LIB notation is case-sensitive only if you enclose the name in single quotation marks.

---

## Using the CDRCV Command

Use the CDRCV command to receive a file from a remote node. This command receives files from any file systems supported by the i5/OS V3R1M0 or later. You can also specify a user-exit command (either **EXITCMD** or **FAILCMD**) to be executed after the completion of a copy process in either FROMSYSOPTS or TOSYSOPTS. Using EXITCMD, you can specify a command to be executed only if the copy process is successful. You can also specify a command to be executed only if the copy process is not successful by using the FAILCMD command. The EXITCMD and FAILCMD commands must be executed on an i5/OS using Connect:Direct release 3.3.00 and above. The EXITCMD and FAILCMD are executable iSeries commands as well as Connect:Direct as long as the size of the command does not exceed 127 bytes.

## CDRCV Command Example

The command shown in the following figure is an example of the CDRCV command.

```
CDRCV      PNAME(CDRCVD) SNODE(DALLAS.OS3901)
          SNODENVIRN(OS390)
          FDSN(DALLAS.REPORT)
          TDSN('DALLIB/REPORT')
          TOSYSOPTS('TYPE(FILE)')
```



## CDRCV Command Format

Following are the parameters for the CDRCV command. Required parameters are in bold.

Command	Parameter
CDRCV	<b>SNODE(remote-node-name)</b>
	<b>SNODENVIRN(remote-environment-name)</b>
	<b>FDSN(from remote-data-set-name)</b>
	<b>TDSN(to data-set-name)</b>
	<b>FMSYSOPTS('</b> PRECMPR(*YES *NO) CODEPAGE(nnnnn,nnnn) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command) SNDFFD(*YES *NO) TEXTFILE(*YES *NO) TYPE(FILE MBR OBJ) CCSID(nnnnn) EORCHAR(xxxx) ALWNULL(Y) )
	<b>Note:</b> FMSYSOPTS is required if sending node is i5/OS)
	<b>TOSYSOPTS('</b> DECMPR(*YES *NO) CODEPAGE(nnnnn,nnnn) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command) <b>TYPE(FILE MBR OBJ SPLF)</b> XTRAN(table-name)[XTRANLSO(so-code)] XTRANLSI(si-code)[XTRANLDATA(MIXED DBCS)] CCSID(nnnnn) EORCHAR(xxxx) )
	CKPTINV(n)
	CLASS(n)
	COMPRESS(*NO *YES *EXT X xx C cc)
	CRC(*INITPARMS *NETMAP *OFF *ON)
	ECTUNING (eccllevel, ecwinsize, ecmlevel)
	EXCLUDE(generic, member-name, (start-range/stop-range), 'list')
FALIAS(*YES *NO)	
FBUFND(n)	
FDBPARMS(dbid, dbsubid)	
FDCB(modeldcb, blksize, den, dsorg, keylen, linct, lrecl, optcd, recfm, rkp, trtch)	
FDISP(current, normal end, abnormal end)	
FIOEXIT(program-name)	
FIOPARM(parameter-name)	

Command	Parameter
	FLABEL(seqno,labeltype,pswdread,inout,retpd expdt)
	FLINK(vmuserid,pwd,accmode,ccuu)
	FMSVGP(data-set-name)
	FPDSDIR(*YES *NO)
	FSQL(sql-statement DSN=data-set-name)
	FTYPE(entry-name)
	FUNIT(unit-name)
	FVOL(private, volcount, ser)
	FVOLREF(data-set-name)
	FVSAMCAT(dsn,vmid,pwd,accmode,ccuu)
	NOTIFY(userid)
	PLEXCLASS(user-defined class)
	PNAME(process-name)
	PNODEID(id, pswd, newpswd)
	REPLACE(*YES *NO)
	RMTWORK(*YES *NO)
	SACCT(snode-accounting-data)
	SECOPTIONS(* *)
	SELECT(member, new-name, R/NR, 'list')
	SNODEID(id, pswd, newpswd)
	TDISP(current)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS *CURRENT Command Trace Options: L,S,F,C,X,K,D,H,V,P *ALL *EXT *COMMON  *NONE)

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

## Required Parameters

Unless otherwise noted, these required parameters apply to all remote node environments.

### **SNODE(remote-node-name)**

specifies a 1–16 character alphanumeric name for the remote node. The node name must be defined in the network map.

### **SNODENVIRN(remote-environment-name)**

specifies the remote node environment.

**FDSN('from remote-data-set-name')**

specifies the source file name. File names are verified based on the i5/OS standard file name conventions. The length of the input field displayed is 132 characters, by default. If you want to increase the length of the field to 256 characters, type an ampersand(&) followed by a blank in the first two positions of the field.

Connect:Direct for i5/OS provides two systems for handling the files: Native File System and Integrated File System. Refer to *Using the File Systems* on page 47 for detailed information about the file systems. The following table summarizes the data set name requirements for Connect:Direct platforms.

SNODENVIRN	FDSN Format	Special Considerations
OS/400	Native File System 'library-name/file-name' 'library-name/file-name (member-name)'	Enclose in single quotation marks.
	Integrated File System 'QSYS.LIB/library-name.LIB /file-name.FILE /member-name.MBR' or '/directory/file-name  /QLANSrv/file-name  /QDLS/folder-name  /QOpenSys/file-name'	You cannot specify the *SNDFFD, *SELECT, and *EXCLUDE parameters with the integrated file system.
	Spool File Support spooled-file-name	Do not use member names longer than eight characters when sending to z/OS, VSE, and MSP. The results are unpredictable.
OS/390, VSE, MSP	Standard z/OS DSN naming conventions.	Single quotes allow special characters. Do not use member names longer than eight characters. The results are unpredictable.
VM	Fully qualified file-name	
Tandem	\system.\$volume.subvolume.filename	
Windows	drive:\directory\subdirectory\filename or \\servername\sharepointname\subdirectory\filename	
UNIX	/directory/subdirectory/filename	
OpenVMS	node::device:[directory]filename.type;version	When you are copying a file , the file specification must be enclosed in single quotation marks if it uses spaces or other special characters not recognized by the other node.
Stratus VOS	Fully qualified path and file name	Enclose the file name in double quotation marks when you use a reserved word (statement name or keyword) for the file name.

**Note:** The values of \*LIBL, \*CURLIB, and QTEMP cannot be used when sending to or from an iSeries system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the iSeries.

**TDSN('to data-set-name')**

specifies the local file name. File names are verified based on the i5/OS standard file name conventions.

The length of the input field displayed is 132 characters, by default. If you want to increase the length of the field to 256 characters, type an ampersand(&) followed by a blank in the first two positions of the field.

Connect:Direct for i5/OS supports two file systems: native and integrated. The file system description begins on page 47. The DSN format summaries for i5/OS are presented with the FDSN parameter description on page 51.

---

**Note:** The values of \*LIBL, \*CURLIB, and QTEMP cannot be used when sending to or from an iSeries system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the iSeries.

---

**FMSYSOPTS('**

PRECMPR(\*YES|\*NO)  
 EXITCMD(valid i5/OS command)  
 FAILCMD(valid i5/OS command)  
 SNDFFD(\*YES|\*NO)  
 TEXTFILE(\*YES|\*NO)  
 TYPE(FILE|MBR|OBJ)  
 CODEPAGE(nnnnn,nnnnn)  
 CCSID(nnnnn)  
 EORCHAR(xxxx)  
 ALWNULL(Y)  
 ')

specifies remote system operations. All parameters shown are valid only if the sending node is i5/OS. The maximum number of characters for SYSOPTS is 2K. This parameter is required when the remote file is on another Connect:Direct for i5/OS node.

For additional information about valid platform-specific SYSOPTS parameters, see the appropriate COPY statement section of the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

**PRECMPR(\*YES|\*NO)** specifies that the source file was previously compressed using the CDCOMP command. To receive a file that has been compressed, you must specify the PRECMPR(\*YES) parameter with the CDRCV command. For information on how to compress files, refer to Chapter 9, *Compressing Files*. This is for i5/OS only.

**EXITCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is successful.

**FAILCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is not successful.

**SNDFFD(\*YES|\*NO)** specifies whether the receiving node transfers file field descriptions. This is for i5/OS only.

Use the following formula to determine whether a file can be sent with the file field descriptions. If the bytes required value is greater than 12K, you cannot transfer the file field descriptions.

$(12 \times \text{number of keys}) + (45 \times \text{number of fields}) + \text{SYSOPTS} = \text{bytes required}$
--

---

**Note:** You cannot use this parameter with the integrated file system.

---

**TEXTFILE(\*YES|\*NO)** specifies that the file being sent is a text file.

**TYPE(FILE|MBR|OBJ|SPLF)** specifies the type of data to be copied. You *must* define the type of data structure to be copied as one of the following:

**FILE** specifies that the file being copied is a physical database file.

**MBR** specifies that the member being copied is a member of a physical database file.

**OBJ** specifies that the data to be copied is in save file format.

**SPLF** specifies that the data to be copied is in i5/OS spooled output format.

**CODEPAGE(nnnnn,nnnnn)** converts data from one code page to another. The supported conversions are identified on the IBM Web site in the V5R1 Information Center, DB2 Universal Database for iSeries SQL Reference, Appendix E. CODEPAGE() on Connect:Direct for i5/OS relates to CODEPAGE() developed for the z/OS system, UNIX, and Windows.

**nnnnn** identifies the IBM CCSID code page number to convert from and the IBM CCSID code page number to convert to. If the FROM code page or TO code page is locale, the code page for the locale where CONNECT:Direct is executing is used. For example, if `fmsysopts(codepage(locale,1208))`, then the FROM code page is the code page for the locale and the TO code page is UTF-8.

**CCSID(nnnnn)** specifies the coded character set ID to use when creating IFS files. If no character set ID is specified, the system default is used.

**EORCHAR(xxxx)** specifies the two-to four-digit hexadecimal symbol to identify the character to use in text files as the end-of-record character, for example, EORCHAR(0A).

**ALWNULL(Y)** specifies if NULLS characters will be processed when sending DB2 files.

#### TOSYSOPTS(‘

DECMPR(\*YES|\*NO)

EXITCMD(valid i5/OS command)

FAILCMD(valid i5/OS command)

TYPE(FILE|MBR|OBJ|SPLF)

XTRAN(table-name) [XTRANLSO(so-code)]

XTRANLSI(si-code)|XTRANLDATA(MIXED|DBCS)]

CCSID(nnnnn)

EORCHAR(xxxx)

’)

specifies system operation parameters at the local node (i5/OS). The maximum number of characters permitted for TOSYSOPTS is 2K.

**DECMPR(\*YES|\*NO)** specifies whether the local Connect:Direct is to decompress the received file. DECMPR(\*NO) instructs the local system to place the received data in a database file without decompressing it.

---

**Note:** Ensure that the destination file contains no members before you receive a file with DECMPR(\*NO) specified.

---

**EXITCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is successful.

**FAILCMD**(valid i5/OS command) specifies a command to be executed only if the copy process is not successful.

**TYPE(FILE|MBR|OBJ|SPLF)** specifies the type of data structure to be created. You *must* define the data structure to be created as one of the following:

**FILE** specifies that the file being created is a physical database file.

**MBR** specifies that the member being created is a member of a physical database file.

**OBJ** specifies that the file being created is in save file format.

**SPLF** specifies that the data is copied to an i5/OS spooled output file.

The default is TYPE(FILE).

For each type, you can specify certain additional TOSYSOPTS parameters.

The following subparameters are valid for **TYPE(FILE)** and **TYPE(MBR)** files:

**RCDLEN(record-length)** specifies the data length for each record in the file.

**FILETYPE(\*SRC|\*DATA)** specifies the type of file to be created. Use this parameter whenever a file is created.

**TEXT('text description')** specifies a text description to be associated with this member (and file, if created). This description cannot exceed 50 characters and must be enclosed in single quotes.

**EXPDATE(expiration-date)** specifies the date after which the new or replaced member cannot be used. If EXPDATE is not specified, then the file does not have an expiration date.

**MAXMBRS(number|\*NOMAX)** specifies the maximum number of members a physical file can contain.

**SIZE(#-of-recs|incr-value #-of-incrs|\*NOMAX)** is used when a new file is created for the file received.

**AUT(\*CHANGE|\*ALL|\*USE|\*EXCLUDE)** specifies the authority to be given to a user who does not have specific authority to the file or member, is not on the authorization list, and whose user group does not have specific authority to the file or member.

The following parameters are valid for **TYPE(OBJ)** files:

**MAXRCDS(number-of-records|\*NOMAX)** specifies the maximum number of records the save file (which was created to hold the data received) can reach.

**ASP(auxiliary-storage-pool)** specifies the auxiliary storage pool from which the system allocates storage for the save file. Valid values range from 1–16. The default is 1.

**TEXT('text description')** specifies a text description to be associated with this object. This description cannot exceed 50 characters and must be enclosed in single quotes.

**AUT(\*EXCLUDE|\*CHANGE|\*ALL|\*USE)** specifies the authority to be given to a user who does not have specific authority to the object, is not on the authorization list, and whose user group does not have specific authority to the object.

**CODEPAGE(nnnnn,nnnnn)** converts data from one code page to another. The supported conversions are identified on the IBM Web site in the V5R1 Information Center, DB2 Universal Database for iSeries SQL Reference, Appendix E. CODEPAGE() on Connect:Direct i5/OS relates to CODEPAGE() developed for the z/OS system, UNIX, and Windows.

**nnnnn** identifies the IBM CCSID codepage number to convert from and the IBM CCSID code page number to convert to. If the FROM code page or TO code page is locale, the code page for the locale where CONNECT:Direct is executing is used. For example, if `tosysopt(codepage(1208,locale))`, then the FROM code page is UTF-8 code page and the TO code page is the local code page.

The following TOSYSOPTS parameters are valid for **TYPE(SPLF)** files:

**DEV(\*JOB|\*SYSVAL|device-name)** specifies the name of the printer device description.

**DEVTYPE(\*IPDS|\*SCS|\*USERASCII)** specifies the type of data stream created for a printer file from the data received.

**PAGESIZE(page-length page-width)** specifies the length and width of the page used by the printer. This parameter overrides the FORMTYPE parameter.

**LPI(3|4|6|7.5|8|9)** specifies the line space setting (lines per inch) on the printer.

**CPI(5|10|12|13.3|15|16.7|18|20)** specifies the printer character density, in characters per inch, for the printer.

**OVRFLW(overflow-line-number)** specifies the line number on the page at which overflow to a new page begins. The value specified must not exceed the forms length specified for PAGESIZE.

**FOLD(\*NO|\*YES)** specifies whether entire records are printed when the record length exceeds the form width.

**RPLUNPRT(\*YES 'replacement-character'|\*NO)** specifies whether unprintable characters are replaced with printable characters when printed. It also specifies the replacement character.

**ALIGN(\*NO|\*YES)** specifies whether the page must be aligned in the printer before printing is started.

**CTLCHAR(\*NONE|\*FCFC)** specifies whether the data contains printer control characters.

**CHLVAL(\*NORMAL| (channel#1 line#1).. . (channel#12 line#12))** specifies the list of channel numbers with their assigned line numbers.

**FORMFEED(\*DEV|\*CONT|\*CUT|\*AUTOCUT)** specifies the form feed attachments that the printer uses (4214, 5219, and 5553 printers only).

**PRTQLTY(\*STD|\*DRAFT|\*DEV|\*NLQ)** specifies the quality of print produced.

**DRAWER(1|2|3|\*E1)** specifies the source drawer to be used when automatic cut-sheet feed mode is used.

**FONT(\*CPI|\*DEV|font-identifier)** specifies the font identifier to be used for the spooled output file.

**CHRID(\*DEV|\*SYSVAL| graphic-character-set code-page)** specifies the character identifier to use for the spooled output file. \*DEV indicates that the CHRID value the device is designed to handle is used.

**PAGRTT(\*DEV|\*COR|0|90|180|270)** specifies the degree of rotation (clockwise from the edge of the paper first loaded into the printer) of text on each page printed.

**PRTTXT('print-text')** specifies a line of text to be printed at the bottom of each page printed. Up to 30 characters enclosed in single quotes can be specified.

**JUSTIFY(0|50|100)** controls the print positions of the characters on the page (in the spooled file) so the right margin is regular.

**DUPLEX(\*NO|\*YES|\*TUMBLE)** specifies whether the spooled output file is printed on one or both sides of the paper.

**SPOOL(\*YES|\*NO)** specifies whether the data is sent to a spooled file before printing.

**OUTQ(\*JOB|\*DEV|library-name/output-queue- name)** specifies the output queue for the spooled output file created.

**FORMTYPE(form-type)** specifies the type of form to use in the printer when the spooled file is printed. The PAGESIZE parameter overrides this parameter.

**COPIES(number-of-copies)** specifies the number of copies to be printed. Valid values range from 1–255.

**MAXRCDS(maximum-records)** specifies the maximum number of records that can be placed in the spooled file. Valid values range from 1–500000.

**FILESEP(number-of-file-separators)** specifies the number of blank separator pages to be placed between each copy of the file printed.

**HOLD(\*YES|\*NO)** specifies whether the file is to be held on the output queue until released by the user.

**SAVE(\*YES|\*NO)** specifies whether the spooled output file is to be saved on the output queue once printed.

**OUTPTY(\*JOB|output-priority)** specifies the scheduling priority of the file on the output queue.

**USRDTA(user-data)** allows up to 10 characters of data to identify the spooled output file.

**XTRAN(table-name)** specifies the translation table to use. The named table object must exist in a library that is in the library list of the session manager job. If the library name is not in the list, the CDRCV command fails.

If the XTRAN keyword is present, then the following related optional keywords can also be used:

**XTRANLSO (so-code)** specifies extended translate local shift-out code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-out character and overrides the default value of 0E. You can specify any two valid hexadecimal digits.

**XTRANLSI (si-code)** specifies extended translate local shift-in code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-in character and overrides the default value of 0F. You can specify any two valid hexadecimal digits.

**XTRANLDATA (MIXED|DBCS)** specifies extended translate local data format. This keyword specifies the format of the data on the local (i5/OS side) of the COPY.

**MIXED** indicates that the data can contain both DBCS and SBCS characters and that SO/SI characters are used.

**DBCS** indicates that the data is pure DBCS characters and that no SO/SI characters are used.

**XTRANPAD(pc)** specifies that padding characters are in use. When DBCS data is translated from EBCDIC to ASCII, PAD specifies that the SHIFT-OUT and SHIFT-IN characters are replaced by a pad character to allow the displacement of fields within a record to remain unchanged during translation.

When DBCS data is translated from ASCII to EBCDIC, PAD specifies that the input ASCII DBCS file is in a padded format. The character immediately preceding a DBCS character or string is overlaid by the SHIFT-OUT character. The character immediately following a DBCS character or string is overlaid with the SHIFT-IN character. This allows the displacement of fields within a record to remain unchanged during translation.

**pc** is the pad character to be used during EBCDIC to ASCII translation. **pc** is ignored for ASCII to EBCDIC translations. If XTRANPAD is not used, the SI/SO characters are stripped from the file when transferring from EBCDIC to ASCII, and added to the records when sending from ASCII to EBCDIC.

The following rules apply to the use of the XTRAN keyword:

- ❖ You must specify the XTRAN keyword to use extended translation. All other keywords are optional.
- ❖ The default for local shift-out is the IBM standard x0E.
- ❖ The default for local shift-in is the IBM standard x0F.
- ❖ The default local data format is MIXED with SO/SI in use. XTRAN is *not* allowed with PRECMPR(\*YES) or DECMPR(\*NO)



- ❖ If you receive a file from an ASCII platform using extended translation, you *must* specify in the FMSYSOPTS that the data type is binary. Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for platform-specific SYSOPTS parameters.
- ❖ If you receive DBCS data from an ASCII platform using the XTRAN keyword, ensure that the record length of the file on the i5/OS side of the transfer is large enough to contain the translated record.

The record length of the file on the i5/OS side of the transfer is generally larger than the ASCII equivalent. For additional information, refer to the *Maintaining Extended Translation* chapter of the *Connect:Direct for i5/OS Installation and Administration Guide*

**CCSID(nnnnn)** specifies the coded character set ID to use when creating IFS files. If no character set ID is specified, the system default is used.

**EORCHAR(xxxx)** specifies the two-to four-digit hexadecimal symbol to identify the character to use in text files as the end-of-record character, for example, EORCHAR(0A).

## Optional Parameters

The following parameters, listed alphabetically, are optional. Unless otherwise noted, the parameters apply to all remote node environments.

### **CKPTINV(n)**

specifies the number of bytes, from 10K to 2 gigabytes, to receive before taking a checkpoint. The format is *nnnnnnn*, *nnnnnnK*, or *nnnnnnM*, where **K** specifies thousands of bytes and **M** specifies millions of bytes. A value of **0** specifies no checkpoint/restart. Connect:Direct for i5/OS uses the value you specify, rounded up to the nearest record boundary, to determine when to take a checkpoint.

Checkpointing does not occur in the following cases, even if you specify a checkpoint interval:

- ❖ The file being received is compressed by the remote node during the receive operation by the local node using the extended compression feature but is not being decompressed on the local node. That is, the TOSYSOPTS has DECMPR(\*NO).
- ❖ The file being received was precompressed on the remote node and is being decompressed on the local node during the receive operation. That is, the FMSYSOPTS has PRECMPR(\*YES) and TOSYSOPTS has DECMPR(\*YES).
- ❖ A single z/OS partitioned data set member is being received.

A checkpoint interval specified here overrides the system-wide default value.

If you request checkpointing when you are transferring multiple members of a file, checkpoints are taken only at member boundaries regardless of the interval specified in the initialization parameters or in the CDSND or CDRCV command.

### **CLASS(n)**

determines the node-to-node session on which a Connect:Direct operation can execute. The CLASS can be a number from 1–255. If you do not specify the CLASS parameter, it defaults to the number of maximum sessions defined in the mode description for the communications environment.

---

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

---

**COMPRESS(\*NO | \*YES| \*EXT | X\xx\ | C\cc\ )**

specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file is automatically decompressed at its destination unless DECOMP(\*NO) is specified in TOSYSOPTS.

---

**Note:** Compression is CPU-intensive. Its effectiveness is dependent on the data and line speed and should be used only if it benefits the installation.

---

You should specify the primary compression character for text data or single-character repetitive data. Extended compression generally offers improved compression for all types of data.

**\*NO** specifies no compression.

**\*YES** enables compression. If **\*YES** is specified, the default primary compression character is PRIMEchar=X\40\ (hex 40=" ").

If you specify **\*YES**, Connect:Direct reduces the amount of data transmitted based on the following rules:

- ❖ Repetitive occurrences (2–63) of the primary compression character are compressed to one byte.
- ❖ Repetitive occurrences (3–63) of any other character are compressed to two bytes.

**X\xx\** enables compression, where *xx* is the hexadecimal representation of the value to be used for the primary compression character.

**C\cc\** enables compression, where *cc* is the character representation of the value to be used for the primary compression character.

**\*EXT** specifies that extended compression is used to search for repetitive strings of characters in data and compress them to codes that are transmitted and converted back to the original string during decompression. Specify this parameter when line transmission speeds are limited, the CPU is available, and data is repetitive.

**CRC(\*INITPARMS|\*NETMAP|\*OFF|\*ON)**

Determines whether node and Process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to **\*OFF**, setting overrides for CRC checking will be ignored. The default value is **\*INITPARMS**.

**\*INITPARMS** specifies that the CRC setting is determined by the value for the initialization parameter TCP.CRC.

**\*NETMAP** specifies that the CRC setting is determined by the value for the NETMAP parameter, Cyclic Redundancy Check. The value for the initialization parameter TCP.CRC must be set to **\*ON** and the value for the initialization parameter TCP.CRC.OVERRIDE must be set to **\*YES**.

**\*OFF** specifies that CRC is disabled.

**\*ON** specifies that CRC is enabled.

**ECTUNING (ecclevel,ecwsize,ecmlevel)**

**ECCLEVEL** specifies the extended compression level, which affects how much CPU the extended compression routines use. Higher compression levels use more CPU but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is **\*INITPARMS**.

**ECWINSIZE** specifies the extended compression window size, which is specifically for the history buffer that is filled from the user's input buffer ( both compressing and decompressing). The window specifies the amount of storage designated to maintain data previously read.

This data can be scanned for string matches. The extended compression window size affects how much virtual memory the extended compression routines use. Higher window size values use more memory but achieve greater compression. The valid values for this subparameter are 8–15, inclusive. The default value is \*INITPARMS.

**ECMLEVEL** specifies the extended compression memory level parameter, which determines how much memory should be allocated for other internal data structures like the hash table and the previous table (pointers to previous strings starting with the same 3 characters). The extended memory level affects how much memory the extended compression routines use. Higher memory levels use more virtual memory but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is \*INITPARMS.

**EXCLUDE(generic, member-name, (start-range/stop-range), 'list')**

specifies criteria that identifies the file members that are not to be copied. EXCLUDE allows you to make exceptions to members specified generically or by range in the SELECT option. Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for the syntax and override priority for the SELECT and EXCLUDE parameters appropriate to the remote node.

**generic** specifies a generic member name. For example, if you specify CD\*, all member names beginning with CD are excluded.

**member-name** specifies an individual member name.

**start-range/stop-range** specifies first and last names in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). A slash (/) separates the first (start-range) and last (stop-range) member names. The two members specified, as well as all members in the range, are excluded.

**'list'** specifies any combination of the subparameters.

**FALIAS(\*YES)\*NO)**

specifies whether aliases are copied when their associated member names are copied.

**FBUFND(n)**

specifies the number of I/O buffers VSAM uses for transmitting data between virtual and auxiliary storage. (A buffer is the size of a control interval in the data component.) Valid values range from 1–510. The default is 2. Increasing this number generally improves the I/O performance on the file but requires more memory. This parameter is valid only when the source side is an z/OS node.

**FDBPARMS(dbid, dbsubid)**

indicates that the Connect:Direct for z/OS SQL SERVER is to process the I/O request against the database identified by the subparameters of this keyword and defined by the SQL= parameter. The DBPARMS parameter is used with the optional Connect:Direct for SQL feature. This parameter is valid only when the source side is a z/OS node. Only DB2 is supported.

To determine the values to code to access a given DB2 table or set of DB2 tables, consult your database administrator.

**dbid** specifies the 1–4 byte DB2 database ID.

**dbsubid** specifies the 1–4 byte DB2 database sub-ID.

**FDCB(modeldcb, blksize, den, dsorg, keylen, linct, lrecl, optcd, recfm, rkp, trtch)**

overrides attributes of the source file. DCB attributes are determined from the operating system unless specified.

**modeldcb** specifies a model data set control block (DSCB).

**blksize** specifies the length in bytes of the block. The minimum length is 18 bytes, and the maximum length is 32,760 bytes.

**den** specifies the magnetic tape mode setting. The following table shows the values for the DEN parameter for 7- and 9-track tapes. When coded together, the DEN and TRTCH values are used to select a tape device for allocation by Connect:Direct for z/OS.

DEN	7-Track Tape	9-Track Tape
1	556 bpi	--
2	800 bpi	800 bpi
3	--	1600 bpi
4	--	6250 bpi

**dsorg** specifies the file organization. Supported file organizations are BDAM, ISAM, PO, PS, and VSAM.

**keylen** specifies the length of the keys used in a file. Maximum length in bytes is 255.

**linct** specifies the blocks or tracks searched to find a free block or available space.

**lrecl** specifies the record length in bytes.

**optcd** specifies optional processing associated with this file. This specification only applies to this file and is not automatically applied to the other files involved in the RECEIVE operation.

Valid options are:

**W** performs write validity checks on direct access storage devices.

**Q** performs ASCII-to-EBCDIC conversion for input files and EBCDIC-to-ASCII conversion for output files. **Q** is the default and only used for AL-labeled tape files.

**Z** performs reduced error recovery for tape files.

**recfm** specifies the format of the records in the file. You can specify any valid record format, such as:

**F** specifies fixed format.

**FA** specifies fixed format with ASA printer control.

**FB** specifies fixed block format.

**FBA** specifies fixed block format with ANSI carriage control.

**FM** specifies fixed format with machine code control character.

**U** specifies undefined format.

**V** specifies variable format.

**VB** specifies variable block format.

**VBA** specifies variable block format with ASA printer control.

**VBM** specifies variable format with block machine code control character.

**VS** specifies variable spanned format.

**VBS** specifies variable block spanned format.

For FDR volumes, you must specify RECFM=U.

**rkp** specifies the position of the first byte of the record key within each logical record. The beginning byte of a record is addressed as **0**.

**trtch** specifies the magnetic tape mode setting. When coded together, Connect:Direct for z/OS uses the TRTCH and DEN values to select a tape device for allocation. Valid options are:

**C** specifies data conversion, odd parity, and no translation.

**E** specifies no data conversion, even parity, and no translation.

**T** specifies no data conversion, odd parity, and BCD or EBCDIC translation.

**ET** specifies no data conversion, even parity, and BCD or EBCDIC translation.

**COMP** is a feature for 3480X tape drives only. It enables Improved Data Recording Capability (IDRC), which compresses the data. This subparameter overrides the system-wide IDRC setting for no compression. If you are specifying COMP, you must also include a UNIT= parameter that specifies either 3480X or a systems programmer-defined name equivalent to a 3480X tape drive.

**NOCOMP** overrides the system-wide IDRC setting for compression. NOCOMP applies to 3480X tape drives only.

#### **FDISP(current, normal end, abnormal end)**

specifies the status of the remote file and what is to be done with the file after notification of successful transmission.

**current** specifies the status of the file. Valid values are:

**SHR** specifies that the file can be read simultaneously by another operation. SHR is the default.

**OLD** specifies that the operation is to be given exclusive control of the file.

**normal end** specifies the disposition of the file following normal operation termination. Valid dispositions are as follows:

**KEEP** specifies that the system is to keep the file after the operation completes.

**DELETE** specifies that the system is to delete the file after the operation completes.

**abnormal end** specifies the disposition of the file after abnormal operation termination. Valid source file dispositions are:

**KEEP** specifies that the system is to keep the file after the operation terminates abnormally.

**DELETE** specifies the system is to delete the file if the operation terminates abnormally.

For subparameters for other Connect:Direct platforms, refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

---

**Note:** Currently, normal end disposition and abnormal end disposition are not applicable for Connect:Direct for i5/OS.

---

#### **FIOEXIT(program-name)**

specifies the name of the user-written program to be given control for I/O-related requests for the associated data.

#### **FIOPARM(parameter-name)**

specifies a parameter, or list of parameters, to be passed to the specified exit. This parameter is valid only when the source side is a z/OS node. The FIOPARM syntax depends on the parameter list that the program specified in FIOEXIT requires.

#### **FLABEL(seqno, labeltype, pswdread, inout, retpd|expdt)**

specifies label information for the tape.

**seqno** specifies the relative file position on the tape.

**labeltype** is designated as:

**SL** specifies IBM standard labels.

**AL** specifies American National Standard labels.

**BLP** specifies bypass label processing.

**LTM** specifies bypass leading tape marks.

**NL** specifies no labels.

**pswdread** specifies that a password must be supplied by the operator or user before the file can be accessed.

**inout** specifies that a BSAM data set opened for INOUT or a BDAM data set opened for UPDAT is to be read only.

**retpd** specifies the retention period for the data set in days, where *nnnn* is 1–4 digits.

**expdt** specifies the expiration date for the data set, where *yyddd* or *yyyyddd* is a valid Julian date.

**FLINK(vmuserid,pwd,accmode,ccuu)**

specifies the disk where the CMS file is located. This parameter allows the user to access the CMS file. This parameter is valid for VM nodes only.

**vmuserid** specifies the owner ID for the CMS minidisk where the file is located. The valid length ranges from 1–8 characters.

**pwd** specifies the appropriate password for the CMS minidisk where the file is located. The maximum length is 256 characters. The default password is ALL.

**accmode** specifies the link access mode.

When used with the FROM parameter, the access modes are **W** (primary read/write access), **M** (primary multiple access), **R** (primary read only), **RR** (primary and secondary read only access), **WR** (primary read/write access; alternate read only access), **MR** (primary multiple access; alternate read only access), and **MW** (primary multiple access; alternate read/write only access).

When used with the TO parameter, the access modes are **W**, **M**, and **MW**.

**ccuu** specifies the virtual address of the disk where the CMS file is located. Any three-digit number is valid.

**FMSVGP(data-set-name)**

specifies the remote file mass storage volume group that resides on a mass storage system (MSS) device. This group must be a valid DD (data definition) name. This parameter is valid when the source side is a z/OS node only.

**FPDSDIR(\*YES|\*NO)**

specifies whether the user-related information in the directory is sent.

**FSQL(sql-statement|DSN=data-set-name)**

defines the SQL statement to be processed against the database identified by the DBPARMS parameter or the DSN containing the SQL statements. The SQL parameter is used with the optional Connect:Direct for z/OS SQL feature. This parameter is valid only when the source side is an Connect:Direct for z/OS node.

**sql-statement** defines the actual SQL statement to be processed.

**data-set-name** specifies the data set containing the SQL statements to be processed.

**FTYPE(entry-name)**

specifies the entry name of the type defaults file on the remote node that contains the default file attributes. Specify this typekey only when the user requests defaults.

**FUNIT(name)**

specifies the unit address, device type, or user-assigned group name where the file resides or will reside. For SAM-to-SAM copies where the destination file is new and the UNIT parameter is not coded, the device type from the source file is used. For VSE nodes, you must specify either this parameter or FVOL.

Specify a unit-count to allow additional units to be allocated if required, or specify **P** to allocate the same number of units as volumes and then parallel mount the volumes. This parameter is valid only when the source side is a z/OS node.

**FVOL(private, volcount, ser)**

specifies the volume serial numbers containing the file and optional processing associated with the file. If VOL is not specified, the file must be cataloged. For VSE nodes, you must specify either this parameter or FUNIT.

**private** specifies allocation of an output file only if the volume is specifically requested and is used for direct access storage devices only. Valid values are \*YES and \*NO.

**volcount** specifies the maximum number of volumes required by an output file.

**ser** identifies by serial number the volumes on which the output file resides or will reside.

**FVOLREF(data-set-name)**

allows you to retrieve the specified data set from the same volume as the referenced data set. You must catalog it on the system where it is referenced. This parameter is valid only when the source side is a z/OS node.

**FVSAMCAT(dsn,vmid,pwd,accmode,uu)**

specifies the catalog for the VSAM file to be copied. This parameter is required only if using a catalog other than the master catalog.

**dsn** specifies the filename of the VSAM catalog containing the file to be copied. The maximum length is 44 characters.

**vmid** specifies the owner ID for the VSAM minidisk where the file is located. The maximum length is eight characters.

**pwd** specifies the appropriate password for the VSAM minidisk where the file is located. The maximum length is eight characters.

**accmode** specifies the link access mode. Valid access modes are NULL, **W** (primary read/write access), **M** (primary multiple access), and **MW** (primary multiple access; alternate read/write only access).

**uu** specifies the virtual address of the disk where the VSAM file is located. Any three-digit number is valid.

**NOTIFY(userid)**

specifies the user ID on the local node to receive status messages that are directed to the user's message queue.

**PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to run a task, for example TAPE.

---

**Note:** This parameter is for use only in Connect:Direct OS/390 environments.

---

**PNAME(process-name)**

identifies the Process in any messages or statistics. The name can be from 1 to 12 characters long. The first character must be alphabetic. The default name is CDRCV.

---

**Note:** Only Connect:Direct for i5/OS and Connect:Direct for UNIX support Process names with more than 8 characters.

---

**PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local i5/OS node. Use this parameter only to validate security with a different ID than the one you used to sign on to the system.

**id** specifies the security ID passed to the security system at the remote node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. The security system can use this parameter at the remote node to validate the current security password (1–10 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password. Connect:Direct for i5/OS does not support this parameter.

**REPLACE(\*YES|\*NO)**

specifies that the remote file or PDS replaces members of the same name at the local file.

**RMTWORK(\*YES|\*NO)**

specifies whether a remote Process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*YES** allows a remote Process to run on the session after Connect:Direct for i5/OS completes its Process.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function.

**SACCT(snode-accounting-data)**

specifies the accounting data for the remote node. 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.

**SECOPTIONS(\* \*)**

specifies Encryption and Digital Signature values.

**Encryption** specifies if data encryption is to be performed and what algorithm to use.

**Digital Signature** specifies if a digital signature is created while a file is being transmitted.

---

**Note:** The SECOPTIONS parameter is valid only if the Connect:Direct Secure+ Option is installed. For information about these parameters, refer to the *Connect:Direct Secure+ Option for i5/OS Implementation Guide*.

---



**SELECT(member, new-name, R/NR, 'list')**

specifies selection criteria by which file members are to be copied. Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for the syntax and override priority for the SELECT and EXCLUDE parameters appropriate to the remote node.

Use this parameter when you specify TYPE(FILE) in the TOSYSOPTS parameter.

The SELECT parameter consists of three positional subparameters separated by commas.

**member** specifies the members to be copied. Member selections are:

\* represents a global generic. A global generic indicates that all members of the file are to be included.

**generic** specifies a generic member name. For example, if you specify CD\*, all member names beginning with CD are selected for copying.

**member-name** specifies an individual member name.

**start-range/stop-range** specifies first and last names in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). A slash (/) separates the first (start-range) and last (stop-range) member names. The two members specified, as well as all members in the range, are copied.

**new-name** specifies a new name for a member. You cannot use the new-name subparameter if a generic or range is used to select the members to be copied, or if you specify a list of member names.

**R/NR** specifies whether a copied member replaces an existing member of the same name in the local file. When used with a generic or a range, this subparameter applies to all members selected.

---

**Note:** When using a generic and specifying NR or R, the second positional subparameter (new-name) must be null, for example, member-name,,NR.

---

**R** specifies that a member replaces an existing member of the same name in the local file. When used with the new-name subparameter, **R** applies to the specified new-name and not the original member name.

**NR** specifies that a member does not replace an existing member of the same name in the local file. NR overrides the REPLACE parameter.

**'list'** specifies any combination of the subparameters. You can specify a list of individual member names or generic member names separated by spaces. However, you cannot use the new-name and replace-member subparameters when specifying a list.

---

**Note:** Member names longer than eight characters are truncated to eight characters when the remote node is not an i5/OS node.

---

**SNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the remote node. If nothing is specified for this parameter, the user profile of the user who submits the command is sent to the remote node with a blank password.

**id** specifies the security ID passed to the security system on the remote node (1–64 alphanumeric characters).

**pswd** specifies the current security password. The security system can use this parameter on the SNODE to validate the current security password (1–64 alphanumeric characters). This is optional unless the user has set security to require a password.

**newpswd** specifies the new security password. Connect:Direct for i5/OS does not currently support this parameter if the remote node is an i5/OS node.

---

**Note:** Parameter values are case sensitive.

---

### **TDISP(current)**

specifies the status of the file on the local i5/OS node.

**current** specifies the status of the file before the operation executes. Valid values for this subparameter are:

**NEW** specifies that the operation creates the destination file or member. If you are creating a member, the file can already exist but the member must not already exist. If you are creating a file (SAVF), the file must not already exist.

**OLD** specifies that the destination file or member already exists. The operation has exclusive control of the member. If the file or member does not exist, the receive operation fails.

**MOD** specifies that the operation modifies the file or member by adding data to the end of the member or, if none exists, allocates a new member. The file is also created if it does not exist.

**RPL** indicates that the destination file or member is replaced. The file is created if it does not exist.

**SHR** indicates that another operation can read the file simultaneously. The file or member is created if it does not exist on i5/OS.

---

**Note:** Currently, normal end disposition and abnormal end disposition are not applicable when sending to Connect:Direct for i5/OS.

---

### **TRACEOPT**

#### **Trace Instructions**

(L, S, F, C, X, K, D, H, V, W, P|\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)

- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

#### Command Trace Options

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

---

## Using Block Mode Transfer

Connect:Direct for i5/OS supports block mode transfer of COPY operations. In this mode, Connect:Direct uses the file block as the data unit. Using the block mode transfer improves performance because the remote and local nodes handle fewer data units. This type of transfer reduces instruction path length and decreases execution time. Larger units move between Connect:Direct and the file access routines, requiring fewer calls to the access routines and further improving performance.

Connect:Direct for i5/OS simulates block reads and writes. Using logical blocks instead of physical blocks, Connect:Direct for i5/OS allows you to modify the blocksize to match the other node in some cases.

Connect:Direct for i5/OS supports block mode transfer when the following conditions exist:

- ❖ Remote node supports block mode
- ❖ Source and destination files have like characteristics
- ❖ Files are supported file types

Connect:Direct for i5/OS supports block mode transfers for the following i5/OS file types:

- ❖ DB2 for i5/OS files (without staged compression)
- ❖ Spool files
- ❖ Save files
- ❖ Stream files (using the integrated file system)

## Simulating Blocksizes

The simulated blocksize determines whether a block mode transfer occurs. If the block mode transfer does occur, the simulated blocksize is used for the transfer. The following rules determine what simulated blocksize Connect:Direct for i5/OS uses.

As the local node, Connect:Direct for i5/OS determines the simulated blocksize by these conditions in order of precedence:

- ❖ Destination file blocksize specified on the request
- ❖ Source file blocksize sent by the remote node in the COPY control block

If neither condition exists, Connect:Direct uses the record mode.

As the remote node, Connect:Direct for i5/OS determines the simulated blocksize by these conditions in order of precedence:

- ❖ Source file blocksize specified on the request
- ❖ Destination file blocksize specified on the request
- ❖ Calculated value\*

\*Largest multiple of record size less than some arbitrary value, such as 4096.

## Block Mode Transfer Examples

The following block mode transfer examples include both sending and receiving examples.

### CDRCV Examples-Not Specifying Source File Blocksize

In the first example, the request does not specify the source file blocksize. In this scenario, the following occurs:

- ❖ The Connect:Direct for i5/OS node sets the source file simulated blocksize to 3200 by using the destination file value.
- ❖ The receiving Connect:Direct for i5/OS sets the destination file simulated blocksize to the value specified on the request.
- ❖ Because the blocksizes match, the block mode copy occurs.

```
C:D i5: receiving
request: source blksize=not specified; dest blksize = 3200
srce file: lrecl=80, (on another C:D i5 3.3 node)
dest file: NEW
```

### CDRCV Example-Specifying Source File Blocksize

In the second example, the following occurs:

- ❖ Connect:Direct for i5/OS calculates the simulated blocksize at 3200 because the destination file is new and the source file specifies a blocksize.
- ❖ The receiving node uses the existing destination file blocksize of 3200.
- ❖ A block mode transfer occurs because the blocksizes match.

```
C:D i5: receiving
request: source blksize=not specified;dest blksize = not specified;
srce file: lrecl=80, blksize=3200 (on z/OS node)
dest file: NEW
```

## CDSND Examples-Simulating Blocksize

In the third example, the following events occur:

- ❖ Connect:Direct for i5/OS sets the simulated blocksize at 6400 from the request.
- ❖ The receiving node creates the destination file using source file attributes of blksize=6400.
- ❖ Because the file attributes match, the block mode transfer occurs.

```
C:D i5: sending
request: source blksize=6400; dest blksize = not specified;
srce file: lrecl=80
dest file: NEW (on z/OS)
```

## CDSND Examples-Setting Actual Blocksize

In the fourth example, the following events occur:

- ❖ the remote node sets the destination file blocksize to 3200, the actual blocksize of the destination file.
- ❖ Connect:Direct for i5/OS calculates the simulated blocksize at 3200.
- ❖ The blocksizes are identical, so block mode transfer occurs.

```
C:D i5: sending
request: source blksize=not specified; dest blksize = not specified;
srce file: lrecl=80
dest file: existing: lrecl=80, blksize=3200 (on z/OS)
```

## CDSND and CDRCV Example

In the fifth example, Connect:Direct for i5/OS calculates the simulated blocksize at 32720 since the copy is between two Connect:Direct for i5/OS nodes.

```
C:D i5 sending and receiving
request: source blksize=not specified; dest blksize=not specified
srce file: lrecl=32720
dest file: new
```

---

## Using the CDRCV Screens

Use the **Connect:Direct Receive File (CDRCV)** screens to initiate a Connect:Direct session and receive a file from a remote node. CDRCV uses certain special function keys as follows:

- ❖ **F24** more keys gives you access to additional function keys used in CDRCV.
- ❖ **F5** refresh which clears any field entries you have made.
- ❖ **F9** which gives you access to all parameters.
- ❖ **F11** which displays key words.
- ❖ **F14** which displays command strings.
- ❖ **F15** which displays any error messages pending.
- ❖ **F16** which marks the command complete.

## Receiving a File From a Remote Node

Complete the following steps to receive a file from a remote node.

1. Access the **CDRCV** screens by typing **CDRCV** at the command line and pressing **F4** or by choosing the receive file option from the Connect:Direct main menu.
2. Type in the remote node name at the screen prompt and press **Enter**. The command retrieves the remote node environment and display fields for required parameters.
3. Type values for required and optional parameters as appropriate and press **Enter**.

The following are samples of the screens in the **Connect:Direct Receive File (CDRCV)** screen flow. The sample shows the keyword display option.

### Connect:Direct Receive File (CDRCV) Screen 1

```
Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

Sending Node Name . . . . . _____
Sending Node Environment . . . . . _____ OS/400, AS400, MSP, MVS...
Sending File Name . . . . . _____
_____
Receiving File Name . . . . . _____
_____
Receiving System Operations . . 'TYPE(FILE) ' _____
_____
_____
_____
_____
_____
_____

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

More...
```

The content of this screen and the screens that follow depend on the remote node environment.

### Connect:Direct Receive File (CDRCV) Screen 2

```

Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

Sending System Operations . . . .
.'TYPE(FILE)'  

_____  

_____  

_____  

_____  

_____  

_____  

Trace Options:
  Trace Instructions . . . . . *CURRENT
  Command trace options . . . . *NONE
  Process Name . . . . . CDRCV      Name
Receiving Node User Info.:
  User Id . . . . . *CURRENT      Character value, *CURRENT
  Password . . . . .              Character value
  New Password . . . . .          Character value
  User to Notify . . . . . *CURRENT      Name, *CURRENT, *NONE
                                          More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

### Connect:Direct Receive File (CDRCV) Screen 3

```

Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

Sending Node User Info.:
  User Id . . . . . _____
  _____
  Password . . . . .
  _____
  New Password . . . . .
  _____
  Class . . . . . *MODED      1-255, *MODED
  Sending Accounting Data . . . . _____
  _____
  _____
  Checkpoint Interval . . . . . *INITPARMS  0-9999999, *INITPARMS
  Allow remote process . . . . . *NO        Character value, *YES, *NO
                                          More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

### Connect:Direct Receive File (CDRCV) Screen 4

```

Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

File Members Selection List  . . _____
_____
_____

File Members Exclusion List  . . _____
_____
_____

Same Member Replaced . . . . . *YES          *YES, *NO
CRC . . . . . *INITPARMS      *INITPARMS, *NETMAP, *ON, *OFF
Secure+ Options:
  Encryption . . . . . *          */Y/N/Algorithm
  Digital Signature . . . . . *          *, N, Y
  Compress Data . . . . . *NO         Character value, *NO, *YES...

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

### Connect:Direct Receive File (CDRCV) Screen 5

```

Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

Extended compression tuning:
  Extended compression level . . *INITPARMS  1-9, *INITPARMS
  Extended compression window . *INITPARMS  8-15, *INITPARMS
  Extended compression memory . *INITPARMS  1-9, *INITPARMS
Sending File Type Parameter . . Character value
Receiving File Disposition:
  Current Disposition . . . . . NEW          NEW, OLD, MOD, RPL, SHR
  Normal End Disposition . . . . CATLG        CATLG, KEEP
  Abnormal End Disposition . . . _____  KEEP, CATLG, DELETE

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```



### Connect:Direct Receive File (CDRCV) Screen 6

```
Connect:Direct Receive File (CDRCV)

Type choices, press Enter.

Sending DCB Info.:
Model File Name . . . . . _____

-----
Block Size in bytes . . . . . _____ Character value
Tape Mode Density Setting . . . . . ___ 1, 2, 3, 4
Organization of File . . . . . - BDAM, ISAM, PO, PS, VSAM...
Key Length of file . . . . . 000 Character value
Searching Free Block/Space . . . . . 00000 Character value
Record Length in bytes . . . . . 00000 Character value
Process Option . . . . . - W, Q, Z
Record Format . . . . . _____ F, FA, FB, FBA, FM, U, V...
Position of 1st Byte of Key . . . . . - Character value
Tape Mode Setting . . . . . _____ C, E, T, ET, COMP, NOCOMP

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```



## Sending a Spool File

This chapter describes the Connect:Direct Send Spool (CDSNDSPL) command and its function. This chapter also shows sample **CDSNDSPL** screens and describes the required and optional parameters for the CDSNDSPL command.

### Using the CDSNDSPL Command

Use the CDSNDSPL command to send a spool file on the local i5/OS node to a remote node. You can also send spool files to a remote printer through the Connect:Direct for i5/OS Spool Monitor. You can specify a user-exit command (either **EXITCMD** or **FAILCMD**) to be executed after the completion of a copy process in either **FROMSYSOPTS** or **TOSYSOPTS**. Using **EXITCMD**, you can specify a command to be executed only if the copy process is successful. You can also specify a command to be executed only if the copy process is not successful by using the **FAILCMD** command. The **EXITCMD** and **FAILCMD** are executable i5/OS commands as well as Connect:Direct as long as the size of the command does not exceed 127 bytes. Refer to the *Maintaining Spool Support* chapter of the *Connect:Direct for i5/OS Installation and Administration Guide*.

### CDSNDSPL Command Example

The command that follows is an example of using the CDSNDSPL command.

```
CDSNDSPL SNODE(CD2200) SNODENVIRN(AS400) SPLF(CDLOG) JOB(CDSMGR          CDADMIN
nnnnn) SPLFNUM(1) TDSN('qa/splfout(splfout)')          TOSYSOPTS('TYPE(mbr)')
TDISP(RPL)
```

### CDSNDSPL Command Format

The following parameters are for the CDSNDSPL command. Required parameters are shown in bold type.

Command	Parameters
CDSNDSPL	<b>SNODE(remote-node-name)</b>
	<b>SNODENVIRN(remote-environment-name)</b>
	<b>SPLF(name)</b>

† Connect:Direct Secure+ Option for i5/OS parameter.

Command	Parameters
	<b>JOB</b> (*  <b>job-name user-name job-number</b> )
	<b>SPLFNUM</b> (n * <b>ONLY</b>  * <b>LAST</b> )
	<b>TDSN</b> ( <b>spooled-file-name data-set-name</b> )
	<b>TOSYSOPTS</b> ( DECMPR(* <b>YES</b>  * <b>NO</b> ) EXITCMD(valid i5/OS command) FAILCMD(valid i5/OS command) TYPE( <b>FILE</b>   <b>MBR</b>   <b>OBJ</b>   <b>SPLF</b> ) CODEPAGE(nnnnn,nnnnn) XTRAN(table-name)[XTRANLSO(so-code)]XTRANLSI(si-code)]XTRANLDATA( <b>MIXED</b>   DBCS)XTRANPAD(pc)] CCSID(nnnnn) EORCHAR(xxxx) ')
	<b>Note:</b> TOSYSOPTS is required if remote node is on i5/OS
	<b>BLKSIZE</b> (n * <b>CALC</b> )
	<b>CHLVAL</b> (* <b>NORMAL</b>  ((channel#1 line#1)...(channel#12 line#12))
	<b>CKPTINV</b> (n)
	<b>CLASS</b> (n)
	<b>COMPRESS</b> (* <b>YES</b>  * <b>NO</b>  X xx C cc EXT)
	<b>CRC</b> (* <b>INITPARMS</b>  * <b>NETMAP</b>  * <b>OFF</b>  * <b>ON</b> )
	<b>CTLCHAR</b> (* <b>NONE</b>  * <b>FCFC</b>  * <b>PRTCTL</b> )
	<b>ECTUNING</b> (ecclevel,ccminsize,ecmlevel)
	<b>FDSNLEN</b> (* <b>SPLF</b> )
	<b>NOTIFY</b> (userid)
	<b>PLEXCLASS</b> (user-defined class)
	<b>PNAME</b> (name)
	<b>PNODEID</b> (id, pswd, newpswd)
	<b>PRTY</b> (n)
	<b>RMTWORK</b> (* <b>YES</b>  * <b>NO</b> )
	<b>SACCT</b> (snode-accounting-data)
	‡ <b>SECOPTIONS</b> (* *)
	<b>SNODEID</b> (id, pswd, newpswd)
	<b>TBUFND</b> (n)
	<b>TDBPARMS</b> (dbid,dbsubid)
	<b>TDCB</b> (modelcdb, blksize, den, dsorg, keylen, limct, lrecl, optcd, recfm, rkp, trtch)
	<b>TDISP</b> (current  normal end  abnormal end)
	<b>TIOEXIT</b> (program-name)

‡ Connect:Direct Secure+ Option for i5/OS parameter.

Command	Parameters
	TIOPARM(parameter-name)
	TLABEL(seqno,labeltype,pswdread,inout,retpd,expdt)
	TLINK(vmid,pwd,accmode,cuu)
	TMSVGP(data-definition-name) (For z/OS nodes only)
	TPROTCT(*YES *NO)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS *CURRENT Command Trace Options: L,S,F,C,X,K,D,H,V,W,P *ALL *EXT *COMMON  *NONE)
	TSPACE(avgbk,type,primary,secondary,pdsblk,rise, contig,round) (For z/OS nodes only)
	TSQL(sql-statement DSN=data-set-name) (For z/OS nodes only)
	TTYTYPE(name)
	TUNIT(unit-name)
	TVOL(*YES *NO)
	TVOLREF(data-set-name) (For z/OS nodes only)
	TVSAMCAT(dsn vmid pwd accmode cuu)

† Connect:Direct Secure+ Option for i5/OS parameter.

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

## Required Parameters

Unless otherwise noted, these required parameters apply to all remote node environments.

### **SNODE(remote-node-name)**

specifies a 1–16 character alphanumeric name for the remote node.

**Note:** This value is case sensitive.

### **SNODENVIRN(remote-environment-name)**

specifies the remote node environment.

### **SPLF(name)**

specifies the name of the spooled output file to be copied.

### **JOB(\*job-name user-name job-number)**

specifies the name of the job that created the spooled output file. Valid values are:

\* specifies that the job issuing the command is the job that created the spooled output file.

**job-name** specifies the name of the job that created the spooled-output file.

**user-name** specifies the user name that identifies the user profile for the job.

**job-number** specifies the system-assigned job number.

**SPLFNUM(n|\*ONLY|\*LAST)**

specifies the spooled output file number from the job whose data records are to be sent. Valid values are:

**1-9999** indicates to copy the specified spooled output file number.

**\*ONLY** indicates that only one of the job's spooled output files has the specified file name.

**\*LAST** indicates to copy the highest-numbered spooled output file with the specified job name.

**TDSN(spooled-file-name|data-set-name)**

specifies the remote file name. The length of the input field displayed is 132 characters, by default. To increase the length of the field to 256 characters, type an ampersand(&) followed by a blank in the first two positions of the field.

---

**Note:** Do not use member names longer than eight characters when sending to z/OS nodes. The results are unpredictable.

---

If the remote node is a Connect:Direct for i5/OS node, TDSN specifies the destination file name. File names are verified based on the i5/OS standard file name conventions. If you are using Connect:Direct for i5/OS with the **F4** prompt feature, the system supplies the single quotes if you fail to type them with the file name on the command line.

**spooled-file-name** specifies the name to be used for the spooled file when placed in the Connect:Direct out queue on the remote system. Use this parameter when you specify ('TYPE(SPLF)') in the TOSYSOPTS.

**data-set-name** specifies the remote file name to be copied. The source file member name is used for the destination database file member name when you use TOSYSOPTS('TYPE(FILE)'), unless you specify otherwise with the SELECT parameter. The file name is used for the member name when you use TOSYSOPTS('TYPE(MBR)').

---

**Note:** For TDSN, the values of \*LIBL, \*CURLIB, and QTEMP cannot be used when sending to an i5/OS system. A specific library name must be used for copying to a data set on the i5/OS.

---

For additional information about valid platform-specific parameters, refer to the Connect:Direct Processes Web site at

<http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

The following table summarizes the data set name requirements for Connect:Direct platforms.

SNODENVIRN	TDSN	Format	Special Considerations
OS/400	Native File System	'library-name/file-name' 'library-name/file-name (member-name)'	Enclose in single quotation marks.
	Integrated File System	'/QSYS.LIB/library-name.LIB /file-name.FILE /member-name.MBR' or '/directory/file-name  /QLANSrv/file-name  /QDLS/folder-name  /QOpenSys/file-name'	You cannot specify the *SNDFFD, *SELECT, and *EXCLUDE parameters with the integrated file system.

SNODENVIRN	TDSN	Format	Special Considerations
		Spool File Support spooled-file-name	Do not use member names longer than eight characters when sending to z/OS, VSE, and MSP. The results are unpredictable.
OS/390, VSE, MSP		Standard z/OS DSN naming conventions.	Single quotes allow special characters. Do not use member names longer than eight characters. The results are unpredictable.
VM		Fully qualified file-name	
Tandem		\system.\$volume.subvolume.filename	
Windows		drive:\directory\subdirectory\filename or \servername\sharepointname\subdirectory\filename	
UNIX		/directory/subdirectory/filename	
OpenVMS		node::device:[directory]filename.type;version	When you are copying a file, the file specification must be enclosed in single quotation marks if it uses spaces or other special characters not recognized by the other node.
Stratus VOS		Fully qualified path and file name	Enclose the file name in double quotation marks when you use a reserved word (statement name or keyword) for the file name.

**TOSYSOPTS(‘**

```

DECMPR(*YES|*NO)
EXITCMD(valid i5/OS command)
FAILCMD(valid i5/OS command)
TYPE(FILE|MBR|SPLF)
XTRAN(table-name) [XTRANLSO(so-code)|
XTRANLSI(si-code)|XTRANLDATA(MIXED|DBCS)]
CCSID(nnnnn)
EORCHAR(xxxx)
’)

```

specifies system operation parameters at the remote node. The maximum number of characters permitted for TOSYSOPTS is 2K. This parameter is only required when the remote file is on another Connect:Direct for i5/OS node.

For additional information about valid SYSOPTS parameters, see the appropriate COPY statement chapter of the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

**EXITCMD** specifies a command to be executed only if the copy process is successful.

**FAILCMD** specifies a command to be executed only if the copy process is not successful.

**DECMPR(\*YES|\*NO)** states whether the remote Connect:Direct should decompress the received file. **DECMPR(\*NO)** instructs the remote system to place the received data in a database file without

decompressing it. The option `DECMPR(*NO)` cannot be specified if you are using Secure+ Option to perform a secured copy function.

---

**Note:** Ensure that the destination file contains no members before you send a file with `DECMPR(*NO)` specified.

---

**TYPE(FILE|MBR|SPLF)** specifies the type of data to be sent. You *must* define the type of data to be copied as one of the following:

**FILE** specifies that the file being copied is a physical database file.

**MBR** specifies that the object being copied is a member of a physical database file.

**SPLF** specifies that the data is to be copied to an i5/OS spooled output file.

For each TYPE, you can specify certain parameters as follows.

The following parameters are valid for **TYPE(FILE)** and **TYPE(MBR)** files:

**RCDLEN(record-length)** specifies the data length for each record in the file.

**FILETYPE(\*SRC|\*DATA)** establishes the type of file to be created. Use this parameter whenever a file is created.

**TEXT('text description')** sets a text description to be associated with this member (and file, if created). This description cannot exceed 50 characters and must be enclosed in single quotes.

**EXPDATE(expiration-date)** specifies the date after which the new or replaced member is no longer valid. If `EXPDATE` is not specified, then the file does not have an expiration date.

**MAXMBRS(number|\*NOMAX)** specifies the maximum number of members a physical file can contain.

**SIZE(#-of-recs, incr-value #-of-incrs|\*NOMAX)** is used when a new file is created for the file received.

**AUT(\*CHANGE|\*ALL|\*USE|\*EXCLUDE)** specifies the authority to give to a user who does not have specific authority to the file or member, is not on the authorization list, and whose user group does not have specific authority to the file or member.

The following parameters are valid for **TYPE(SPLF)** files:

**DEV(\*JOB|\*SYSVAL|device-name)** specifies the name of the printer device description.

**DEVTYPE(\*IPDS|\*SCS|\*USERASCII)** sets the type of data stream created for a printer file from the data received.

**PAGESIZE(page-length page-width)** specifies the length and width of the page used by the printer. (This parameter overrides the `FORMTYPE` parameter.)

**LPI(3|4|6|7.5|8|9)** sets the line space setting (lines per inch) on the printer.

**CPI(5|10|12|13.3|15|16.7|18|20)** specifies the printer character density, in characters per inch, for the printer.

**OVRFLW(overflow-line-number)** specifies the line number on the page at which overflow to a new page begins.

**FOLD(\*NO|\*YES)** specifies whether entire records are printed when the record length exceeds the form width.

**RPLUNPRT(\*YES 'replacement-character'|\*NO)** specifies whether unprintable characters are replaced with printable characters when printed. Also specifies the replacement character.

**ALIGN(\*NO|\*YES)** specifies whether the page must be aligned in the printer before printing is started.



**CTLCHAR(\*NONE|\*FCFC)** specifies whether the data contains printer control characters.

**CHLVAL(\*NORMAL| (channel#1 line#1) ... (channel#12 line#12))** specifies the list of channel numbers with their assigned line numbers. Note that CTLCHAR(\*FCFC) must be specified as a SYSOPTS parameter for this to be a valid parameter.

**FORMFEED(\*DEV|\*CONT|\*CUT|\*AUTOCUT)** specifies the form feed attachments used by the printer (4214, 5219, and 5553 printers only).

**PRTQLTY(\*STD|\*DRAFT|\*DEV|\*NLQ)** specifies the quality of print produced.

**DRAWER(1|2|3|\*E1)** specifies the source drawer to be used when you use automatic cut-sheet feed mode.

**FONT(\*CPI|\*DEV|font-identifier)** specifies the font identifier to be used for the spooled output file.

**CHRID(\*DEV|\*SYSVAL| graphic-character-set code-page)** specifies the character identifier to use for the spooled output file.

**PAGRTT(\*DEV|\*COR|0|90|180|270)** specifies the degree of rotation (clockwise from the edge of the paper first loaded into the printer) of text on each page printed.

**PRTTXT('print-text')** specifies a line of text to be printed at the bottom of each page printed. Up to 30 characters enclosed in single quotes is specified.

**JUSTIFY(0|50|100)** controls the print positions of the characters on the page (in the spooled file) so the right margin is regular.

**DUPLEX(\*NO|\*YES|\*TUMBLE)** specifies whether the spooled output file is printed on one or both sides of the paper.

**SPOOL(\*YES|\*NO)** specifies whether the data is sent to a spooled file prior to printing.

**OUTQ(\*JOB|\*DEV|library-name/output-queue- name)** specifies the output queue for the spooled output file created.

**FORMTYPE(form-type)** specifies the type of form to use in the printer when the spooled file is printed. The PAGESIZE parameter overrides this parameter.

**COPIES(number-of-copies)** specifies the number of copies to be printed. Valid values range from 1–255.

**MAXRCDS(maximum-records)** specifies the maximum number of records that are placed in the output queue. Valid values range from 1–500,000.

**FILESEP(number-of-file-separators)** specifies the number of blank separator pages to be placed between each copy of the file printed.

**HOLD(\*YES|\*NO)** specifies whether the file is to be held on the output queue until released by the user.

**SAVE(\*YES|\*NO)** specifies whether the spooled output file is to be saved on the output queue once printed.

**OUTPTY(\*JOB|output-priority)** specifies the scheduling priority of the file on the output queue.

**USRDTA(user-data)** allows up to 10 characters of data to identify the spooled output file.

Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for general information on SMS support.

**XTRAN( table-name)** specifies the translation table to use. The named table object must exist in a library that is in the session manager job library list or else the COPY step fails.

If the XTRAN keyword is present, then the following related optional keywords can also be used:

**XTRANLSO(so-code)** specifies extended translation local shift-out code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-out character and overrides the default value of 0E. You can specify any two valid hexadecimal digits.

**XTRANLSI(si-code)** specifies extended translation local shift-in code. This keyword specifies the hexadecimal code to use for the local (i5/OS side) shift-in character and overrides the default value of 0F. You can specify any two valid hexadecimal digits.

**XTRANLDATA(MIXED|DBCS)** specifies extended translation local data format. This keyword specifies the format of the data on the local (i5/OS side) of the COPY.

**MIXED** indicates that the data can contain both DBCS and SBCS characters and that SO/SI characters are used. MIXED is the default.

**DBCS** indicates that the data is pure DBCS characters and that no SO/SI characters are used.

**XTRANPAD(pc)** specifies that padding characters are in use. When DBCS data is translated from EBCDIC to ASCII, PAD specifies that the SHIFT-OUT and SHIFT-IN characters are replaced by a pad character to allow the displacement of fields within a record to remain unchanged during translation.

When DBCS data is translated from ASCII to EBCDIC, PAD specifies that the input ASCII DBCS file is in a padded format. The character immediately preceding a DBCS character or string is overlaid by the SHIFT-OUT character. The character immediately following a DBCS character or string is overlaid with the SHIFT-IN character. This allows the displacement of fields within a record to remain unchanged during translation.

**pc** is the pad character to be used during EBCDIC to ASCII translation. **pc** is ignored for ASCII to EBCDIC translations.

If XTRANPAD is not used, the SI/SO characters are stripped from the file when transferring from EBCDIC to ASCII, and added to the records when sending from ASCII to EBCDIC.

The following rules apply to the use of the XTRAN keyword:

- ❖ You must specify the XTRAN keyword to use extended translation all others are optional.
- ❖ The default for local shift-out is the IBM standard x0E.
- ❖ IBM standard x0F is the default for local shift-in.
- ❖ The default local data format is MIXED with SO/SI in use XTRAN is *not* allowed with PRECMPR(\*YES) or DECMPR(\*NO)
- ❖ To send a file to an ASCII platform using extended translation, you *must* specify in the TOSYSOPTS that the data type is binary. Refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for platform-specific SYSOPTS parameters.
- ❖ To use the XTRAN keyword to receive DBCS data from an ASCII platform, ensure that the record length of the file on the i5/OS side of the transfer is large enough to contain the translated record.

The record length of the file on the i5/OS side of the transfer is generally larger than the ASCII equivalent. For additional information, refer to the *Maintaining Extended Translation* chapter of the *Connect:Direct for i5/OS Installation and Administration Guide*

**EXITCMD** specifies a command to be executed only if the copy process is successful.

**FAILCMD** specifies a command to be executed only if the copy process is not successful.

**CODEPAGE(nnnnn,nnnnn)** converts data from one code page to another. The supported conversions are identified on the IBM Web site in the V5R1 Information Center, DB2 Universal

Database for iSeries SQL Reference, Appendix E. CODEPAGE() on Connect:Direct for i5/OS relates to CODEPAGE() developed for the z/OS system, UNIX, and Windows.

**nnnnn** identifies the IBM CCSID codepage number to convert from and the IBM CCSID code page number to convert to. If the FROM code page or TO code page is locale, the code page for the locale where CONNECT:Direct is executing is used. For example, if `fmsysopts(codepage(locale,1208))`, then the FROM code page is the code page for the locale and the TO code page is UTF-8.

**CCSID(nnnnn)** specifies the coded character set ID to use when creating IFS files. If no character set ID is specified, the system default is used.

**EORCHAR(xxxx)** specifies the two-to four-digit hexadecimal symbol to identify the character to use in text files as the end-of-record character, for example, `EORCHAR(0A)`.

## Optional Parameters

The following parameters are optional. Unless otherwise noted, the parameters apply to all remote node environments.

### **BLKSIZE(n)\*CALC)**

specifies the blocksize value for the source file. This value must be a multiple of the file record size.

**n** specifies the block size of the file. This value must be a multiple of the record size. The valid range is 1–32720

**\*CALC** indicates that the blocksize is calculated.

### **CHLVAL(\*NORMAL| (channel#1 line#1) . . . (channel#12 line#12))**

specifies the list of channel numbers with their assigned line numbers. Note that you must specify `CTLCHAR(*FCFC)` for this to be a valid parameter.

**\*NORMAL** indicates that channel 1 causes a skip to the next line, and channel 12 causes a skip to the overflow line (`OVERFLOW` parameter). Channels 2–11 cause a space-one-line operation.

**(channel#1 line#1) . . . (channel#12 line#12)** specifies channel number and line number. Any combination of channel numbers, 1–12, is specified along with a line number to be assigned to that channel number. Valid line numbers range from 1–255. If no line number is assigned to a channel number and that channel number is found in the data, a default of space-one-line before printing is taken. You can specify each channel and line number once.

### **CKPTINV(n)**

specifies the number of bytes, from 10K to 2 gigabytes, to send before taking a checkpoint. The format is `nnnnnnn`, `nnnnnnK`, or `nnnnnnM`, where **K** specifies thousands of bytes and **M** specifies millions of bytes. A value of **0** specifies no checkpoint restart. Connect:Direct for i5/OS uses the value you specify, rounded up to the nearest record boundary, to determine when to take a checkpoint.

A checkpoint interval specified here overrides the system-wide default value in the initialization parameters.

### **CLASS(n)**

determines the node-to-node session on which a Connect:Direct operation can execute. The **CLASS** is a number from 1–255. If you do not specify the **CLASS** parameter, it defaults to the number of maximum sessions defined in the communications environment mode description.

---

**Note:** The **CLASS** parameter is not used by Connect:Direct for i5/OS.

---

**COMPRESS (\*YES|\*NO|X\xx|C\cc|EXT)**

specifies data compression, which reduces the amount of data transmitted as the file is copied from one node to another. The file is automatically decompressed at its destination.

---

**Note:** Compression is CPU-intensive. Its effectiveness is dependent on data and line speed and should be used only if it benefits the installation.

---

If compression is specified, Connect:Direct reduces the amount of data transmitted based on the following rules:

- ❖ Repetitive occurrences (ranging from 2–63) of the primary compression character are compressed to one byte.
- ❖ Repetitive occurrences (ranging from 3–63) of any other character are compressed to two bytes.

Specifying the primary compression character is recommended for text data or single-character repetitive data. Extended compression generally offers improved compression for all types of data.

**\*NO** specifies no compression.

**\*YES** enables compression. If **\*YES** is specified, the default primary compression character is `PRIMEchar=X\40\` (hex 40=“ ”).

**X\xx\** enables compression, where *xx* is the hexadecimal representation of the value to be used for the primary compression character.

**C\cc\** enables compression, where *cc* is the character representation of the value to be used for the primary compression character.

**EXT** specifies that extended compression is used to search for repetitive strings of characters in data and compress them to codes that are transmitted and converted back to the original string during decompression. It is advantageous to specify this parameter when line transmission speeds are limited, CPU is available, and data is repetitive.

**CRC(\*INITPARMS|\*NETMAP|\*OFF|\*ON)**

Determines whether node and Process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to **\*OFF**, setting overrides for CRC checking will be ignored. The default value is **\*INITPARMS**.

**\*INITPARMS** specifies that the CRC setting is determined by the value for the initialization parameter, `TCP.CRC`.

**\*NETMAP** specifies that the CRC setting is determined by the value for the `NETMAP` parameter, Cyclic Redundancy Check. The value for the initialization parameter `TCP.CRC` must be set to **\*ON** and the value for the initialization parameter `TCP.CRC.OVERRIDE` must be set to **\*YES**.

**\*OFF** specifies that CRC is disabled.

**\*ON** specifies that CRC is enabled.

**ECTUNING (ecclevel,ecwsize,ecmlevel)**

**ECCLEVEL** specifies the extended compression level, which affects how much CPU the extended compression routines use. Higher compression levels use more CPU but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is **\*INITPARMS**.

**ECWINSIZE** specifies the extended compression window size, which is specifically for the history buffer that is filled from the user's input buffer ( both compressing and decompressing). The window specifies the amount of storage designated to maintain data previously read.

This data can be scanned for string matches. The extended compression window size affects how much virtual memory the extended compression routines use. Higher window size values use more memory but achieve greater compression. The valid values for this subparameter are 8–15, inclusive. The default value is \*INITPARMS.

**ECMLEVEL** specifies the extended compression memory level parameter, which determines how much memory should be allocated for other internal data structures like the hash table and the previous table (pointers to previous strings starting with the same 3 characters). The extended memory level affects how much memory the extended compression routines use. Higher memory levels use more virtual memory but achieve greater compression. The valid values for this subparameter are 1–9, inclusive. The default value is \*INITPARMS.

**CTLCHAR(\*NONE|\*FCFC|\*PRTCTL)**

specifies whether the data contains printer control characters. Valid values are:

\*NONE indicates that the data does not contain printer control characters.

\*FCFC indicates that the first character of each record contains an ANSI forms-control character. Any incorrect control characters are ignored, and single spacing is assumed. This parameter should be used when the source z/OS file is RECFM=xxA, which indicates it contains ANSI carriage control.

\*PRTCTL indicates that skip and space-before values are in the first four characters of every record. These values are useful in high-level language (HLL) programs. Refer to the CPYSPLF command in the *i5/OS CL Reference* for more information on these values.

**FDSNLEN(\*SPLF)**

specifies the record length of the local file. The default value \*SPLF means: look at the actual spool file and use its length in this parameter.

---

**Note:** For TDSN, the values of \*LIBL, \*CURLIB, QTEMP cannot be used when sending to or from an i5/OS system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the i5/OS.

---

**NOTIFY(userid)**

specifies the user ID on the local node to receive status messages that are directed to the message queue of the user.

**PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to run a task, for example TAPE.

---

**Note:** This parameter is for use only in Connect:Direct OS/390 environments.

---

**PNAME(name)**

identifies the Connect:Direct Process in any messages or statistics related to the operator. The name is from 1 to 12 characters long. The first character must be alphabetic. Use this name to identify the operation in any messages or statistics. The default name is CDSNDSPL.

---

**Note:** Only Connect:Direct for i5/OS and ProductNameB UNIX support Process names with more than 8 characters.

---

**PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local node. This parameter should only be used to validate security with a different ID than the one you used to sign on to Connect:Direct. PNODEID is not used when both the ID and password match on the local and remote nodes.

**id** specifies the security ID passed to the security system on the local node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. This parameter is used by the security system on the local node to validate the current security password (1–10 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password. It is used by the security system to change the current security password to the new security password (1–10 alphanumeric characters).

---

**Note:** Although you can specify a value for newpswd, Connect:Direct for i5/OS does not support this parameter.

---

**PRTY(n)**

specifies the priority of the operation. High numbers indicate high priorities; low numbers indicate low priorities. Use this priority only for selection within CLASS. It does not affect transmission priority. The range is from 0–15. The default is \*INITPARMS. Type **DSPPFM INITPARMS** to view the default.

**RMTWORK(\*YES|\*NO)**

specifies whether a remote Process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*YES** allows the remote Process to run on the session after Connect:Direct for i5/OS completes its Process.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function. \*NO is the default for this parameter.

**SACCT(snode-accounting-data)**

specifies the accounting data for the remote node. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, the string must be enclosed in single quotation marks. This parameter is valid for z/OS nodes only.

**SECOPTIONS(\* \*)**

specifies Encryption and Digital Signature values.

**Encryption** specifies if data encryption is to be performed and what algorithm to use.

**Digital Signature** specifies if a digital signature is created during file transmission.

---

**Note:** The SECOPTIONS parameter is valid only if Connect:Direct Secure+ Option is installed. For information about these parameters, refer to the *Connect:Direct Secure+ Option for i5/OS Implementation Guide*.

---

**SNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the remote node. Use this parameter only to validate security with a different ID than the one you used to sign on to Connect:Direct.

---

**Note:** Parameter values are case sensitive.

---

**id** specifies the security ID passed to the security system on the remote node (1–64 alphanumeric characters).

**pswd** specifies the current security password and is used by the security system on the SNODE to validate the current security password (1–64 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password and is used by the security system to change the current security password to the new security password (1–64 alphanumeric characters).

---

**Note:** newpswd is not supported if the remote node is an i5/OS node.

---

#### **TBUFND(n)**

specifies the number of I/O buffers VSAM uses for transmitting data between virtual and auxiliary storage. Valid values range from 1–510. The default is **2**. Increasing this number generally improves the I/O performance on the file but requires more memory. This parameter is valid for z/OS nodes only.

#### **TDBPARMS(dbid, dbsubid)**

indicates that the Connect:Direct z/OS SQL SERVER is to process the I/O request against the database identified by the subparameters of this keyword and defined by the SQL= parameter. The DBPARMS parameter is used with the optional Connect:Direct z/OS SQL feature. This parameter is valid for z/OS nodes only. Only DB2 is supported.

To determine the values to code for access to a given DB2 table or set of DB2 tables, consult the database administrator.

**dbid** specifies the 1–4 byte DB2 database ID.

**dbsubid** specifies the 1–4 byte DB2 database sub ID.

#### **TDCB(modeldcb, blksize, den, dsorg, keylen, limct, lrecl, optcd, recfm, rkp, trtch)**

overrides attributes of the source file. For existing source and destination files, DCB attributes are determined from the operating system unless specified. For a new destination file, the DCB attributes of the source file are used to allocate the destination file unless DCB information is provided in the operation.

**modeldcb** specifies a model data set control block (DSCB).

**blksize** specifies the block length in bytes. The minimum and maximum lengths are 18 bytes, and 32,760 bytes.

**den** specifies the magnetic tape mode setting. The following table shows values for the DEN parameter for 7- and 9-track tapes. When coded together, the DEN and TRTCH values are used to select a tape device for allocation by Connect:Direct z/OS.

DEN	7-Track Tape	9-Track Tape
1	556 bpi	--
2	800 bpi	800 bpi
3	--	1600 bpi
4	--	6250 bpi

**dsorg** specifies the file organization. Supported file organizations are BDAM, ISAM, PO, PS, and VSAM.

**keylen** specifies the length of the keys used in a file. Maximum length in bytes is 255.

**limct** specifies the blocks or tracks searched to find a free block or available space.

**lrecl** specifies the record length in bytes.

**optcd** establishes optional processing associated with a specific file. This optional processing only applies to this file and is not automatically applied to other files involved in a RECEIVE operation. Valid options are:

**W** performs write validity checks on direct access storage devices.

**Q** performs ASCII-to-EBCDIC conversion for input files and EBCDIC-to-ASCII conversion for output files. **Q** is the default and only used for AL-labeled tape files.

**Z** performs reduced error recovery for tape files.

**recfm** specifies the format of the records in the file. You can specify any valid record format, such as:

**F** specifies fixed format.

**FA** specifies fixed format with ASA printer control.

**FB** specifies fixed block format.

**FBA** specifies fixed block format with ANSI carriage control.

**FM** specifies fixed format with machine code control character.

**U** specifies undefined format.

**V** specifies variable format.

**VB** specifies variable block.

**VBA** specifies variable block format with ASA printer control.

**VBM** specifies variable block format with machine code control character.

**VS** specifies variable spanned format.

**VBS** specifies variable block spanned format.

For FDR volumes, you must specify RECFM=U.

**rkp** specifies the position of the first byte of the record key within each logical record. The beginning byte of a record is addressed as **0**.

**trtch** specifies the magnetic tape mode setting. When coded together, the TRTCH and DEN values are used to select a tape device for allocation by Connect:Direct z/OS. Valid options are:

**C** specifies data conversion, odd parity, and no translation.

**E** sets no data conversion, even parity, and no translation.

**T** specifies no data conversion, odd parity, and BCD or EBCDIC translation.

**ET** specifies no data conversion, even parity, and BCD or EBCDIC translation.

**COMP** is a feature for 3480X tape drives only. It enables Improved Data Recording Capability (IDRC), which compresses the data. This subparameter overrides the system-wide IDRC setting for no compression. If you are specifying COMP, you must also include a UNIT= parameter that specifies either 3480X or a systems-programmer-defined name equivalent to a 3480X tape drive.

**NOCOMP** overrides the system-wide IDRC setting for compression. NOCOMP applies to 3480X tape drives only.



**TDISP(current, normal end, abnormal end)**

specifies the status of the file on the remote node.

**current** specifies the status of the file before the operation executes. Valid choices for this subparameter are:

**NEW** specifies that the operation creates the destination member for an existing file. NEW is the default. The file is created if the member does not exist. The file cannot be created if the member already exists.

**OLD** specifies that the destination member already exists. The operation has exclusive control of the member.

**MOD** specifies that the operation modifies the member by adding data to the end of the member or, if none exists, allocates a new member. The file is also created if it does not exist.

**RPL** specifies that the destination member replaces any existing member or, if none exists, allocates a new member. The file is created if it does not exist.

**SHR** specifies that the destination member already exists. The file is read simultaneously by another job or operation.

---

**Note:** Normal and abnormal second disposition are not applicable when sending to Connect:Direct for i5/OS.

---

For information on parameters for other platforms, refer to the appropriate volume of the Connect:Direct Processes Web site at

<http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

**TIOEXIT(program-name)**

specifies the name of the user-written program to be given control for I/O-related requests for the associated data. This parameter is valid for z/OS nodes only.

**TIOPARM(parameter-name)**

establishes a parameter, or list of parameters, to be passed to the specified exit. The TIOPARM syntax is dependent on the parameter list required by the program specified in TIOEXIT. This parameter is valid for z/OS nodes only.

**TLABEL(seqno,labeltype,pswdread,inout,retpd,expdt)**

specifies label information for the tape. This parameter is valid for z/OS nodes only.

**seqno** specifies the relative file position on the tape.

**labeltype** is designated as:

**SL** specifies IBM standard labels.

**AL** specifies American National Standard labels.

**BLP** specifies bypass label processing.

**LTM** specifies bypass leading tape marks.

**NL** specifies no labels.

**pswdread** specifies that a password must be given by the operator or user before the file is accessed.

**inout** specifies that a BSAM data set opened for INOUT or a BDAM data set opened for UPDAT is to be read only.

**retpd** specifies the retention period for the data set in days, where *nnnn* is 1–4 digits.

**expdt** specifies the expiration date for the data set, where *yyddd* or *yyyyddd* is a valid Julian date.

**TLINK(vmid,pwd,accmode,cuu)**

specifies the disk where the CMS file is located. This parameter allows the user to access the CMS file. This parameter is valid for VM or VSE nodes only.

**vmid** specifies the owner ID for the CMS minidisk where the file is located. The valid length ranges from 1–8 characters.

**pwd** specifies the appropriate password for the CMS minidisk where the file is located. The maximum length is 256 characters. The default password is ALL.

**accmode** specifies the link access mode.

When used with the FROM parameter, the access modes are **W** (primary read/write access), **M** (primary multiple access), **R** (primary read only), **RR** (primary and secondary read only access), **WR** (primary read/write access; alternate read only access), **MR** (primary multiple access; alternate read only access), and **MW** (primary multiple access; alternate read/write only access).

When used with the TO parameter, the access modes are **W**, **M**, and **MW**.

**cuu** specifies the virtual address of the disk where the CMS file is located. Any three-digit number is valid.

**TMSVGP(n)**

specifies the remote file mass storage volume group that resides on a mass storage system (MSS) device, which must be a valid DD (data definition) name. This parameter is valid for z/OS nodes only.

**TPROTCT(\*YES|\*NO)**

specifies whether an IBM RACF profile is created for a new file on a remote VM system.

**TRACEOPT****Trace Instructions**

(L, S, F, C, X, K, D, H, V, W, P|\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)

- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

#### Command Trace Options

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

#### **TSPACE(avgbk, type, primary, secondary, pdsblk, rlse, contig, round)**

specifies the amount of storage to be allocated for new files on the destination node. If TSPACE is specified, the DISP of the destination file must be NEW. If TSPACE is not specified and the DISP is NEW, space allocation defaults to the value obtained from the source file (unless a TTYPE has been specified). The default allocation is in blocks, and if supplied with the ROUND option, provides device-independent space allocation. This parameter is valid only for z/OS nodes.

**avgbk** defines the average block size.

**type** defines the space allocation. Valid values are:

- ❖ **CYL** specifies that space is to be allocated by cylinder.
- ❖ **TRK** specifies that space is to be allocated by track.
- ❖ **blk** specifies that space is to be allocated by the average block length of the data and is the default. The system computes the number of tracks to be allocated. 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. If no space information is coded, allocation is in blocks, due to device dependencies.

**primary** specifies the primary allocation of storage (number of units).

**secondary** specifies the secondary allocation of storage (number of units).

**pdsblk** specifies the number of PDS directory blocks to be created in the file.

**rlse** sets the release of unused storage allocated to the output file. Valid values are \*YES and \*NO.

**contig** specifies the contiguous storage for the primary allocation. Valid values are \*YES and \*NO.

**round** specifies the rounding of storage allocated by average block length to an integral number of cylinders. Valid values are \*YES and \*NO.

**TSQL(sql-statement|DSN=data-set-name)**

defines the SQL statement to be processed against the database identified by the DBPARMS parameter or the DSN containing the SQL statements(s). The SQL parameter is used with the optional Connect:Direct for SQL feature. This parameter is valid for z/OS nodes only.

**sql-statement** defines the actual SQL statement to be processed.

**data-set-name** specifies the data set containing the SQL statement or statements to be processed.

**TTYTYPE(name)**

specifies the entry name of the type defaults file on the remote node that contains the default file attributes used to allocate the destination file. TTYTYPE is required only if sending to a MS-DOS node.

**TUNIT(unit-name)**

specifies the unit address, device type, or user-assigned group name where the file resides or will reside. For SAM-to-SAM copies where the destination file is new and the UNIT parameter is not coded, the device type from the source file is used. For VSE nodes, you must specify either this parameter or TVOL.

Specify a unit-count to allow additional units to be allocated if required, or specify **P** to allocate the same number of units as volumes and then parallel mount the volumes.

**TVOL(\*YES|\*NO)**

specifies the volume serial number(s) containing the file and optional processing associated with the file. If TVOL is not specified, the file must be cataloged. For VSE nodes, you must specify either this parameter or TUNIT.

**private** specifies allocation of an output file only if the volume is specifically requested and is used for direct access storage devices only. Valid values are \*YES and \*NO.

**volcount** specifies the maximum number of volumes required by an output file.

**ser** identifies by serial number the volumes where the output file resides or will reside.

**TVOLREF(data-set-name)**

allows you to place a data set on the same volume as the referenced data set. It must be cataloged on the system where it is referenced. This parameter is valid for z/OS nodes only.

**TVSAMCAT(dsn,vmid,pwd,accmode,cuu)**

specifies the catalog for the VSAM file to be copied. This parameter is required only if using a catalog other than the master catalog. This parameter is valid for VSE nodes only.

**dsn** specifies the filename of the VSAM catalog containing the file to be copied. The maximum length is 44 characters.

**vmid** specifies the owner ID for the VSAM minidisk where the file is located. The maximum length is eight characters.

**pwd** specifies the appropriate password for the VSAM minidisk where the file is located. The maximum length is eight characters.

**accmode** specifies the link access mode. Valid access modes are NULL, **W** (primary read/write access), **M** (primary multiple access), and **MW** (primary multiple access; alternate read/write only access).

**cuu** specifies the virtual address of the disk where the VSAM file is located. Any three-digit number is valid.

## Sending a Spooled File to a Remote Node

Use the **Connect:Direct Send Spool (CDSNDSPL)** screens to send a spooled file to a remote node. Complete the following steps to send a spooled file to a remote node:

1. Access the **CDSNDSPL** screens by typing **CDSNDSPL** at the command line and pressing **F4** or by choosing the send spooled file option from the Connect:Direct main menu.
2. Type the remote node name and press **Enter**. The command retrieves the remote node environment and display fields for required parameters.
3. Type values for required and any optional parameters as appropriate and press **Enter**.

The following is a sample of the **Connect:Direct Send Spool (CDSNDSPL)** screen.

```

                                Connect:Direct Send Spool File (CDSNDSPL)

Type choices, press Enter.

Receiving Node Name . . . . . _____
Receiving Node Environment . . . _____ AS400, MSP, MVS, OS/400...
Spooled file . . . . . _____ Name
Job name:
    User Id . . . . . _____ Name, *
    Number . . . . . _____ Name
    Spooled file number . . . . . _____ 000000-999999
    Receiving File Name . . . . . _____ 1-9999, *ONLY, *LAST
    _____

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
    
```



---

# Running a Remote Task

This chapter describes the Connect:Direct Run Task (CDRUNTASK) command and its function. This chapter also shows sample **CDRUNTASK** screens and describes the required and optional CL parameters for the CDRUNTASK command.

---

## Using the CDRUNTASK Command

Use the CDRUNTASK command to run a task at a remote node. This command allows user programs, or subtasks, to be attached and run as a subtask under the remote Connect:Direct during command execution.

For additional information, refer to the appropriate RUN TASK section of the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html> for information about the remote node.

---

**Note:** RUN JOB and RUN TASK Process statements use the authenticated user's JOBD for the library list and not the CDJOB library list, which is the CDADMIN job description used to start Connect:Direct. RUN JOB and RUN TASK Process statements use the CDJOBQ job queue.

---

## CDRUNTASK Command Example

The command shown in the following figure is an example of the CDRUNTASK command.

```
CDRUNTASK      PNAME(CDTASK) SNODE(DALLAS.OS3901)
               SNODENVIRN(OS390) PGM(ACCTUPDT)
               PARM('CL44'DALLAS.ACCTDATA' 'CL44'
                   'DALLAS.REPORT''')
```

## CDRUNTASK Example With New TraceOpt Parameters

```

==> CDRUNTASK SNODE(CDDV3600) SNODENVIRN(AS400)
CMD('WRKACTJOB SBS(CDDV3600) OUTPUT(*PRINT)') TRACEOPT(*ALL *ALL)
SNODEID(RMTUSER ())

The associated Statistics record:

                                Display Spooled File
File . . . . . :   CDSELSTAT                                Page/Line   1/6
Control . . . . .                               Columns    1 - 130
Find . . . . .
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
C:D i5          Select Statistics (CDSELSTAT) Detail 03/16/04 11:03:12 Page 0001

-----
Event:           Communications Session Started (SMSES)
Process Name:    CDTASK                                     Process Number: 00014891
Date/Time:       03/16/04   10:57:12
Local Node:      CDDV3600                                  Local User:    LCLUSER
Remote Node:     CDDV3600
Protocol:        TCP/IP                                    Type:          INIT
Message Id:      ASMT292I

-----
Event:           Process Started (SMPST)
Process Name:    CDTASK                                     Process Number: 00014891
Date/Time:       03/16/04   10:57:12
Local Node:      CDDV3600                                  Local User:    RMTUSER
Remote Node:     CDDV3600
Submit User:     CDDV3600                                  Pnode:        N

-----
Event:           Process Started (SMPST)
Process Name:    CDTASK                                     Process Number: 00014891
Date/Time:       03/16/04   10:57:13
Local Node:      CDDV3600                                  Local User:    LCLUSER
Remote Node:     CDDV3600
Submit User:     IDEVUSER                                   Pnode:        Y

```

## CDRUNTASK Command Format

The following are the parameters for the CDRUNTASK command. Required parameters are in bold.

Command	Parameters
CDRUNTASK	<b>SNODE(remote-node-name)</b>
	<b>SNODENVIRN(remote-environment-name)</b>
	† <b>PGM(program-name)</b>
	‡ <b>CMD(command)</b>
	CLASS(n)
	CRC(* <b>INITPARMS</b>  *NETMAP *OFF *ON)
	NOTIFY(userid)
	‡‡ <b>PARM(parameters)</b>

† For tasks on z/OS, Windows, HP NonStop, VM, and VSE  
‡ For tasks on i5/OS, Windows, OpenVMS, UNIX, Stratus VOS  
‡‡ For tasks on z/OS, HP NonStop, VM, and VSE



Command	Parameters
	PLEXCLASS(user-defined class)
	PNAME(name)
	PNODEID(id, pswd, newpswd)
	RMTWORK(*YES *NO)
	SACCT(snode-accounting-data)
	SNODEID(id, pswd, newpswd)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS *CURRENT Command Trace Options: L,S,F,C,X,K,D,H,V,P *ALL *EXT *COMMON  *NONE)
†	For tasks on z/OS, Windows, HP NonStop, VM, and VSE
‡	For tasks on i5/OS, Windows, OpenVMS, UNIX, Stratus VOS
†‡	For tasks on z/OS, HP NonStop, VM, and VSE
<b>Note:</b> The CLASS parameter is not used by Connect:Direct for i5/OS.	

## Required Parameters

The following two parameters are required when you use the CDRUNTASK command.

### **SNODE(remote-node-name)**

specifies a 1–16 character alphanumeric name for the remote node.

**Note:** This value is case sensitive.

### **SNODENVIRN(remote-environment-name)**

specifies the remote environment to execute the task.

One of the next two parameters is required for the CDRUNTASK command. Each parameter lists the environments for which they apply.

### **PGM(program-name)**

specifies the RUN TASK program name on the remote node. The following environments can use this parameter: OS/390, Windows, Tandem, VM, and VSE

### **CMD(command)**

specifies batch-capable i5/OS CL commands and parameters for a remote node. The maximum number of permitted characters is 256. The following environments can use this parameter: i5/OS, MS-DOS, Windows, OpenVMS, UNIX, and Stratus VOS.

Refer to the Connect:Direct Processes Web site at

<http://www.sterlingcommerce.com/documentation/processes/processhome.html> for parameter values for the remote node.

## Optional Parameters

The following parameters are optional. Unless otherwise noted, the parameters apply to all remote node environments.

### **CLASS(n)**

determines the node-to-node session on which a Connect:Direct operation can execute. The CLASS is a number from 1–255. If you do not specify the CLASS parameter, it defaults to the maximum number of sessions defined in the mode description for the communications environment.

---

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

---

### **CRC(\*INITPARMS|\*NETMAP|\*OFF|\*ON)**

Determines whether node and Process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to \*OFF, setting overrides for CRC checking will be ignored. The default value is \*INITPARMS.

\*INITPARMS specifies that the CRC setting is determined by the value for the initialization parameter TCP.CRC.

\*NETMAP specifies that the CRC setting is determined by the value for the NETMAP parameter, Cyclic Redundancy Check. The value for the initialization parameter TCP.CRC must be set to \*ON and the value for the initialization parameter TCP.CRC.OVERRIDE must be set to \*YES.

\*OFF specifies that CRC is disabled.

\*ON specifies that CRC is enabled.

### **NOTIFY(userid)**

specifies the user ID on the local node to receive status messages that are directed to the message queue of the user.

### **PARM(parameters)**

specifies the remote parameters needed for the program specified in the PGM parameter. This parameter is valid for tasks on z/OS, HP NonStop, VM, and VSE.

### **PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to run a task.

---

**Note:** This parameter is for use only in Connect:Direct OS/390 environments.

---

### **PNAME(name)**

identifies the Connect:Direct Process in any messages or statistics. The name is from 1 to 12 characters long. The first character must be alphabetic. The default name is CDSND.

---

**Note:** Only Connect:Direct for i5/OS and Connect:Direct UNIX support Process names with more than 8 characters.

---

### **PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local node. Only use this parameter to validate security with a different ID than the one you used to sign on to Connect:Direct.

**id** specifies the security ID passed to the security system at the local node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. The security system at the local node can use this parameter to validate the current security password (1–10 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password. The security system can use this parameter to change the current security password to the new security password (1–10 alphanumeric characters).

---

**Note:** Although you can specify a value for the new password, Connect:Direct for i5/OS does not currently support this parameter.

---

#### **RMTWORK(\*YES|\*NO)**

specifies whether a remote Process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*YES** allows a remote Process to run on the session after the Connect:Direct for i5/OS completes its Process.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function. **\*NO** is the default.

#### **SACCT(snode-accounting-data)**

specifies the accounting data for the remote node. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, the string must be enclosed in single quotation marks.

#### **SNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the remote node. This parameter should only be used to validate security with a different ID than the one you used to sign on to Connect:Direct.

**id** specifies the security ID passed to the security system on the remote node (1–64 alphanumeric characters).

**pswd** specifies the current security password and is used by the security system on the SNODE to validate the current security password (1–64 alphanumeric characters). This parameter is optional unless the user has security set to require a password.

**newpswd** specifies the new security password and is used by the security system to change the current security password to the new security password (1–64 alphanumeric characters).

---

**Note:** newpswd is not valid if the remote node is a Connect:Direct for i5/OS node. Parameter values are case sensitive.

---

#### **TRACEOPT**

Trace Instructions

(L, S, F, C, X, K, D, H, V, W, P, |\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.

- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)
- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

#### Command Trace Options

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

---

## Using the CDRUNTASK Screens

Use the **Connect:Direct Run Task (CDRUNTASK)** screen to initiate a task on the remote node.

## Initiating a Task on a Remote Node

Complete the following steps to initiate a task on a remote node:

1. Access the CDRUNTASK screen by typing **CDRUNTASK** at the command line and pressing **F4** or by selecting the Run Task option from the main menu.
2. Type the remote node name at the command prompt and press **Enter**. The command retrieves the remote node environment and display fields for required parameters.
3. Type values for required and optional parameters as appropriate, and press **Enter**.

The following figure shows a sample of the first screen in the **Connect:Direct Run Task (CDRUNTASK)** screen flow for a Connect:Direct for i5/OS remote environment.

```

Connect:Direct Run Task (CDRUNTASK)

Type choices, press Enter.

Remote Node Name . . . . . > DALLAS_____
Remote Environment . . . . . > OS/400_____ AS400, MVS, OS390, OS/400...
Remote Task Command . . . . . _____

-----
CRC . . . . . *INITPARMS *INITPARMS, *NETMAP, *ON, *OFF
Trace Options:
  Trace Instructions . . . . . *CURRENT_____
  Command trace options . . . . . *NONE_____
  Process Name . . . . . CDTASK_____ Name
Local Node User Info.:
  User Id . . . . . *CURRENT_____ Character value, *CURRENT
  Password . . . . . _____ Character value
  New Password . . . . . _____ Character value
  User to Notify . . . . . *CURRENT_____ Name, *CURRENT, *NONE
                                                    More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Additional screens vary according to the remote node environment.



---

# Submitting a Connect:Direct Process on a Remote Node

This chapter describes the Connect:Direct Submit (CDSUBMIT) command and its function. This chapter also shows sample **CDSUBMIT** screens and describes the CL parameters used for the CDSUBMIT command.

---

## Using the CDSUBMIT Command

Use the CDSUBMIT command to submit a Connect:Direct Process on a remote node. This command allows you to perform the following tasks:

- ❖ Initiate a session with a remote node
- ❖ Submit a Process to the remote Connect:Direct
- ❖ Override certain parameters of the submitted Process

The remote node must support the Connect:Direct Process language.

---

**Note:** The Process to be submitted must exist on the remote system.

---

## CDSUBMIT Command Example

The following example presents a sample of the CDSUBMIT command.

```
CDSUBMIT SNODE(OS390.V20100) SNODENVIRN(OS390) PROCFILE(file1.tst) PNAME(CDSUBOS390)
```

## CDSUBMIT Command Format

The following are the parameters for the CDSUBMIT command. Required parameters are in bold.

Command	Parameters
CDSUBMIT	<b>SNODE(remote-node-name)</b>
	<b>SNODENVIRN(remote-environment-name)</b>
	<b>PROCFILE(file-name)</b>

† These parameters override the associated parameters of the remote Connect:Direct Process.

Command	Parameters
	CLASS(n  <u>MODED</u> )
	CRC( <u>INITPARMS</u>  *NETMAP *OFF *ON)
	†NEWNAME(file-name  <u>NONE</u> )
	NOTIFY( <u>CURRENT</u>  *NONE userid)
	†PCLASS(n)
	†PHOLD(*YES *NO *CALL)
	PLEXCLASS(user-defined class)
	PNAME(process-name)
	PNODEID(id,pswd,newpswd)
	†PNOTIFY(userid)
	†PPACCT(PNODE accounting-data)
	†PPNODEID(id,pswd,newpswd)
	†PPRTY(n)
	†PREQUEUE(*YES *NO)
	†PRETAIN(*YES *NO *INITIAL)
	†PSACCT(SNODE accounting-data)
	†PSNODE(secondary-node-name)
	†PSNODEID(id,pswd,newpswd)
	†PSTARTT(day date,time)
	†PSYMBOLS(&symbolic n = variable n)
	RMTWORK(*YES *NO)
	SACCT(SNODE accounting-data)
	SNODEID(id,pswd,newpswd)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON  <u>INITPARMS</u>   <u>CURRENT</u> Command Trace Options: L,S,F,C,X,K,D,H,V,W,P *ALL *EXT *COMMON  * <u>NONE</u> )

† These parameters override the associated parameters of the remote Connect:Direct Process.

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

## Required Parameters

The following parameters are required. Unless otherwise noted, the parameters apply to all remote node environments.



**SNODE(remote-node-name)**

specifies a 1–16 character alphanumeric name for the remote node.

**SNODENVIRN(remote-environment-name)**

specifies the remote node environment.

**PROCFILE(file-name)**

specifies the name of the file or member that contains the Connect:Direct Process to be submitted.

This file must already exist on the node where the Connect:Direct Process is to be submitted.

For information about parameters for the remote node, refer to the Connect:Direct Processes Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>.

## Optional Parameters

The following parameters are optional. Unless otherwise noted, the parameters apply to all remote node environments.

**CLASS(n|\*MODED)**

determines the node-to-node session on which an operation can execute. The CLASS is a number from 1–255. If you do not specify the CLASS parameter, it defaults to the number of maximum sessions defined in the mode description for the communications environment.

---

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

---

**\*MODED** uses the class value specified for the destination node.

**CRC(\*INITPARMS|\*NETMAP|\*OFF|\*ON)**

Determines whether node and process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to \*OFF, setting overrides for CRC checking will be ignored. The default value is \*INITPARMS.

**\*INITPARMS** specifies that the CRC setting is determined by the value for the initialization parameter TCP.CRC.

**\*NETMAP** specifies that the CRC setting is determined by the value for the NETMAP parameter, Cyclic Redundancy Check. The value for the initialization parameter TCP.CRC must be set to \*ON and the value for the initialization parameter TCP.CRC.OVERRIDE must be set to \*YES.

**\*OFF** specifies that CRC is disabled.

**\*ON** specifies that CRC is enabled.

**NEWNAME(file-name|\*NONE)**

specifies a new name to be given to the submitted Process on the remote system. This name overrides the name in the Process file.

**file-name** specifies the name for the submitted Process on the remote system.

**\*NONE** uses the name specified in the Process.

**NOTIFY(\*CURRENT|\*NONE|userid)**

specifies the user ID on the receiving node to receive status messages that are directed to the message queue of the user.

**\*CURRENT** uses the ID of the current user.

**\*NONE** indicates that no user receives the messages.

**userid** specifies the name of the user to receive the status messages.

**PCLASS(n)**

specifies the Connect:Direct Process CLASS for the submitted remote Process. This entry overrides the class specified in the Process.

**PHOLD(\*YES|\*NO|\*CALL)**

specifies whether the submitted remote Process is placed in the hold queue at submission. The default is the HOLD value of the Process.

**\*YES** specifies that the Process submits to the hold queue.

**\*NO** specifies that the Process executes as soon as possible.

**\*CALL** specifies that the remote Process is placed in the hold queue until a session is established with its SNODE.

**PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to submit a Connect:Direct Process.

---

**Note:** This parameter is for use only in Connect:Direct OS/390 environments.

---

**PNAME(process-name)**

identifies the Process in any messages or statistics. The name is from 1 to 12 characters long. The first character must be alphabetic. The default name is CDSND.

---

**Note:** Only Connect:Direct for i5/OS and Connect:Direct UNIX support Process names with more than 8 characters.

---

**PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local node. Use this parameter only to validate security with a different ID than the one you used to sign on to Connect:Direct. Do not use PNODEID when both the ID and password match on the local and remote nodes.

**id** specifies the security ID passed to the security system at the local node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. The security system at the local node can use this parameter to validate the current security password (1–10 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password.

---

**Note:** Although you can specify a value for newpswd, Connect:Direct for i5/OS does not support this parameter.

---

**PNOTIFY(userid)**

specifies the user who is to receive remote Process completion messages. This entry overrides the one specified in the Process.

**PPACCT(PNODE accounting-data)**

specifies the accounting data for the PNODE in the submitted remote Process. This entry overrides the one specified in the Process.

**PPNODEID(id,pswd,newpswd)**

specifies the primary node user ID and password for the submitted remote Process. This entry overrides the one specified in the Process.

---

**Note:** Parameter values are case sensitive.

---

**PPRTY(n)**

specifies the Connect:Direct priority class for the submitted remote Process. This entry overrides the one specified in the Process. The valid range is 0–15.

**PREQUEUE(\*YES|\*NO)**

specifies whether a copy step should be requeued with HE status after an abnormal termination. This parameter is only valid when checkpointing.

**PRETAIN(\*YES|\*NO|\*INITIAL)**

tells Connect:Direct for i5/OS to keep a copy of the remote Process in the hold queue of the remote node after the Process executes.

**\*INITIAL** specifies that the Process executes each time the remote Connect:Direct is initialized.

**PSACCT(SNODE accounting-data)**

specifies the accounting data for the remote node in the remote submitted Process. This entry overrides the one specified in the Process.

**PSNODE(secondary node-name)**

specifies the secondary node for the submitted remote Process. This entry overrides the one specified in the submitted remote Process.

**PSNODEID(id,pswd,newpswd)**

specifies the secondary node user ID and password for the submitted remote Process. This entry overrides the one specified in the Process.

---

**Note:** Parameter values are case sensitive.

---

**PSTARTT(day|date,time)**

specifies the day or date and time the submitted remote Process is to execute. The date format must be specified as the format expected by the remote node. The time field is 10 characters in length and the date field is 9 characters in length.

**PSYMBOLS(&symbolic n = variable n)**

specifies values to be used for symbolic substitution in the submitted remote Process.

**RMTWORK(\*YES|\*NO)**

specifies whether a remote Process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*YES** allows a remote Process to run on the session after the Connect:Direct for i5/OS completes its Process.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function.

**SACCT(SNODE accounting-data)**

specifies the accounting data for the remote node. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, the string must be enclosed in single quotation marks.

**SNODEID(id,pswd,newpswd)**

specifies security user IDs and passwords at the remote node. This parameter should only be used to validate security with a different user ID than the one you used to sign on to Connect:Direct.

**id** specifies the security user ID passed to the security system on the remote node (1–64 alphanumeric characters).

**pswd** specifies the current security password on the remote node and is used by the security system on the remote node to validate the current security password (1–64 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password and is used by the security system to change the current security password to the new security password (1–64 alphanumeric characters).

---

**Note:** Connect:Direct for i5/OS does not support newpswd if the remote node is an i5/OS node. Parameter values are case sensitive.

---

**TRACEOPT****Trace Instructions**

(L, S, F, C, X, K, D, H, V, W, P, |\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)
- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

**Command Trace Options**

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.

- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

---

## Using the CDSUBMIT Screens

Use the **CDSUBMIT** screens to submit a Connect:Direct Process on a remote node.

### Submitting a Process on a Remote Node

Complete the following steps to submit a Process on a remote node:

1. Access the **CDSUBMIT** screens by typing **CDSUBMIT** at the command line and pressing **F4** or by selecting the **SUBMIT** option from the main menu.
2. Type the remote node name at the command prompt and press **Enter**. The command retrieves the remote node environment and display fields for required parameters.
3. Type values for required and optional parameters as appropriate and press **Enter**. Additional screens relative to the remote environment are displayed.

## Connect:Direct Submit Primary Screen

The following is a sample of the first screen in the **Connect:Direct Submit (CDSUBMIT)** screen flow.

```

Connect:Direct Submit (CDSUBMIT)

Type choices, press Enter.

Remote Node Name . . . . . > MVS.LEVEL1_____
Remote Environment . . . . . > OS/390_____ MVS, OS390, MSP, VSE, VM...
Remote process file name . . . . . 'KMOOR1.AS400.CNT($BOTH)'_____

-----
CRC . . . . . *INITPARMS *INITPARMS, *NETMAP, *ON, *OFF
Trace Options:
  Trace Instructions . . . . . *CURRENT_____
  Command trace options . . . . . *NONE_____
Process Name . . . . . CDSUBMIT_____ Name
Local Node User Info.:
  User Id . . . . . *CURRENT_____ Character value, *CURRENT
  Password . . . . . _____ Character value
  New Password . . . . . _____ Character value
  User to Notify . . . . . *CURRENT_____ Name, *CURRENT, *NONE

More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

---

# Submitting a Job on a Remote Node

This chapter describes the Connect:Direct Run Job (CDRUNJOB) command and its function. This chapter also shows sample **CDRUNJOB** screens and describes the required and optional CL parameters for the CDRUNJOB command.

---

## Using the CDRUNJOB Command

Use the CDRUNJOB command to submit a job on the remote node. The job can be:

- ❖ A CL command that is submitted on a remote i5/OS node.
- ❖ JCL that is submitted by the remote Connect:Direct for z/OS, VM, or VSE to the z/OS, VM, or VSE system.
- ❖ A UNIX command or program that is submitted by Connect:Direct Windows, or UNIX to the Windows, or UNIX node.
- ❖ An OpenVMS command or program to submit a job on the OpenVMS node.

The command, JCL, or program must exist on the remote system before you submit it with CDRUNJOB.

---

**Note:** RUN JOB and RUN TASK Process statements use the authenticated user's JOBD for the library list and not the CDJOB library list, which is the CDADMIN job description used to start Connect:Direct. RUN JOB and RUN TASK Process statements use the CDJOBQ job queue.

---

## CDRUNJOB Command Example

The command shown in the following figure is an example of the CDRUNJOB command.

```
CDRUNJOB      SNODE(DALLAS.OS3901) SNODENVIRN(OS390)
              JOBFILE('USER1.AS400.CNTL(IEFBR14)')
              SNODEID(USERID PASSWORD)
```

For additional examples, refer to the Connect:Direct Process Web site at <http://www.sterlingcommerce.com/documentation/processes/processhome.html>

## CDRUNJOB Command Format

The following are the parameters for the CDRUNJOB command. Required parameters are in bold.

Command	Parameters
CDRUNJOB	<b>SNODE(node-name)</b>
	<b>SNODENVIRN(environment-name)</b>
	BATCHID(idname)
	CLASS(n)
	CRC(* <b>INITPARMS</b>  I*NETMAP N*OFF F*ON O)
	†JOBFILE(filename)
	LINK(CMS-file-information)
	NOTIFY(userid)
	PLEXCLASS(user-defined class)
	PNAME(name)
	PNODEID(userid,pswd,newpswd)
	RMTWORK(*YES * <b>NO</b> )
	SACCT(SNODE accounting-data)
	SNODEID(userid,pswd,newpswd)
	‡TSYSOPTS(remote-job-command)
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS * <b>CURRENT</b> Command Trace Options: L,S,F,C,X,K,D,H,V,W,P *ALL *EXT *COMMON  * <b>NONE</b> )
	† Valid for z/OS, OpenVMS, VM, VSE, and Windows only
‡ Valid for i5/OS, MS-DOS, OpenVMS, UNIX, Stratus VOS, and Windows	

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

## Required Parameters

The following parameters are required. Unless otherwise noted, the parameters apply to all remote node environments.

### **SNODE(node-name)**

specifies a 1–16 character alphanumeric name for the remote node.

**Note:** This value is case sensitive.

### **SNODENVIRN(environment-name)**

specifies the remote environment to execute the job.



## Optional Parameters

The following parameters, listed alphabetically, are optional. Unless otherwise noted, the parameters apply to all remote node environments.

### **BATCHID(idname)**

specifies the VM remote Virtual Machine ID name. The valid range is 1–8 characters.

### **CLASS(n)**

determines the node-to-node session where a Connect:Direct operation can execute. The CLASS is a number from 1–255. If you do not specify the CLASS parameter, it defaults to the maximum number of sessions defined in the mode description for the communications environment.

---

**Note:** The CLASS parameter is not used by Connect:Direct for i5/OS.

---

### **CRC(\*INITPARMS|\*NETMAP|N\*OFF|F\*ON|O)**

Determines whether node and process statement overrides for Cyclic Redundancy Check (CRC) are allowed. If this value is set to \*OFF, setting overrides for CRC checking will be ignored. The default value is \*INITPARMS.

\*INITPARMS (or \*I) specifies that the CRC setting is determined by the value for the initialization parameter TCP.CRC.

\*NETMAP (or \*N) specifies that the CRC setting is determined by the value for the NETMAP parameter, Cyclic Redundancy Check. The value for the initialization parameter TCP.CRC must be set to \*ON and the value for the initialization parameter TCP.CRC.OVERRIDE must be set to \*YES.

\*OFF (or \*F) specifies that CRC is disabled.

\*ON (or \*O) specifies that CRC is enabled.

### **JOBFILE(filename)**

specifies the name of the file that contains the job to be submitted. This file must already exist on the node where the job is to be submitted.

### **LINK(CMS-file-information)**

specifies where the CMS file is located. This parameter allows you to access the VM CMS file.

**owner id** specifies the user ID for the owner of the CMS minidisk where the file is located.

**owner password** specifies the password for the CMS minidisk where the file is located. This parameter accepts 1–8 characters. The default is ALL.

**LINK access mode** specifies the LINK access mode. The following entries are valid:

- ❖ W for primary read/write
- ❖ M for primary multiple access
- ❖ R for primary read only
- ❖ RR for primary and secondary read only
- ❖ WR for primary read/write; alternate read only
- ❖ MW for primary multiple access; alternate read/write

**virtual address** specifies the virtual address of the disk where the CMS file is located. A three-digit number is valid.

**NOTIFY(userid)**

specifies the user ID on the local node to receive status messages that are directed to the message queue of the user.

**PLEXCLASS(user-defined class)**

specifies a user-defined class containing up to eight characters associated with a Connect:Direct/Server in a Connect:Direct/Plex environment. This represents a type of server on which you want to submit a job.

---

**Note:** This parameter is for use only in Connect:Direct OS/390 environments.

---

**PNAME(name)**

identifies the Process in any messages or statistics related to the operation. The name is from 1–12 characters long. The first character must be alphabetic. The default name is CDSND.

---

**Note:** Only Connect:Direct for i5/OS and Connect:Direct UNIX support Process names with more than 8 characters.

---

**PNODEID(id, pswd, newpswd)**

specifies security user IDs and passwords at the local node. You should only use this parameter to validate security with a different ID than the one you used to sign on to Connect:Direct.

**id** specifies the security ID passed to the security system at the local node (1–10 alphanumeric characters).

**pswd** specifies the current security password for the specified ID. The security system at the local node can use this parameter to validate the current security password (1–10 alphanumeric characters). This is optional unless you have security set to require a password.

**newpswd** specifies the new security password. The security system can use this parameter to change the current security password to the new security password (1–10 alphanumeric characters).

---

**Note:** Although you can specify a value for the new password, Connect:Direct for i5/OS does not support this parameter.

---

**RMTWORK(\*YES|\*NO)**

specifies whether a remote Process can run on the session after Connect:Direct for i5/OS has completed its Process on the session.

**\*YES** allows a remote Process to run on the session after Connect:Direct for i5/OS completes its Process.

**\*NO** terminates the session after Connect:Direct for i5/OS executes the command function.

**SACCT(SNODE accounting-data)**

specifies the accounting data for the remote node. The maximum length of the accounting data is 256 characters. If special characters are part of the accounting data, the string must be enclosed in single quotation marks.

**SNODEID(userid, pswd, newpswd)**

specifies security user IDs and passwords at the remote node. You should only use this parameter to validate security with a different ID than the one you used to sign on to Connect:Direct.

**userid** specifies the security ID passed to the security system on the remote node (1–64 alphanumeric characters).

**pswd** specifies the current security password and is used by the security system on the SNODE to validate the current security password (1–64 alphanumeric characters). This is optional unless the user has security set to require a password.

**newpswd** specifies the new security password and is used by the security system to change the current security password to the new security password (1–64 alphanumeric characters).

---

**Note:** Parameter values are case sensitive.

Newpswd is not valid if the remote node is a Connect:Direct for i5/OS node.

---

## TRACEOPT

### Trace Instructions

(L, S, F, C, X, K, D, H, V, W, P, |\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT)

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)
- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

### Command Trace Options

(L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.

- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

**TSYSOPTS(remote-job-command)**

specifies system-specific submission information. For a remote i5/OS system, this parameter specifies a string 'cmd(any valid batch CL command) [any valid parameters for the i5/OS SBMJOB command]'. This CL command is submitted on the remote i5/OS system through a SBMJOB command. This parameter is valid for i5/OS, OpenVMS, UNIX, Stratus VOS, and Windows.

See the Connect:Direct Processes Web site at

<http://www.sterlingcommerce.com/documentation/processes/processhome.html> for valid platform-specific parameters.

## Using the CDRUNJOB Screens

Use the **Connect:Direct Run Job (CDRUNJOB)** screens to submit a job to a remote node.

### Submitting a Job to a Remote Node

Complete the following steps to submit a job to a remote node:

1. Access the **CDRUNJOB** screens by typing **CDRUNJOB** at the command line and pressing **F4** or by selecting the RUNJOB option from the main menu.
2. Type the remote node name at the command prompt and press **Enter**. The command retrieves the remote node environment and display fields for required parameters.
3. Type values for required and optional parameters as appropriate and press **Enter**. Additional screens displays as required by the remote node type.

The primary **CDRUNJOB** screen looks similar to the following.

```

                                Connect:Direct Run Job (CDRUNJOB)

Type choices, press Enter.

Remote Node Name . . . . . > hotpepper
Remote Environment . . . . . > AS400      AS400, MVS, OS390, OS/400...
Remote Job command . . . . . _____

_____

CRC . . . . . *INITPARMS      *INITPARMS, *NETMAP, *ON, *OFF
Trace Options:
  Trace Instructions . . . . . *CURRENT
  Command trace options . . . . . *NONE
  Process Name . . . . . CDJOB      Name
Local Node User Info.:
  User Id . . . . . *CURRENT      Character value, *CURRENT
  Password . . . . . _____      Character value
  New Password . . . . . _____      Character value
  User to Notify . . . . . *CURRENT      Name, *CURRENT, *NONE
                                          More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```



---

# Compressing Files

This chapter describes the Connect:Direct Compress (CDCOMP) command and its function. This chapter also shows sample **CDCOMP** screens and describes the parameters used with the CDCOMP command.

---

## Using the CDCOMP Command

Use the CDCOMP command to compress a file on the i5/OS node before sending it to a remote node.

---

**Note:** When you send a precompressed file to a remote node, specify the PRECMPR(\*YES) in the FMSYSOPTS of the CDSND command to have the remote node decompress the file. See Chapter 3, *Sending a File*, for information on using the PRECMPR parameter with the Send operation. Tuning parameters are not supported with the CDCOMP command. If Secure+ Option is installed, PRECMPR(\*YES) or DECMR(\*NO) options cannot be specified.

---

## Required CDCOMP Tasks

You must *always* complete one of the following activities when you use the CDCOMP command:

- ❖ Compress members into a newly created file. You can create the file manually, or you can allow CDCOMP to create the file.
- ❖ Ensure that the output file contains no members before you execute CDCOMP.
- ❖ Ensure that the output file is a physical data file with a record length of at least 256 bytes.

## CDCOMP Command Example

The following command is an example of how to code the CDCOMP command.

```
CDCOMP INFILE(TESTLIB/FILE1) OUTFILE(TESTLIB/CMP_FILE)
      REPLACE(*YES) SELMBR((TEST1 TEST2)) EXCMR((M1 M2))
```

## CDCOMP Command Format

The following table shows the format for the CDCOMP command.

Command	Parameters
CDCOMP	<b>INFILE(library-name/file-name)</b>
	<b>OUTFILE(library-name/file-name)</b>
	CKPTINV(n)
	EXCMBR(start-mbr end-mbr)
	REPLACE(*NO *YES)
	SELMBR(start-mbr end-mbr new-mbr)

## Required Parameters

The CDCOMP command has the following required parameters.

### **INFILE (library-name/file-name)**

specifies the name and library of the file to be compressed. The input file is a database or a save file.

**library-name** specifies the name of the library to be used.

\***LIBL** specifies that the library list is used to locate the input file.

\***CURLIB** specifies that the current library for the job is used to locate the input file.

**file-name** specifies the name of the file to be compressed.

### **OUTFILE (library-name/file-name)**

specifies the name and library of the database file to receive the compressed data. If the output file does not exist, it is created. The default value for library is \*CURLIB.

**library-name** specifies the name of the library to be used.

\***CURLIB** specifies that the current library for the job is used to locate the output file.

**file-name** specifies the name of the file to receive the compressed data.

## Optional Parameters

The optional parameters, listed alphabetically, are as follows. Unless otherwise noted, the parameters apply to all remote node environments.

### **CKPTINV(n)**

specifies the number of bytes, from 10K to 2 gigabytes, to send before taking a checkpoint. The format is *nnnnnnn*, *nnnnnnK*, or *nnnnnnM*, where **K** specifies thousands of bytes and **M** specifies millions of bytes. A value of **0** specifies no checkpoint/restart. Connect:Direct for i5/OS uses the value you specify, rounded up to the nearest record boundary, to determine when to take a checkpoint.

The value specified here is later used as the checkpoint interval for subsequent transfers.

### **EXCMBR(start-mbr end-mbr)**

specifies criteria by which file members are excluded from compression. In any case where a member meets the criteria of both the SELMBR and EXCMBR parameters, the EXCMBR specification overrides the selection of the member.



You can type in multiple sets of values for this parameter.

**start-mbr** specifies the name of the member to be excluded from compression or the first name in a range of members to be excluded. If no members are to be excluded, this field should be left blank.

Possible values are:

**member-name** specifies the name of the file member to be excluded from compression.

**generic\*-member-name** specifies a generic member name to exclude all members whose names begin with the specified characters.

**start-range** specifies the first name in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). The first and last members specified in the range, as well as all members between, are excluded from compression.

**end-mbr** specifies the last name in a range of members to be excluded from compression. Possible values are:

**\*STARTMBR** specifies that the name of the last member in the range is the same as the first member in the range, limiting the range to a single member. You should also use this value if you specified a generic in the starting member name (start-mbr) field. This is the default.

**end-range** specifies the last name in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). The first and last members specified in the range, as well as all members between, are excluded from compression.

#### **REPLACE (\*NO|\*YES)**

specifies whether to replace the output member if it exists. This parameter applies to all members.

Possible values are:

**\*NO** specifies that the member is not replaced if it exists. \*NO is the default.

**\*YES** specifies that the member is replaced if it exists.

#### **SELMBR(start-mbr end-mbr new-mbr)**

specifies criteria by which file members are selected for compression. In any case where a member meets the criteria of both the SELMBR and EXCMBR parameters, the EXCMBR specification overrides the selection of the member.

You can type in multiple sets of values for this parameter.

**start-mbr** specifies the name of the member to be compressed or the first name in a range of members to be compressed. Possible values are:

**\*INFILE** specifies the name of the member is the same as the name of the input file. \*INFILE is the default.

**\*ALL** indicates all members in the input file is selected for compression.

**member-name** specifies the name of the file member to be compressed.

**generic\*-member-name** specifies a generic member name to select all members whose names begin with the specified characters.

**start-range** specifies the first name in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). The first and last members specified in the range, as well as all members between, are compressed.

**end-mbr** specifies the last name in a range of members to be compressed. The possible values are:

**\*STARTMBR** indicates that the last member in the range is the same as the first member in the range, limiting the range to a single member. You should also use this value if you specified a generic in the Starting member name (start-mbr) field. **\*STARTMBR** is the default.

**end-range** specifies the last name in an alphanumeric range of members. Although member names in a range are treated as generics, they cannot be used with an asterisk (\*). The first and last members specified in the range, as well as all members between, are compressed.

**new-mbr** specifies a new name for the member specified in the start-mbr field. If **\*ALL**, a generic\*, or a range has been specified for member selection, this parameter must contain the **\*NONE** value. Possible values are:

**\*NONE** indicates the name of the output member is the same as the name of the input member.

**\*NONE** is the default.

**newname** specifies the new name for the output member.

---

## Using the CDCOMP Screen

Use the **Connect:Direct Compress File (CDCOMP)** screen to compress a file on the i5/OS node before sending it to a remote node.

### Compressing a File Prior to Sending

Complete the following steps to compress a file before sending it to a remote node:

1. Access the **CDCOMP** screen by typing **CDCOMP** at the command line and pressing **F4** or by selecting the Compress file option from the main menu.
2. Type values for parameters as appropriate and press **Enter**.

If you need additional entry fields for the fields that allow multiple sets of values, type a plus sign (+) in the field labeled + for more values and press **Enter**.

The following figure is a sample **Connect:Direct Compress File (CDCOMP)** screen.

```

Connect:Direct Compress File (CDCOMP)

Type choices, press Enter.

Input file name . . . . . _____ Name
  Library . . . . . *LIBL Name, *LIBL
Output file name . . . . . _____ Name
  Library . . . . . *CURLIB Name, *CURLIB
File Members Selection List:
  Starting member name . . . . . *INFILE Name, generic*, *INFILE, *ALL
  Ending member name . . . . . *STARTMBR Name, *STARTMBR
  New member name . . . . . *NONE Name, *NONE
    + for more values
Replace members . . . . . *NO *NO, *YES
File Members Exclusion List:
  Starting member name . . . . . _____ Name, generic*
  Ending member name . . . . . *STARTMBR Name, *STARTMBR
    + for more values
Checkpoint Interval . . . . . *INITPARMS 0-9999999, *INITPARMS

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```



---

# Controlling Processes

Connect:Direct for i5/OS implements a Connect:Direct Process Management Queue. You can use two commands to schedule, control, and monitor any active or queued Processes:

- ❖ CDSUBPROC command allows you to:
  - ◆ Submit Connect:Direct Processes for asynchronous execution
  - ◆ Control certain characteristics of the execution
- ❖ WRKCDPROC command allows you to work with active or queued Connect:Direct Processes. With this command, you can perform the following Process tasks:
  - ◆ Schedule
  - ◆ Control
  - ◆ Monitor

This chapter explains the Process Management Queue, the CDSUBPROC command, and the WRKCDPROC command and their use for Process management.

---

## Understanding the Process Management Queue

The Process Management Queue is a repository for Processes currently in the Connect:Direct for i5/OS system. Five logical queues are associated with the Process Management Queue:

- ❖ ACTIVQ - active, executing Processes
- ❖ WAITQ - Processes waiting to run
- ❖ HOLDQ - Processes currently held
- ❖ TIMERQ - Processes waiting for a timer event
- ❖ STARTQ - Processes being activated

---

## Understanding Asynchronous and Synchronous Processes

The Process Management Queue contains synchronous and asynchronous Processes.

Asynchronous Processes are those submitted using the CDSUBPROC command. These Processes are scheduled and maintained according to run instructions provided in the CDSUBPROC command.

Synchronous Processes are Connect:Direct commands issued through CL programs and CL commands other than CDSUBPROC. They cannot be scheduled once they are in the Process Management Queue.

The Process Management Queue handles all processes:

- ❖ Local Synchronous (Examples: CDSND, CDRCV, CDRUNTASK)
- ❖ Local Asynchronous (CDSUBPROC)
- ❖ Remote

---

## Submitting Processes for Asynchronous Execution

The CDSUBPROC command allows you to submit a Connect:Direct Process to the Process queue for asynchronous execution. This command also allows you to determine:

- ❖ When the Process starts
- ❖ Whether the Process is retained in the queue after execution
- ❖ Whether the Process is placed in the HOLD queue initially

---

**Note:** For TDSN and FDSN, the values of \*LIBL, \*CURLIB, QTEMP cannot be used when sending to or from an i5/OS system using Connect:Direct for i5/OS. A specific library name must be used for copying from or to a data set on the i5/OS.

---

## CDSUBPROC Command Example

The following command example shows the CDSUBPROC command used to submit an asynchronous Process.

```
CDSUBPROC CMD(CDSND SNODE(remote.node) SNODENVIRON(Windows)
FDSN('library/file(mbr)') TDSN('\\system\drive\directory\file')
FMSYSOPTS('TYPE(mbr)')) STARTT('11/22/96' *NOON *WEEK) RETAIN(*YES)
```

## CDSUBPROC Command Format

The following format is for the CDSUBPROC command. Required parameters are shown in bold print.

Command	Parameters
CDSUBPROC	<b>CMD(command string)</b>
	HOLD(*YES *NO)
	PRIORITY( <i>n</i> )
	RETAIN(*YES *NO *INITIAL)
	STARTT(day,time,freq)

Command	Parameters
	TRACEOPT Trace Instructions: (L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT  *COMMON *INITPARMS *CURRENT Command Trace Options: L,S,F,C,X,K,D,H,V,P *ALL *EXT *COMMON  *NONE)

## Required Parameter

The CMD parameter is required for the CDSUBPROC command.

### **CMD(command string)**

specifies the Connect:Direct command string to execute. This value must be a valid Connect:Direct for i5/OS command with any required parameters and subparameters. Connect:Direct checks the command and its parameters at submission. If the command or the parameters are not valid, the submission fails with an error message.

## Optional Parameters

The following parameters are optional for the CDSUBPROC command.

### **HOLD(\*YES|\*NO)**

specifies whether Connect:Direct places the Process in the HOLD queue upon submission.

\*YES places the Process in the HOLD queue immediately. HOLD(\*YES) is mutually exclusive with the STARTT parameter.

\*NO prevents Connect:Direct from placing the Process in the HOLD queue.

### **PRIORITY(n)**

specifies a priority value for the submitted Process. This parameter determines the Process position in the Process WAIT queue and can affect the order of Process execution. The value is 1–15, with 15 being the highest priority. The default value is 7.

### **RETAIN(\*YES|\*NO|\*INITIAL)**

specifies whether Connect:Direct keeps a copy of the Process in the HOLD queue after the Process executes.

\*YES places the Process in the HOLD queue after execution.

\*NO deletes the Process after execution.

\*INITIAL indicates that the Process executes every time the Connect:Direct for i5/OS node initializes. If you select \*INITIAL, the Process does not execute when you submit it.

### **STARTT(day,time,freq)**

specifies when the submitted command runs. Day and time together indicate the scheduled time. Frequency indicates a repeated execution. If you specify frequency, Connect:Direct ignores any RETAIN parameter.

**day** specifies the day of the week when the Process is executed. You can specify any of the following: \*NONE, \*SUN, \*MON, \*TUE, \*WED, \*THU, \*FRI, \*SAT, \*TODAY, \*TOMORROW, or the date. If you specify the date, you must give the date in the job date format. Input dates are interpreted according to the date format of the current job (i5/OS job date format).

If you use day without a time, the time defaults to 00:00 (midnight). The default value is \*NONE, which indicates no scheduled day.

**time** indicates the time of day when the Process runs. You can specify time as any of the following: \*NONE, \*NOON, \*MIDNIGHT, or hh:mm:ssXM.

If you give time in the hh:mm:ssXM format, you can enter the time with or without separators. Minutes and seconds are optional. If you do not enter minutes and seconds, they default to 00. If you specify AM or PM, Connect:Direct assumes the time is in 12-hour format. If you do not select AM or PM, Connect:Direct assumes the time is in 24-hour format.

If you specify time but not day, the Process is run on the first occurrence of the specified time.

**frequency** indicates how often the Process runs. You can determine the frequency by entering one of the following selections: \*NONE, \*DAY, or \*WEEK.

If you enter \*NONE, the Process is run only once automatically. \*DAY indicates that the Process is run every day at the same time after the first scheduled time. \*WEEK releases the Process weekly at the same time and day of the week as the first execution.

## TRACEOPT

### Trace Instructions

(L, S, F, C, X, K, D, H, V, W, P, |\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)
- ❖ **\*EXT** enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ **\*COMMON** enables the communications traces. (L, S, F, C, and H)
- ❖ **\*INITPARMS** indicates the traces specified in the initialization parameters.
- ❖ **\*CURRENT** selects the trace bits that the session manager is currently using.

### Command Trace Options

L,S,F,C,X,K,D,H,V,P|\*ALL|\*EXT|\*COMMON| \*NONE)

specifies the trace bits that define functions to be traced for this command but not for the Process it runs. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.



- ❖ **F** enables the file handler trace.
- ❖ **C** enables the common logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, and P)
- ❖ **\*EXT** enables all product traces plus two other traces. (TRCICF and TRCJOB)
- ❖ **\*COMMON** enables the most commonly used traces. (L, S, F, C, and H)
- ❖ **\*NONE** turns off all traces.

---

## Using the CDSUBPROC Screens

Use the **Connect:Direct Submit Process (CDSUBPROC)** screen to submit a Connect:Direct Process to the Process queue for asynchronous execution.

### Submitting a Process for Asynchronous Execution

Complete the following steps to submit a Process to the Process queue for asynchronous execution:

1. Access the **CDSUBPROC** screen by typing **CDSUBPROC** at the command line and pressing **F4** or by choosing the submit process option from the Connect:Direct menu.
2. Enter the parameters that define the Process execution: **CMD**, **STARTT**, **RETAIN**, **HOLD**, and **PRIORITY**.
3. After you enter the command, press **F4** to display the applicable entry fields for the command.
4. After you complete the parameters for the CDSUBPROC command, press **Enter**. Your Process name and number are assigned and displayed. The Process is placed in the Process Management Queue.

The following screen is an example of the **Connect:Direct Submit Process** screen.

```

Submit C:D Process (CDSUBPROC)

Type choices, press Enter.

Connect:Direct command . . . . . CMD_____
_____
_____
_____
_____
_____
_____
...

Execution date & time:
  Start date . . . . . *NONE_____ Character value, *SUN...
  Start time . . . . . *NONE_____ Character value, *NONE...
  Frequency . . . . . *NONE_____ *NONE, *DAY, *WEEK
Retain process after execution *NO_____ *NO, *YES, *INITIAL
Hold process on queue . . . . . *NO_____ *NO, *YES
Priority on queue . . . . . 7_____ 1-15

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display   F24=More keys
    
```

## Working with Connect:Direct Processes

Use the WRKCDPROC command to schedule, monitor, and control the Processes. The following options are available:

### Selecting Specific Processes

Selection options allow you to specify the Processes with which you work.

### Viewing All Processes

The Display option allows you to view detailed information for all Processes, including execution information for active Processes, both synchronous and asynchronous.

---

**Note:** Your system administrator specifies the user's authority. If you do not have administrative authority, you can only select Processes that you submit.

---

### WRKCDPROC Command Example

The following command example shows the WRKCDPROC command used to work with Connect:Direct Processes.

```
WRKCDPROC QNAME(ACTIVQ) PNAME(CDSND) PNUM(1896)
```

## WRKCDPROC Command Format

The following syntax is for the WRKCDPROC command.

Command	Parameters
WRKCDPROC	PNAME(*ALL name generic)
	PNUM(n *ALL)
	PUSER(*ALL name generic)
	QNAME(ACTIVQ HOLDQ WAITQ TIMERQ STARTQ *ALL)
	RMTNODE(*ALL name generic)
	TYPE(_ L R S A)

### Required Parameters

There are no required parameters for the WRKCDPROC command. If you do not specify any parameters, Connect:Direct displays information for all Processes that you are authorized to view.

### Optional Parameters

The following parameters are optional for the WRKCDPROC command.

#### **PNAME(\*ALL|name|generic)**

specifies the Connect:Direct Processes you are selecting. The PNAME parameter can contain 1 to 12 characters and has the following options:

**\*ALL** indicates that you select all Process names.

**name** is the name of a specific Process you want to select. Enter the name of a Connect:Direct Process.

**generic** identifies a generic Process name. For example, CD\* selects all Processes that begin with CD.

#### **PNUM(n|\*ALL)**

specifies the Process number for the selected Process. This parameter has two options:

**\*ALL** indicates that you select all Process numbers.

**n** is the number of a specific Process you want to select. Enter the number of a Connect:Direct Process.

#### **PUSER(\*ALL|name|generic)**

specifies the Connect:Direct Processes submitted by a specific user. This parameter has the following options:

**\*ALL** indicates that you choose all submitter IDs.

**name** is the user ID of a specific user whose Processes you want to select. Enter the user ID of a Connect:Direct user.

**generic** identifies a generic user name. For example, TH\* would select Processes submitted by all users whose submitter IDs begin with TH.

**QNAME(ACTIVQ|HOLDQ|WAITQ|TIMERQ|STARTQ|\*ALL)**

specifies the Connect:Direct queue for the Processes you want to select. This parameter has six options:

**ACTIVQ** selects all Processes that are currently executing.

**HOLDQ** selects all Processes that are currently in the HOLD queue.

**WAITQ** selects all Processes that are waiting for a session manager to run.

**TIMERQ** selects all Processes that are awaiting execution time or retry time after a session failure.

**STARTQ** selects all Processes that are ready for execution.

**\*ALL** selects all Processes.

**RMTNODE(\*ALL|name|generic)**

specifies the selection of Processes that have the specified remote node name. The RMTNODE parameter has the following three options:

**\*ALL** indicates all remote node names.

**name** indicates the name of a specific remote node.

**generic** indicates a generic node name. For example, CD\* would select all remote node names that begin with CD.

**TYPE(\*|L|R|S|A)**

specifies the category or type of Connect:Direct Process to select. Five values are options for the TYPE parameter:

**\*** specifies all Processes.

**L** specifies all Processes initiated on the local node.

**R** specifies all Processes initiated on the remote node.

**S** specifies all synchronous Processes initiated on the local node.

**A** specifies all Processes except synchronous processes initiated on the local node.

---

## Using the WRKCDPROC Screens

Use the **Work with Connect:Direct Process (WRKCDPROC)** screen to monitor and control Processes in the subsystem.

### Monitoring and Controlling Processes

Complete the following steps to monitor or control a Process:

1. Access the **WRKCDPROC** screen by typing **WRKCDPROC** at the command line and press **F4** or choose the Work with Connect:Direct Processes option from the Connect:Direct main menu.
2. Complete the information requested on the screen and press **Enter**.

The following screen is an example of the **Work with Connect:Direct Process** prompt screen.

```

Work with C:D Process (WRKCDPROC)

Type choices, press Enter.

Process Type . . . . . *          *, L, R, S, A
Process Queue . . . . . *ALL      *ALL, ACTIVQ, HOLDQ, WAITQ...
Process Name . . . . . *ALL      Character value
Process number . . . . . *ALL     Character value
Processing User . . . . . *ALL    Character value
Remote Node . . . . . *ALL

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Bottom
    
```

The next screen is a sample of the **WRKCDPROC** screen (View 1) that displays in response to the information you entered on the prompt screen.

```

Work with C:D Process

Type option and press Enter.
  2=Change  3=Hold  4=Delete  5=Display  6=Release  7=End  8=Suspend

Process          Process          Remote
Opt  Name        Number  Ty  Queue  Submitter  Node        Task
CDSND  00014895  A  HOLDQ  IDEVUSER  FRAN.CDDV3600

F1=Help  F3=Exit  F5=Refresh  F11=View 2  F12=Cancel  F21=Command line
Bottom
    
```

Press **F11** to display View 2, which provides additional information about Processes you selected. You can select an option to monitor or change Processes from this screen or the View 1 screen.

Work with C:D Process								
Type option and press Enter.								
		2=Change	3=Hold	4>Delete	5=Display	6=Release	7=End	8=Suspend
Opt	Process Name	Process Number	Ty	Security ID	Scheduled Date	Scheduled Time	Retry	Prty
	CDSND	00014895	A	IDEVUSER		: :	00	07
								Bottom
F1=Help		F3=Exit	F5=Refresh	F11=View 2	F12=Cancel	F21=Command line		

## WRKCDPROC Option Descriptions

The seven options you can select from the **Work With Connect:Direct Process** screens are described in the following paragraphs. To select these options, type the appropriate number in the option column next to the applicable Process.

### Changing Process Characteristics

**Change** (Option 2) allows you to modify certain Process characteristics. This option is only valid for asynchronous local Processes. The date format is subject to the interpretation of the job currently running.

### Moving an Nonexecuting Asynchronous Process

**Hold** (Option 3) allows you to move a nonexecuting asynchronous Process (WAIT or TIMER queue) to the HOLD queue. No additional screen is displayed.

### Removing an Nonexecuting Asynchronous Process

**Delete** (Option 4) allows you to remove a nonexecuting asynchronous Process from the system. The Process is in any queue except ACTIVE or WAITQ when you delete it. When you delete a Process, a confirmation panel displays.

### Displaying Process Details

**Display** (Option 5) provides you the option to display details about any Process in the system. A sample of the Process information follows.

---

**Note:** If Connect:Direct Secure+ Option is installed and you have proper authority, a **WRKCDPROC** screen containing Secure+ Option information displays after you press **Enter**. Refer to the *Connect:Direct Secure+ Option for i5/OS Implementation Guide* for additional information.

---

The following is an example of the **Display C:D Process** screen.

```

                                Display C:D Process

Process Name . . : CDSND                Queue . . . . : HOLDQ
Number . . . . : 00014895             Remote Node . . : CDDV3600
Submitter . . . : LCLUSER              Priority . . . . : 07
Frequency . . . : *NONE                Retain . . . . : *NO
Submit Date . . : 03/16/04             Submit Time . . : 11:59:28
Schedule Date . :                      Schedule Time . :

----- Activity Details -----

Function . . . :
From File . . :
From Member . . :
To File . . . :
To Member . . . :
File Bytes . . :          0
Transmitted . . :          0
Compression % . :    0.00
Check point . . : *NO
% Complete . . . : N/A

F1=Help  F3=Exit  F5=Refresh  F12=Cancel  F21=Command line

```

## Releasing a Process in the HOLD Queue

**Release** (Option 6) permits you to release a Process in the HOLD queue and move it to the WAIT queue. There it is available for execution. No additional screen displays.

## Terminating and Flushing an Executing PNODE Process

**End** (Option 7) terminates and flushes any executing synchronous or asynchronous PNODE process from the system. When using this option, use a different terminal from the one where the Process is submitted. If you use the same terminal, the next submitted Process fails. A confirmation panel displays when you end a Process. The default for this option is \*CNTRL D. The value \*IMMED should not be used unless \*CNTRL D has first been attempted.

## Interrupting an Executing Asynchronous Process

**Suspend** (Option 8) allows you to interrupt any executing asynchronous Process. This option also allows you to move the Process to the HOLD queue (LOCAL). You can then release the (LOCAL) Process from the HOLD queue to resume execution or delete it. If the COPY operation was checkpointed, you can use suspend/release to interrupt the COPY and later resume it from the latest checkpoint.

## Releasing or Holding Asynchronous Processes

RLSCDPROC and HLDCDPROC allow you to release all held Processes or all Processes currently in the TIMER queue that meet the selection criteria identified when the command is entered. The commands can be used in either batch or interactive mode. Only local asynchronous Processes are considered for release. Following are sample RLSCDPROC and HLDCDPROC screens.

### RLSCDPROC Screen

```
Release Held C:D Process (RLSCDPROC)

Type choices, press Enter.

Process Type . . . . . A          A
Process Queue . . . . . HOLDQ     HOLDQ
Process Name . . . . . *ALL       Character value
Process number . . . . . *ALL     Character value
Processing User . . . . . *ALL     Character value
Remote Node . . . . . *ALL

F1=Help  F3=Exit  F5=Refresh  F12=Cancel  F21=Command line
```

### HLDCDPROC Screen

```
Hold Specified C:D Process (HLDCDPROC)

Type choices, press Enter.

Process Type . . . . . A          A
Process Queue . . . . . WAITQ     WAITQ, TIMERQ
Process Name . . . . . *ALL       Character value
Process number . . . . . *ALL     Character value
Processing User . . . . . *ALL     Character value
Remote Node . . . . . *ALL

F1=Help  F3=Exit  F5=Refresh  F12=Cancel  F21=Command line
```



---

# Using Connect:Direct Statistics

Connect:Direct for i5/OS logs statistics information for all Processes. Each component of Connect:Direct for i5/OS generates statistics data to the statistics manager (STATMGR) job. The long-running STATMGR job formats the statistics information and writes it to the statistics file. You can view the results of the Processes by using the CDSELSTAT command. This chapter explains how to use the CDSELSTAT command, the command format, and the **CDSELSTAT** screens. For detailed information about the statistics facility, refer to the *Connect:Direct for i5/OS Installation and Administration Guide*.

---

## Selecting Statistics Information

Use the CDSELSTAT command to select statistics information for viewing or printing.

## Specifying Selection Criteria

Specify several selection criteria to limit the records selected. The command selects records from either the active statistics file or a named statistics database file. Within the file, all members are available to CDSELSTAT and are viewed together as a single logical file.

## CDSELSTAT Command Example

The following figure presents a sample CDSELSTAT command.

```
CDSELSTAT PERIOD('10:00:00' '11/13/98') PNAME(CDSND)
```

## CDSELSTAT Command Format

The following table displays the command format for the CDSELSTAT command.

Command	Parameters
CDSELSTAT	FILE(filename[*ACTIVE])
	FORMAT(*SUMMARY *DETAIL)
	OUTPUT(␣ *PRINT)

Command	Parameters
	PERIOD([start-time, end-time]*NOON *MIDNIGHT *AVAIL] [start-date, end-date]*TODAY *AVAIL])
	PNUM(process-number *ALL)
	PNAME(process-name generic *ALL)
	PUSER(local-userid generic *ALL)
	RCDTPYE(record-type *ALL)
	RMTNODE(remote-node-name *ALL)
	TRACE(L,S,F,C,X,K,D,H,V,W,P *NONE *ALL *EXT *COMMON *INITPARMS *CURRENT)

## Required Parameters

The CDSELSTAT command has no required parameters.

## Optional Parameters

The following parameters are optional for the CDSELSTAT command.

**FILE(filename|\*ACTIVE)** allows you to specify the statistics database file to search.

**filename** specifies a statistics database file.

**\*ACTIVE** directs the command to search the file to which Connect:Direct is currently logging statistics.

**FORMAT(\*SUMMARY|\*DETAIL)** controls how the selection results are presented. Connect:Direct for i5/OS presents the selected records in timestamp sequence regardless of the output format chosen. FORMAT allows you to select two formats.

**\*SUMMARY** specifies that the summary information for the selected record is presented as a single row in a scrollable list.

**\*DETAIL** specifies that the entire contents of information for the selected record is formatted into a spooled file. You can view this file at a display station, or you can queue the file to be printed. Each format is displayed or printed. When you use FORMAT(\*SUMMARY) with OUTPUT(\*), you can toggle between the two views.

**OUTPUT(\*|\*PRINT)** controls how the selection results are output. OUTPUT allows you to select two options.

**\*** specifies that the statistics information displays on the screen.

**\*PRINT** specifies that the statistics information is created as a spool file named CDSELSTAT.

**PERIOD([start-time, end-time]\*NOON|\*MIDNIGHT|\*AVAIL] [start-date, end-date]\*TODAY|\*AVAIL])** allows the search to be limited to a specific period of time rather than including the entire range available in the statistics file. The parameters for the PERIOD keyword are positional.

**start-time** and **end-time** specify the records being requested by giving the starting time and the ending time for statistics selection. Both start-time and end-time accept any one of the following subparameters:

**hh:mm:ss XM** indicates a beginning and/or an ending time to use as selection criteria for the requested records. Indicate the times as follows: hh specifies hours, mm specifies minutes, ss specifies seconds, XM specifies AM or PM. If you do not specify AM or PM, Connect:Direct assumes a 24-hour clock.

**\*NOON** specifies 12:00:00 for the time value

**\*MIDNIGHT** specifies 00:00:00 for the time value.

**\*AVAIL** specifies the earliest available time for the date specified.

**start-date** and **end-date** indicate the records being selected by date criteria for statistics selection. Both start-date and end-date accept one of the following formats:

mmddy, mm/dd/yy, mm.dd.yy.

Input dates are interpreted according to the date format of the current job (i5/OS job date format).

**\*TODAY** specifies the current date. The starting and ending date fields default to **\*TODAY**.

**\*AVAIL** specifies the earliest and latest date in the file. The starting and ending time fields default to **\*AVAIL**.

**PNUM(process-number|\*ALL)** specifies the number of the Process to be selected.

**PNAME(process-name|generic|\*ALL)** specifies the name of the Process to be selected. The PNAME parameter can contain 1 to 12 characters and allows generic selections.

**PUSER(local-userid|generic|\*ALL)** specifies the local user ID of the Process to be selected. This parameter allows generic selections.

**RCDTYPE(record-type|\*ALL)** specifies the record type of the Process to be selected.

**RMTNODE(remote-node-name|\*ALL)** specifies the remote node name of the Process to be selected. This parameter allows generic selections.

## TRACE

Trace Instructions

(L, S, F, C, X, K, D, H, V, W, P, |\*NONE|\*ALL|\*EXT| \*COMMON|\*INITPARMS|\*CURRENT)

specifies the trace bits that define functions to be traced by Connect:Direct for i5/OS. The following are valid options:

- ❖ **L** enables the main line logic trace for all components.
- ❖ **S** enables the Process step logic trace.
- ❖ **F** enables the file handler trace.
- ❖ **C** enables the communications logic trace.
- ❖ **X** enables the XDR trace.
- ❖ **K** enables the KQV logic trace.
- ❖ **D** enables the all-purpose data dump.
- ❖ **H** enables the FMH dump.
- ❖ **V** enables the XDR dump.
- ❖ **W** enables the communications dump.
- ❖ **P** enables the CIPC dump.
- ❖ **\*NONE** turns off all traces.
- ❖ **\*ALL** enables all product traces.(L, S, F, C, X, K, D, H, V, W and P)

- ❖ \*EXT enables all product traces plus two system traces (TRCICF and TRCJOB).
- ❖ \*COMMON enables the communications traces. (L, S, F, C, and H)

## Using the CDSELSTAT Screens

Use the **CDSELSTAT** screen to select statistics information for viewing or printing. You can specify several selection criteria to limit the records selected or choose a summary of all records. In this section, two methods of displaying and printing both summary and detail statistics are shown.

### Selecting and Displaying Summary Statistics

Complete the following steps to select and display summary statistics:

1. Access the **CDSELSTAT** screens by typing **CDSELSTAT** at the command line and pressing **F4** or by choosing the select statistics option (13) from the Connect:Direct main menu.  
The following screen is a sample of the first screen in the **Connect:Direct for i5/OS CDSELSTAT** screen flow.

```

                                Select C:D Statistics (CDSELSTAT)

Type choices, press Enter.

Input file name . . . . . *ACTIVE__  Name, *ACTIVE
  Library . . . . . _____  Name, *LIBL
Time period for selection.:
  Period starting time . . . . . *AVAIL__  Time, *AVAIL
  Period starting date . . . . . *TODAY__  Date, *TODAY, *AVAIL
  Period ending time . . . . . *AVAIL__  Time, *AVAIL
  Period ending date . . . . . *TODAY__  Date, *TODAY, *AVAIL
Process number . . . . . *ALL__      Number, *ALL
Process name . . . . . *ALL__      Name, *ALL
User ID . . . . . *ALL__      Name, *ALL
Remote node name . . . . . *ALL__
Statistics record type . . . . . *ALL__      Record type, *ALL
      + for more values
      _____
Output destination . . . . . *          *, *PRINT
Output format . . . . . *SUMMARY    *SUMMARY, *DETAIL
Trace bits string . . . . . *NONE__  *NONE, trace string

                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
    
```

2. To display summary information, press **Enter** (\* is the default Output destination and \*SUMMARY is the default Output format). The following screen, **Select C:D Statistics**, is displayed. The displayed summary information includes an option column where you can request to display or print detailed information about a particular statistics record. Type **5** in the option column for the Record type you want to review to

display the detailed information and press **Enter**.

**Note:** You could also type **6** and press **Enter** for a print version of the detail list, which is spooled for printing to the default output queue. See the next section for information on retrieving a spooled file.

```

                               Select C:D Statistics

Process number . . . *ALL          Statistics file : CDSTATFILE
Process name . . .  *ALL          Library . . . . : CDDV3600
                               Position to date

Type options, press Enter.
   5=Display details   6=Print details

   Process  Process  Record  Step  Event  Event  Message
   Opt  number  name      type  name   time   date   ID
   --  -
   5           GNMSG      09:15:46 03/16/04
   -           JOBEND      09:15:46 03/16/04
   -           GNMSG      09:15:51 03/16/04
   -           TLSES       09:16:53 03/16/04
   -           JOBSUB      09:16:53 03/16/04
   -           GNMSG      09:16:53 03/16/04
   -           JOBSTR      09:16:55 03/16/04
   -           GNMSG      09:16:55 03/16/04
                                           More...

Parameters or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F11=View 2   F12=Cancel

```

The following screen shows the results of using the display option, Output destination(\*).

```

                               Display Spooled File

File . . . . . : CDSELSTAT                               Page/Line 1/6
Control . . . . .                               Columns 1 - 130
Find . . . . .
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...
C:D i5          Select Statistics (CDSELSTAT) Detail 03/16/04 12:10:55 Page 0001
-----
Event:           General Message Event (GNMSG)
Date/Time:       03/16/04      09:15:46
Text:           Connect:Direct job ended: 186578/CDDV3600/CDSMGR, CODE=N.

```

**Note:** If you have admin authority and Connect:Direct Secure+ Option is installed, the **CDSELSTAT** screen displays Secure+ Option information. Refer to the *Connect:Direct Secure+ Option for i5/OS Implementation Guide* for additional information.

## Selecting and Printing Detail Statistics

Complete the following steps to select and print detail statistics:

1. Access the **CDSELSTAT** screens by typing **CDSELSTAT** at the command line and pressing **F4** or by selecting the select statistics option from the Connect:Direct main menu.

The first CDSELSTAT screen is displayed.

```

                                Select C:D Statistics (CDSELSTAT)

Type choices, press Enter.

Input file name . . . . . *ACTIVE_____ Name, *ACTIVE
Library . . . . . _____ Name, *LIBL
Time period for selection.:
  Period starting time . . . . . *AVAIL_____ Time, *AVAIL
  Period starting date . . . . . *TODAY_____ Date, *TODAY, *AVAIL
  Period ending time . . . . . *AVAIL_____ Time, *AVAIL
  Period ending date . . . . . *TODAY_____ Date, *TODAY, *AVAIL
Process number . . . . . *ALL_____ Number, *ALL
Process name . . . . . *ALL_____ Name, *ALL
User ID . . . . . *ALL_____ Name, *ALL
Remote node name . . . . . *ALL_____
Statistics record type . . . . . *ALL_____ Record type, *ALL
      + for more values _____
Output destination . . . . . *PRINT_____ *, *PRINT
Output format . . . . . *DETAIL_____ *SUMMARY, *DETAIL
Trace bits string . . . . . *NONE_____ *NONE, trace string

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

2. To print a detail report of the CDSELSTAT file, change the Output destination to **\*PRINT** and the Output format to **\*DETAIL**, and then press **Enter**. Connect:Direct for i5/OS creates a print version of the detail list and spools it for printing to the default output queue.

---

**Note:** The default output queue is named CDOUT and is located in the Connect:Direct library.

---

- To display the list of spooled files in the output queue, at the command line, type **WRKOUTQ** *cdlib*/**CDOUT** where *cdlib* is the name of your Connect:Direct library.

```

                                Work with Output Queue

Queue: CDOUT           Library: CDLIB           Status: RLS

Type options, press Enter.
 1=Send  2=Change 3=Hold  4=Delete  5=Display  6=Release  7=Messages
 8=Attributes  9=Work with printing status

Opt  File           User           User Data  Sts  Pages  Copies  Form Type  Pty
--  -
_   CDTRACE        CDDV3600      SMGR       RDY   159    1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDTRACE        CDDV3600      RMTSYSTEM  RDY    5     1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDLOG          CDDV3600              RDY    1     1      *STD       5
_   CDSELSTAT     LCLUSER              RDY   27     1      *STD       5
                                          Bottom

Parameters for options 1, 2, 3 or command
====>
F3=Exit  F11=View 2  F12=Cancel  F20=Writers  F22=Printers
F24=More keys

```

- In the Opt column, type **5** next to the CDSELSTAT spooled file and press **Enter** to go to the Display Spooled File screen. (If you have a lot of files in the output queue, you can press **F18** to quickly go to the bottom of the spooled file list.)

The following screen shows the results of using the display option, Output destination, (\*PRINT).

```

                                Display Spooled File
File . . . . . : CDSELSTAT                                Page/Line  1/6
Control . . . . .
Columns      1 - 130
Find . . . . .

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...
+...9...+...0...+...1...+...2...+...3
C:D i5          Select Statistics (CDSELSTAT) Detail 03/18/04 15:45:05 Page 0001

-----
Event:          Job Submitted (JOBSUB)
Date/Time:     03/10/04    11:09:07
Job Name:      CDSMGR                      Job Number:  186005
Job Program:   SMMAIN

-----
Event:          General Message Event (GNMSG)
Date/Time:     03/10/04    11:09:07
Text:         C:D i5 Job submitted: CDSMGR.

-----
Event:          Job Started (JOBSTR)
Date/Time:     03/10/04    11:09:09
Job Name:      CDSMGR                      Job Number:  186005
Job Program:   SMMAIN

-----
Event:          General Message Event (GNMSG)
Date/Time:     03/10/04    11:09:10
Text:         Connect:Direct job started: 186005/CDDV3600/CDSMGR

More...
F3=Exit  F12=Cancel  F19=Left  F20=Right  F24=More keys
    
```



---

# Decompressing Files

This chapter describes the Connect:Direct decompress (CDDECOMP) command and its function. This chapter also shows sample **CDDECOMP** screens and describes the parameters used for the CDDECOMP command.

---

## Using the CDDECOMP Command

Files are received from another node and archived on the i5/OS node as compressed files. Use the CDDECOMP command to decompress these files.

---

**Note:** Tuning parameters are not supported with the CDDECOMP command when sending a file.

---

## CDDECOMP Command Example

The following command is an example of how to code the CDDECOMP command.

```
CDDECOMP INFILE(TESTLIB/CMP_FILE1) OUTFILE(TESTLIB/FILE2)
REPLACE(*YES)
```

## CDDECOMP Command Format

The following are the parameters for the CDDECOMP command:

Command	Parameters
CDDECOMP	INFILE(library-name/file-name)
	OUTFILE(library-name/file-name)
	NEWNAME(current-member-name, new-member-name)
	REPLACE(*NO *YES)

## Required Parameters

The CDDECOMP command has two required parameters.

### **INFILE(library-name/file-name)**

specifies the name and library of the file to be decompressed. The default for library is \*LIBL.

**library-name** specifies the name of the library to be used.

\*LIBL specifies that the library list is used to locate the input file.

\*CURLIB specifies that the current library for the job is used to locate the input file.

**file-name** specifies the name of the file to be decompressed.

### **OUTFILE(library-name/file-name)**

specifies the name and library of the file to receive the uncompressed data. If the output file does not exist, it is created. The default value for library is \*CURLIB.

**library-name** specifies the name of the library to be used.

\*LIBL specifies that the library list is used to locate the input file.

\*CURLIB specifies that the current library for the job is used to locate the output file.

**file-name** specifies the name of the file to receive the decompressed data.

## Optional Parameters

The optional parameters are listed alphabetically. Unless otherwise noted, the parameters apply to all remote node environments.

### **NEWNAME(current-member-name, new-member-name)**

specifies whether to rename selected member(s) when decompressed. You can type in multiple sets of values for this parameter.

**current-member-name** specifies the name of the compressed member to be renamed.

**new-member-name** specifies the new name to be given to the member when it is decompressed.

### **REPLACE(\*NO|\*YES)**

specifies whether to replace the output member if it exists. This parameter applies to all members.

\*NO is the default. Possible values are:

\*NO specifies that the member is not replaced if it exists.

\*YES specifies that the member is replaced if it exists.

---

## Using the CDDECOMP Screen

Use the CDDECOMP screen to decompress a file on the i5/OS node after receiving it from a remote node.

### Decompressing a File

Complete the following steps to decompress a file after receiving it from a remote node:

1. Access the CDDECOMP screen by typing **CDDECOMP** at the command line and pressing **F4** or by selecting the Decompress option from the main menu.
2. Type values for parameters as appropriate and press **Enter**.

If you need additional entry fields for multiple sets of values, type a plus sign (+) in the field labeled + for more values and press **Enter**.

The following figure shows a sample **Connect:Direct Decompress File (CDDECOMP)** screen:

```

                                Connect:Direct Decompress File (CDDECOMP)

Type choices, press Enter.

Input file name . . . . . _____ Name
  Library . . . . . *LIBL_____ Name, *LIBL
Output file name . . . . . _____ Name
  Library . . . . . *CURLIB_____ Name, *CURLIB
Replace members . . . . . *NO_____ *NO, *YES
Rename members:
  Current member name . . . . . _____ Character value
  New member name . . . . . _____ Character value
      + for more values _

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```



---

# Displaying Connect:Direct Message Descriptions

Connect:Direct i5/OS generates messages that detail aspects of the Connect:Direct i5/OS operation. Connect:Direct i5/OS routes two types of messages during operations:

- ❖ Connect:Direct messages that are routed to the Connect:Direct log file
- ❖ Status messages that are routed to a single user ID identified in the Connect:Direct initialization parameters file during installation

You can send Connect:Direct messages to the Connect:Direct user ID or the QSYSOPR message queues. You can also use the Connect:Direct i5/OS command Display Connect:Direct Message File (DSPCDMSGD) to display the message text for Connect:Direct messages.

---

## Specifying Recipient of Status Messages

Complete the following steps to specify the user who receives status messages:

1. Access the **Change Connect:Direct Parameters (CHGCDPARM)** screen by typing **CHGCDPARM** at the command line and pressing **F4** or by choosing the Change Connect:Direct parameters option from the main administration menu.
2. Specify the user ID in the User to Receive Connect:Direct Messages (TOUSR) parameter on the **Change Connect:Direct Parameters (CHGCDPARM)** screen and press **Enter**. This is the user profile that receives Connect:Direct status messages.

Refer to the *Connect:Direct i5/OS Installation and Administration Guide* for details on providing user profile information.

---

## Using the DSPCDMSGD Command

Use the DSPCDMSGD command to display the text associated with a specified Connect:Direct message ID.

### DSPCDMSGD Command Example

The following figures illustrate two ways of accessing a Connect:Direct message.

```
DSPCDMSGD CDMSGID(AFLH001I)
```

```
DSPCDMSGD AFLH001I
```

### DSPCDMSGD Command Format

The following is the parameter for the DSPCDMSGD command:

Command	Parameters
DSPCDMSGD	CDMSGID(msgid)

### Required Parameter

The DSPCDMSGD command has one required parameter.

**CDMSGID(msgid)**  
specifies an eight-character Connect:Direct message ID.

### Optional Parameters

There are no optional parameters for the DSPCDMSGD command.

---

## Using the DSPCDMSGD Screen

Use the **Display Connect:Direct Message (DSPCDMSGD)** screen to display the text for Connect:Direct message IDs that are returned in the Connect:Direct log file and statistics file for each operation.

### Displaying Connect:Direct Message Text

Complete the following steps to display the text for Connect:Direct messages IDs:

1. Access the **DSPCDMSGD** screen by typing **DSPCDMSGD** at the command line and pressing **F4** or by choosing the display message option from the main menu.
2. Type the Connect:Direct message ID and press **Enter**. A screen is displayed with the message text for the specified message ID.

The following screen is an example of the screen associated with the DSPCDMSGD command.

```

                                Display Connect:Direct Message (DSPCDMSGD)

Type choices, press Enter.

Connect:Direct Message ID . . .   SCPA000I       Character value

                                                                 Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
    
```

### Sample Message Description

The following figure shows a sample message description.

```

                                Display C:D Message Description
                                                                 System :   FRAN
Message ID . . . . . :   SCPA000I
Issuing module . . . . . :   DMCOPYRT

Copy step successful.

The process copy step successfully completed.
The Return-code may be 4, in which case a "Level-4"
error has occurred in transmitting a PDS. A Level-4
error indicates that an error was found, but was not
sufficient to cause termination of the COPY step.

SYSTEM ACTION:  The process continues executing.

RESPONSE:  See the specific member entry in the Connect:Direct
STAT file for the detailed message id for a Level-
4 error.

Press Enter to continue.

F3=Exit
    
```





# Appendix A

---

## CL Program Examples

The tables in this appendix provide examples of i5/OS CL Programs using various features of Connect:Direct. These examples are stored in a Physical Source File that is restored to the system during installation.

If PDM is installed, you can then use the WRKMBRPDM command to browse the members.

### i5/OS Compress/Decompress Examples

Member Name	Description
ACMPPDS01	CDCOMP/CDDECOMP Example Using Selmbr
ACMPPDS02	CDCOMP/CDDECOMP Example Using Excmb
AAPROCS	Readme file

### i5/OS Block Mode Transfer Examples

Member Name	Description
ASNDBTBI1	CDSND Block Mode Transfers Using Integrated File System Binary File
ASNDBTBN1	CDSND Block Mode Transfers Using Native File System
ASNDBTP31	CDSND Block Mode Transfer to SPLF
ASNDBTSN1	CDSND Block Mode Transfer to Save File
ASNDBTTN1	CDSND Block Mode Transfer Using Native File System to Text File
ARCVB TBI1	CDRCV Block Mode Transfers Using Integrated File System Binary File
ARCVBTPN1	CDRCV Block Mode Transfer to SPLF
ARCVBTP31	CDRCV Block Mode Transfer to SPLF
ARCVBTSN1	CDRCV Block Mode Transfer to Save File
ARCVBTTN1	CDRCV Block Mode Transfer Using Native File System to Text File
ASPLBTTN1	CDSNDSPL Block Mode Transfer Using Native File System

## i5/OS Run Task Example

Member Name	Description
ARNJDS01	CDRUNTASK Example

## i5/OS Send Example Using Send File Field Description

Member Name	Description
ASNDCF1A	CDSND Using SNDFFD Parameter

## i5/OS Send and Receive Examples Using Native File System

Member Name	Description
ASNDCR1A ASNDCRIA.\$	CDSND Physical Source File
ASNDCR1B	CDSND Physical Data File
ASNDCR1D ASNDCRID.T	CDSND Physical Source File to Physical Data File (PDS)
ASNDCR1E	CDSND Multiple Members of Physical Source File
ASNDCR1F ASNDCRIF.1	CDSND Physical Source File to Spool File Spool File (B G)
ASNDCR1G1	CDSND Physical Data File to Spool File
ASNDCR1N	CDSND Save File to Save File
ASNDCR2A	CDSND Precompressed File
ASNDCR2E 4D, 4E	CDSND Multiple Member Precompressed File EXO1
ARCVC1A ARCVCRIA.S	CDRCV Physical Source File
ARCVC1B	CDRCV Physical Data File
ARCVC1D ARCVCRID.1, 3	CDRCV Physical Source File to Physical Data File
ARCVC1E	CDRCV Multiple Members of Physical Source File
ARCVC1F	CDRCV Physical Source File to Spool File
ARCVC1G1	CDRCV Physical Data File to Spool File
ARCVC1N	CDRCV Save File to Save File
ARCVC2A	CDRCV Precompressed File
ARCVC4A	CDRCV Multiple Member Precompressed File

## i5/OS Send and Receive Examples Using Integrated File System

Member Name	Description
ASNDIF1A	CDSND /QSYS Text
ASNDIF1B ASNDIF1B.D	CDSND /QSYS Binary
ASNDIF1J ASNDIF1J.D	CDSND /QOpenSys
ASNDIF1K	CDSND /root
ASNDIF1KB	CDSND /root and block size specified
ASNDIF1L	CDSND Binary to Text File
ASNDIFN1A	CDSND Integrated File System to Native /QSYS
ASNDIFN1J	CDSND Integrated File System to Native /QOpenSys
ASNDIFN1JB	CDSND Integrated File System to Native /QOpenSys, Using Blocksize
ASNDIFN1KB	CDSND Integrated File System to Native /root, Using Blocksize
ASNDIFN1M	CDSND Integrated File System to Native /root Binary File to Binary File
ASNDIFN1N	CDSND Integrated File System to Native /QOpenSys
ASNDIFN1P	CDSND Integrated File System to Native /QDLS
ASNDIFS1C	CDSND Integrated File System /QSYS to SPLF
ASNDIFS1J	CDSND Integrated File System /QOpenSYS to SPLF
ASNDIFS1K	CDSND Integrated File System /root to SPLF
ASNDIFS1P	CDSND Integrated File System /QDLS to SPLF
ASNDIF01A	CDSND NATIVE File System to Integrated File System /QSYS
ASNDIF01J	CDSND Native File System to Integrated File System /QOpenSys
ASNDIF01K	CDSND Native File System to Integrated File System /root
ASNDIF01L	CDSND Native File System to Integrated File System /QDLS
ARCVIF1A	CDSND /QSYS Text
ARCVIF1B	CDSND /QSYS Binary
ARCVIF1J	CDSND /QOpenSys
ARCVIF1K	CDSND /root
ARCVIF1KB	CDSND /root and Blocksize Specified
ARCVIF1L	CDSND /QDLS Binary to Text File
ARCVIFN1A	CDSND Integrated File System to Native /QSYS
ARCVIFN1J	CDSND Integrated File System to Native /QOpenSys
ARCVIFN1JB	CDSND Integrated File System to Native /QOpenSys, Using Blocksize
ARCVIFN1KB	CDSND Integrated File System to Native /root, Using Blocksize

Member Name	Description
ARCVIFN1M	CDSND Integrated File System to Native /root Binary File to Binary File
ARCVIFN1N	CDSND Integrated File System to Native /QOpenSys
ARCVIFN1P	CDSND Integrated File System to Native /QDLS
ARCVIFS1C	CDSND Integrated File System /QSYS to SPLF
ARCVIFS1J	CDSND Integrated File System /QOpenSYS to SPLF
ARCVIFS1K	CDSND Integrated File System /root to SPLF
ARCVIFS1P	CDSND Integrated File System /QDLS to SPLF
ARCVIF01A	CDSND NATIVE File System to Integrated File System /QSYS
ARCVIF01J	CDSND Native File System to Integrated File System /QOpenSys
ARCVIF01K	CDSND Native File System to Integrated File System /root
ARCVIF01L	CDSND Native File System to Integrated File System /QDLS
ASPLIF0SC	CDSNDSPL to Integrated File System /QSYS
ASPLIF0SJ	CDSNDSPL to Integrated File System /QOpenSys
ASPLIF0SK	CDSNDSPL to Integrated File System /root
ASPLIF0SP	CDSNDSPL to Integrated File System /QDLS

## i5/OS–z/OS Send and Receive Examples

Member Name	Description
MRCVCFGDG	CDRCV GDG Files from z/OS
MRCVCFPDS1	CDRCV PDS File
MRCVCRKFD1	CDRCV KSDS File
MRCVCRKFD5	CDRCV KSDS File Using Blocksize
MRCVCRKF14	CDRCV Using Integrated File System
MRCVCRKFP4	CDRCV KSDS to SPLF
MRCVCRKVS1	CDRCV KSDS VB File to Physical Source File (Records are padded)
MRCVCRSAVF	CDRCV Save File
MRCVCRSF12	CDRCV Using Integrated File System /root
MRCVCRSVI2	CDRCV Using Integrated File System /root
MRCVEBCX1	CDRCV Extended Translation to Integrated File System
MSNDCFGDG	CDSND to z/OS GDG
MSNDCFPDS1	CDSND to z/OS PDS
MSNDCF01D	CDSND of Physical Source File

Member Name	Description
MSNDCRKF1	CDSND of Physical Data Binary File to KSDS
MSNDCRKF5	CDSND of Physical Data File to KSDS Using Blocksize Parameter
MSNDCRKF6	CDSND Using Integrated File System /root
MSNDCRKS1	CDSND to VB KSDS
MSNDCRSVF	CDSND Save File to Save File
MSNDCRSF2	CDSND to VB File from Integrated File System /root
MSNDCRSFS1	CDSND from Physical Source to z/OS Physical Sequential CLP
MSNDDBCS01	CDSND DBCS File
MSNDIF01K	CDSND from Integrated File System /root
MSNDRCVPS3	CDSND/CDRCV with *EXT Compression
MSNDRDR	CDSND JCL to z/OS READER Using SPOOL TRANSFER OPTION
MSPLCRKFP4	CDSNDSPL to z/OS KSDS

## i5/OS–NT Send and Receive Examples

Member Name	Description
NRCVCF01A	CDRCV into Physical Source File
NRCVCF01H	CDRCV Text File to Physical Data File
NRCVCF01I	CDRCV Binary File to Physical Data File
NRCVCF01J	CDRCV Binary File to Save File
NRCVIF01A	CDRCV to Integrated File System /QSYS
NRCVIF01B	CDRCV to Integrated File System /QSYS Physical Data File
NRCVIF01C	CDRCV of Binary File to Integrated File System /QSYS
NRCVIF01H	CDRCV of Text File to Integrated File System
NRCVIF01I	CDRCV of Binary File to Integrated File System
NRCVIF01J	CDRCV of File to Integrated File System /QOpenSys
NRCVIF01K	CDRCV to Integrated File System /root
NRCVIF01L	CDRCV of Binary File to Integrated File System /root
NRCVIF01M	CDRCV of Binary File to /QOpenSys
NRCVIF01N	CDRCV of Binary File to /QDLS
NRCVIF01P	CDRCV to Integrated File System /root
NRCVIF01Q	CDRCV to Integrated File System to /QDLS
NSNDCF01A	CDSND of Physical Source File Using Native File System

Member Name	Description
NSNDCF01H	CDSND of Physical Data to Text File Using Native File System
NSNDCF01I	CDSND of Physical Data File to Binary Using Native File System
NSNDIF01A	CDSND Using Integrated File system /QSYS
NSNDIF01C	CDSND Using Integrated File System /QSYS Physical Data File
NSNDIF01H	CDSND Using Integrated File System to Text File
NSNDIF01I	CDSND Using Integrated File System to Binary File
NSNDIF01J	CDSND Using Integrated File System /QOpenSys
NSNDIF01K	CDSND Using Integrated File System /root
NSNDIF01L	CDSND Using Integrated File System /QDLS
NSNDIF01M	CDSND Using Integrated File System /root and Binary File
NSNDIF01N	CDSND Using Integrated File System /QOpenSys and Binary File
NSNDIF01P	CDSND Using Integrated File System /QDLS and Binary File

## i5/OS–Windows Run Job and Run Task Examples

Member Name	Description
NRNJOB01A	CDRUNJOB Without Desktop Parameter, UNC names
NRNJOB01B	CDRUNJOB Without Desktop Parameter, C: names
NRNJOB01C	CDRUNJOB With Desktop Parameter, NT 1.3.00
NRNTDL01A	CDRUNTASK to Delete Data Set
NSBMCF01A	CDSUBMIT an NT copy Process

## i5/OS–UNIX Send and Receive Examples

Member Name	Description
URCVCF01A	CDRCV Using Native File System into Physical Source File
URCVCF01B	CDRCV Using Native File System Binary File into save File
URCVCF01C	CDRCV Using Native File System Binary to Physical Data File
URCVCF01D	CDRCV Using Native File System Text File to Physical Data File
URCVIF01A	CDRCV Using Integrated File System /QSYS
URCVIF01C	CDRCV Using Integrated File System /QSYS and Binary File
URCVIF01D	CDRCV Using Integrated File System /QSYS and Text File
URCVIF01J	CDRCV Using Integrated File System /QOpenSys

<b>Member Name</b>	<b>Description</b>
URCVIF01K	CDRCV Using Integrated File System /root
URCVIF01L	CDRCV Using Integrated File System /root and Binary File
URCVIF01M	CDRCV Using Integrated File System /QOpenSys and Binary File
URCVIF01N	CDRCV Using Integrated File System /QDLS and Binary File
URCVIF01Q	CDRCV Using Integrated File System and /QDLS
USNDCF01A	CDSND of Physical Source File
USNDCF01H	CDSND of Physical Data Text File
USNDCF01I	CDSND of Physical Data Binary File
USNDCF01S	CDSND of Save File
USNDIF01A	CDSND of Physical Source File Using Integrated File System /QSYS
USNDIF01H	CDSND of Physical Data File Using Integrated File System /QSYS
USNDIF01I	CDSND of Physical Data Binary File Using Integrated File System /QSYS
USNDIF01J	CDSND of Text File Using Integrated File System /QOpenSys
USNDIF01K	CDSND of Text File Using Integrated File System /root
USNDIF01L	CDSND of Text File Using Integrated File System/QDLS
USNDIF01M	CDSND of Binary File Using Integrated File System/root
USNDIF01N	CDSND of Binary File Using Integrated File System /QOpenSys
USNDIF01P	CDSND of Binary File Using Integrated

## i5/OS–UNIX Run Task and Run Job Examples

Member Name	Description
URNJOB01A	CDRUNJOB to List Directory
URNTDL01A	CDRUNTASK to List Directory
URNTDL02A	CDRUNTASK to List Directory
USBMCF01A	CDSUBMIT a UNIX copy Process

## i5/OS–UTIL Test Examples

Member Name	Description
AABNDCL	Creates loopback test nodes
CD\$RETRO	cmd - execute several loopback tests
CD\$RETRO\$	pgm - execute several loopback tests



## A

### **Adjacent Node**

An adjacent node is an entry in the Network Map that defines a Connect:Direct node with which the local Connect:Direct node can communicate. The adjacent node is also referred to as a remote node.

### **Application Program Interface (API)**

The Application Program Interface (API) is a Connect:Direct component that accepts commands and places them in an executable format.

### **Asynchronous**

Asynchronous Process execution occurs when the system starts the job. The CDSUBPROC command execution is asynchronous.

## C

### **Checkpoint Restart**

The Checkpoint Restart feature eliminates the need to retransmit an entire file in the event of a transmission failure. If a copy procedure is interrupted, Connect:Direct restarts that copy at the last checkpoint.

### **Command Line Interface**

The Command Line Interface is a Connect:Direct interface that allows users to submit Connect:Direct Processes and commands from their native command line environment.

### **Commands**

Connect:Direct commands initiate and monitor activity within the Connect:Direct system.

## I

### Integrated File System

The integrated file system provides an integrated structure of all stored information. This file system provides stream files, directory structure accessed by path, and a user and application interface that accesses all information stored in the i5/OS.

## L

### Local Node

The local node is the Connect:Direct server on which the Process is submitted. The local node may also be referred to as the primary node, controlling node or initiating node, but should not necessarily be interpreted as the sending node, since PNODE can be the receiver. In every Process, there is one local (PNODE) and one remote (SNODE) specified. The submitter of a Process is always the PNODE.

### Local Process

A local Process is a Connect:Direct Process initiated on the local Connect:Direct for i5/OS node.

## N

### Native File System

The native file system accepts the file name in the following formats: library-name/file-name and library-name/file-name(member-name).

### Network Map (Netmap)

The Network Map (netmap) is a file that identifies all valid Connect:Direct nodes in the network. One Network Map is associated with each Connect:Direct local node. The netmap has one entry for each of the other Connect:Direct nodes to which the local Connect:Direct node communicates. The netmap entries also contain the rules or protocol that the nodes adhere to when communicating.

### Node

A node is any site in a network from which information distribution can be initiated.

## P

### Permanent Session Managers

Permanent Session Managers are reusable session managers submitted when Connect:Direct is started or when the Connect:Direct administrator increases the number of permanent session managers.

### Primary Node (PNODE)

The local node or primary node (PNODE) is the Connect:Direct node on which the Process is submitted. The primary node may also be referred to as the controlling node or initiating node, but should not necessarily be interpreted as the sending node, since PNODE can be the receiver. In every Process, there is one local (PNODE) and one remote (SNODE) specified. The submitter of a Process is always the PNODE.

## Process

A Process is a series of statements that initiate Connect:Direct activity, such as copying files, and running jobs.

## Process Management Queue

The Process Management Queue (PMQ) holds information about Connect:Direct Processes that are currently executing or scheduled to execute in the future.

## Process Manager (PMGR)

The Process Manager initiates and manages the Connect:Direct environment.

## Process Statements

Process Statements are instructions for transferring files, running operating system jobs, executing programs, or submitting other Connect:Direct Processes. They are used to build a Connect:Direct Process.

# Q

## Queue Manager (QMGR)

The Queue Manager is a fixed Connect:Direct for i5/OS component. The Process Manager starts the Queue manager when it starts Connect:Direct, and the QMGR remains active while Connect:Direct is active. The Queue Manager has five logical queues: ACTIVQ, executing processes; WAITQ, processes waiting to run; HOLDQ, processes currently held; TIMERQ, processes waiting to be moved to the WAITQ; and STARTQ, asynchronous processes being activated.

# R

## Remote Node

A remote node is an entry in the Network Map that defines a Connect:Direct node with which the local Connect:Direct node can communicate. The remote node is also referred to as an adjacent node.

## Remote Process

A remote Process is a Connect:Direct Process initiated on the remote Connect:Direct node.

## Retry Interval

The retry interval is the interval at which retries are performed as a part of the checkpoint-restart feature.

## S

### Secondary Node (SNODE)

The remote or secondary node (SNODE) is the Connect:Direct node that interacts with the primary node (PNODE) during Process execution. SNODE may also be referred to as the participating (non-controlling) or partner node. Every Process has one PNODE and one SNODE.

### Session Manager (SMGR)

The session manager component executes the Connect:Direct Processes. You can set the initialization parameters to make the SMGR either fixed or dynamic.

### Spool Monitor (CDSPLMON)

The Connect:Direct Spool Monitor (CDSPLMON) is a variable component of Connect:Direct for i5/OS. CDSPLMON uses Spool Monitor entries to associate output queues with destination Connect:Direct nodes. These entries also include default values used to transfer spooled files.

### Statistics Facility

The Connect:Direct Statistics Facility records Connect:Direct activities.

### Statistics File

The statistics file holds Connect:Direct statistics records that document the history of a Process.

### Statistics Manager (STATMGR)

Connect:Direct for i5/OS logs statistics information for all Connect:Direct Processes. Each component of Connect:Direct for i5/OS generates statistics data to the statistics manager (STATMGR). The long-running STATMGR formats the statistics information and writes it to the statistics file.

Submitted Session Managers are session managers submitted by the Process Manager.

### Synchronous

Synchronous Process execution occurs when you submit a Connect:Direct command. Processes such as CDSND and CDRCV are submitted by the user and they execute immediately.

## T

### TCP/IP Listener

The TCP/IP Listener component requests session managers from the Process Manager when it receives connection requests. The Process Manager starts and controls the TCPLIST component.

### Timer Queue (TIMER)

This Connect:Direct for i5/OS component provides timer services to the other components. The process manager submits and controls the TIMER. The TIMER is always present when Connect:Direct is active.

## A

ACTIVQ definition 132

Administration Menu

- accessing 11
- authority required 11
- screens 11

asynchronous execution

- CDSUBPROC screens 129
- submitting Process to Process queue 129

asynchronous processes

- definition 125
- submitting 126

## B

BLKSIZE 30, 83

- CDSND command 30
- CDSNDSPL command 83

block mode transfer

- CDRCV example 41, 42, 68
- CDSND and CDRCV example 42, 69
- CDSND example 42, 69
- description 40, 67
- examples 41, 68

## C

case sensitivity 15, 19, 22

- integrated file system 20, 48

CDCOMP command

- accessing 122
- CKPTINV 120
- compressing a file 122
- example 119
- EXCMBR 120
- format 120
- general description 119
- INFILE 120
- OUTFILE 120

- parameters 120
- REPLACE 121
- screens 122
- SELMBR 121

CDDECOMP command

- accessing 146
- decompressing a file 146
- example 145
- format 145
- general description 145
- INFILE 146
- NEWNAME 146
- OUTFILE 146
- parameter 146
- parameters 146
- REPLACE 146
- screens 146
- tuning parameters 145

CDMSGID, DSPCDMSGD command 150

CDRCV command

- accessing screens 70
- CKPTINV 57
- CLASS 57
- COMPRESS 58
- CRC parameter 58
- example 48
- EXCLUDE 59
- FALIAS 59
- FBUFND 59
- FDBPARMS 59
- FDCB 59
- FDISP 61
- FDSN 51
- FIOPARM 61
- FLABEL 61
- FLINK 62
- FMSVGP 62
- FMSYSOPTS subparameters, PRECMR 52
- FMSYSOPTS subparameters, TYPE 53
- FMSYSOPTS subparameters, XTRAN 56

- format 49
- FPDSDIR 62
- FSQL 62
- FTYPE parameters 63
- FUNIT parameters 63
- FVOL 63
- FVOLREF 63
- FVSAMCAT 63
- general description 47
- NOTIFY 63
- parameters 50
- PNAME 64
- PNODEID 64
- receiving files 70
- REPLACE parameters 64
- RMTWORK 64
- SACCT 64
- screens 69
- SELECT 65
- SNDDFD 52
- SNODE 50
- SNODEID 65
- SNODENVIRN 50
- TDISP 66
- TDSN 52
- TEXTFILE 53
- TOSYSOPTS 53
- XTRANLDATA 56
- XTRANLSI 56
- XTRANLSO 56
- CDRUNJOB command 103
  - accessing screens 108
  - CLASS 105
  - CRC parameter 105
  - example 103
  - format 104
  - JOBFILE 105
  - LINK 105
  - NOTIFY 106
  - parameters 104
  - PNODEID 106
  - RMTWORK 106
  - SACCT 106
  - SNODEID 106
  - SNODENVIRN 104
  - submitting job to remote node 108
  - TSYSOPTS 108
  - using the screens 108
- CDRUNTASK command
  - CLASS 98
  - CMD 97
  - CRC parameter 98
  - example 95
  - general description 95
  - initiating remote node task 101
  - NOTIFY 98
  - parameters 97
  - PARM 98
  - PGM 97
  - PNAME 98
  - PNODEID 98
  - RMTWORK 99
  - SACCT 99
  - screens 100
  - SNODEID 99
  - syntax 96
- CDSELSTAT command
  - accessing screens 142
  - example 137
  - FILE 138
  - FORMAT 138
  - format 137
  - general description 137
  - OUTPUT 138
  - parameters 138
  - PERIOD 138
  - PNAME 139
  - PNUM 139
  - PUSER 139
  - RCDTYPE 139
  - RMTNODE 139
  - screens 140
  - selecting statistics 140
- CDSND command 139
  - BLKSIZE 30
  - CKPTINV 30
  - CLASS 30
  - COMPRESS 30
  - CRC parameter 31
  - example 21
  - EXCLUDE 32
  - FDISP 32
  - FDSN 23
  - FMSYSOPTS 24
  - format 21
  - general description 19, 20
  - NOTIFY 32
  - parameters 21, 23
  - PNAME 32
  - PNODEID 33
  - PRECMR 25
  - REPLACE 33

- RMTWORK 33
- SACCT 33
- screens 43
- SECOPTIONS 33, 38, 64, 66, 90, 99, 107, 116, 128
- SELECT 33
- sending a file 43
- SNDFFD 25
- SNODE 23
- SNODEID 34
- TBUFND 34
- TDBPARMS 35
- TDCB 35
- TDISP 36
- TDSN 23
- TEXTFILE 25
- TIOEXIT 37
- TIOPARM 37
- TLABEL 37
- TLINK 37
- TOSYSOPTS 26, 82
- TSPACE 39
- TSQL 39
- TTYPE 39
- TUNIT 40
- TVOL 40
- TVOLREF 40
- TVSAMCAT 40
- XTRAN 25
- XTRANLDATA 25
- XTRANLSI 25
- XTRANLSO 25
- CDSNDSPL command 75, 77
  - BLKSIZE 83
  - CHLVAL 83
  - CKPTINV 83
  - CLASS 83
  - COMPRESS 84
  - CRC parameter 84
  - CTLCHAR 85
  - example 75
  - FDSNLEN 85
  - format 75
  - JOB 77
  - NOTIFY 85
  - parameter 75, 77
  - PNAME 85
  - PNODEID 86
  - PRTY 86
  - RMTWORK 86
  - SACCT 86
  - screens 93
  - SECOPTIONS 86
  - SNODE 77
  - SNODEID 86
  - SNODENVIRN 77
  - SPLF 77
  - SPLFNUM 78
  - TBUFND 87
  - TDBPARMS 87
  - TDCB 87
  - TDISP 89
  - TDSN 78
  - TIOEXIT 89
  - TIOPARM 89
  - TLABEL 89
  - TLINK 90
  - TMSVGP 90
  - TOSYSOPTS 79
  - TSPACE 91
  - TSQL 92
  - TTYPE 92
  - TUNIT 92
  - TVOL 92
  - TVOLREF 92
  - TVSAMCAT 92
  - XTRAN 81
  - XTRANLDATA 82
  - XTRANLSI 82
  - XTRANLSO 82
- CDSUBMIT command
  - accessing 117
  - CLASS 113
  - CRC parameter 113
  - example 111
  - format 111
  - general description 111
  - NEWNAME 113
  - NOTIFY 113
  - parameters 112
  - parameters, PPACCT 114
  - PCLASS 114
  - PHOLD 114
  - PNAME 114
  - PNODEID 114
  - PNOTIFY 114
  - PPNODEID 115
  - PPRTY 115
  - PREQUEUE 115
  - PRETAIN 115
  - PROCFILE 113
  - PSACCT 115

- PSNODE 115
- PSNODEID 115
- PSTARTT 115
- PSYMBOLS 115
- RMTWORK 115
- SACCT 115
- screens 117
- SNODE 113
- SNODEID 116
- SNODENVIRN 113
- submitting Process on remote node 117
- CDSUBPROC command
  - accessing 129
  - asynchronous execution 129
  - CMD 127
  - example 126
  - format 126
  - functions 125
  - general description 126
  - HOLD 127
  - PRIORITY 127
  - RETAIN 127
  - screens 129
  - STARTT 127
  - submitting Process to Process queue 129
- CHLVAL
  - CDSNDSPL command 83
  - parameter 83
- CHLVAR 83
- CKPTINV parameter
  - CDCOMP command 120
  - CDRCV command 57
  - CDSND command 30
  - CDSNDSPL command 83
- CL commands
  - administering with 7
  - Connect:Direct, entering 15
  - Connect:Direct, list of 14
  - executing through Main Menu 9
  - functions 14
  - interfacing with 7
  - issuing 7
  - program sample 17
  - programming 16
  - results of 7
- CL program examples 153
  - accessing 153
  - block mode transfer 153
  - compress 153
  - decompress 153
  - receive - i5/OS from MVS 156
  - receive - i5/OS from UNIX 158
  - receive - i5/OS from Windows 157
  - receive using integrated file system 155
  - receive using native file system 154
  - run job - i5/OS to UNIX 160
  - run job - i5/OS to Windows 158
  - RUN TASK 154
  - run task - i5/OS to UNIX 160
  - run task - i5/OS to Windows 158
  - send - i5/OS to MVS 156
  - send - i5/OS to UNIX 158
  - send - i5/OS to Windows 157
  - send using integrated file system 155
  - send using native file system 154
  - send using send file field description 154
  - viewing 153
- CLASS parameter 30, 57, 83, 98, 105, 113
- CMD parameter 97
  - CDSUBPROC command 127
- command level
  - help 17
  - prompt 18
- commands
  - accessing help 17
  - and CL programming 16
  - CDCOMP 119, 120
  - CDDECOMP 145
  - CDRCV 47, 48, 49
  - CDRUNJOB 103, 104
  - CDRUNTASK 95, 97
  - CDSELSTAT 137, 138
  - CDSND 19, 20, 23
  - CDSNDSPL 75, 77
  - CDSUBMIT 111, 112
  - CDSUBPROC 126
  - CL command prompting 15
  - command level help 17
  - DSPCDMSGD 149, 150
  - embedding in application 15
  - entering 15
  - free format 15
  - parameter level help 18
  - sample 13
  - STRCDMNU 10
  - WRKCDPROC 130, 131
- components 8
  - Process Manager (PMGR) 8
  - Queue Manager (QMGR) 8



Session Manager (SMGR) 8  
 Spool Monitor (CDSPLMON) 8  
 Statistics Manager (STATMGR) 8  
 TCP Listener (TCPLIST) 8  
 Timer (TIMER) 8

**COMPRESS**

CDRCV command 58  
 CDSND command 30  
 CDSNDSPL command 84  
 parameter 30, 58, 84

compressing a file 119, 122

**Connect:Direct CL commands**

accessing help 18  
 and CL programming 16  
 entering 15  
 functions 14  
 list of 14

**Connect:Direct messages**

routing to log file 149

**Connect:Direct Secure+ Option**

CDSELSTAT screen 141  
 CDSND parameters 40  
 WRKCDPROC screen 134

controlling processes 130, 132

**CRC parameter**

CDRCV command 58  
 CDRUNJOB command 105  
 CDRUNTASK command 98  
 CDSND command 31  
 CDSNDSPL command 84  
 CDSUBMIT command 113

**CTLCHAR parameter 85**

CDSNDSPL command 85  
 described 85

**D**

decompressing a file 145, 146

**Digital Signature**

SECOPTION parameter 33, 64, 86

displaying message text 150

**DSPCDMSGD command 149**

accessing 150  
 CDMSGID 150  
 displaying message text 150  
 parameters 150  
 sample message 151

screens 150

**E****encryption value**

SECOPTIONS parameter 33, 64, 86

entering CONNECT:Direct CL commands 15

error messages 149

**examples**

CDCOMP 119  
 CDDECOMP 145  
 CDRCV 48  
 CDRUNTASK 95  
 CDSELSTAT 137  
 CDSND 21  
 CDSNDSPL 75  
 CDSUBMIT 111  
 CDSUBPROC 126  
 CL commands 13  
 commands  
   CDRUNJOB 103  
   DSPCDMSGD 150  
   WRKCDPROC 130

EXCLUDE parameter 59, 32

**EXCMBR**

CDCOMP command 120

**F****FALIAS**

CDRCV command 59  
 parameter 59

**FBUFND parameter 59**

CDRCV command 59

**FDBPARMS parameter 59**

CDRCV command 59

**FDCB parameter 59**

CDRCV command 59

**FDISP parameter 32, 61, 32****FDSN parameter 23, 51, 23****FDSNLEN parameter 85****FILE parameter, CDSELSTAT command 138****File Systems 19, 47**

Integrated File System 19, 47  
 format 20, 48  
 iSeries library file system 20

- lan server file system 20, 48
- library file system 20, 48
- open type file system 48
- root file system 20, 48
- Integrated File System, iSeries library file system 48
- Native File System 19, 47

## files

- compressing 122
- decompressing 146
- FIOEXIT parameter 61
- FIONAME parameter 61
- FIOPARM parameter 61
- FLABEL parameter 61
- FLINK, CDRCV command 62
- FMSVGP parameter 62
- FMSYSOPTS parameter 24, 52, 24
- FORMAT parameter, CDSELSTAT command 138
- FPDSDIR
  - CDRCV command 62
- FSQL parameter 62
- FTYPE parameter 63
- function keys 17
- FUNIT parameter 63
- FVOL parameter 63
- FVOLREF parameter 63
- FVSAMCAT, CDRCV command 63

**H**

## help option

- command level 17
- parameter level 18
- HOLD parameter 127
  - CDSUBPROC command 127
- HOLDQ definition 132

**I**

## INFILE

- CDCOMP command 120
- CDDECOMP command 146

- integrated file system 19
  - case sensitivity 20, 48
  - specifying path name 20

## interface

- CL commands 7
- CONNECT:Direct for OS/400 7

- iSeries library file system 20, 48

**J**

- JOB parameter 77

## JOBFILE parameter

- CDRUNJOB command 105
- defined 105

**L**

- LAN server file system 20, 48

- library file system 20, 48

## LINK parameter

- CDRUNJOB command 105
- defined 105

- LU 6.2, starting SMGRs 8

**M**

- Main Menu, accessing 10

## messages

- displaying text 150
  - DSPCDMSGD command 150
  - sample message description 151
  - sending to a specific user 149
  - types 149

- monitoring processes 130, 132

**N**

- native file system 19

## NEWNAME

- CDDECOMP command 146
- CDSUBMIT command 113

- NOTIFY parameter 63, 85, 32, 98, 106, 113

**O**

## OUTFILE

- CDCOMP command 120
- CDDECOMP command 146

- OUTPUT parameter, CDSELSTAT command 138

## P

### parameters

CDSELSTAT 138  
 CMD, CDSUBPROC command 127  
 CRC 31, 58, 84, 98, 105, 113  
 for CDCOMP 120  
 for CDRUNJOB command 104  
 for CDRUNTASK command 97  
 for CDSELSTAT 137  
 for CDSUBMIT 112  
 for CDSUBPROC 127  
 for DSPCDMSGD 150  
 for WRKCDPROC 131  
 FORMAT, CDSELSTAT command 138  
 HOLD, CDSUBPROC command 127  
 JOBFIL, CDRUNJOB command 105  
 LINK, CDRUNJOB command 105  
 NEWNAME, CDSUBMIT command 113  
 OUTPUT, CDSELSTAT command 138  
 PCLASS, CDSUBMIT command 114  
 PHOLD, CDSUBMIT command 114  
 PNODEID 98, 106, 114  
 PNOTIFY, CDSUBMIT command 114  
 PNUM, WRKCDPROC command 131  
 PPNODEID, CDSUBMIT command 115  
 PPRTY, CDSUBMIT command 115  
 PREQUEUE, CDSUBMIT command 115  
 PRETAIN, CDSUBMIT command 115  
 PRIORITY, CDSUBPROC command 127  
 PROCFILE, CDSUBMIT command 113  
 PSACCT, CDSUBMIT command 114, 115  
 PSNODE, CDSUBMIT command 115  
 PSNODEID, CDSUBMIT command 115  
 PSTARTT, CDSUBMIT command 115  
 PSYMBOLS, CDSUBMIT command 115  
 PUSER, WRKCDPROC command 131  
 QNAME, WRKCDPROC command 132  
 RETAIN, CDSUBPROC command 127  
 RMTNODE, WRKCDPROC command 132  
 RMTWORK, CDRUNJOB command 106  
 RMTWORK, CDRUNTASK command 99  
 RMTWORK, CDSUBMIT command 115  
 SECOPTIONS 90  
 SECOPTIONS, CDSND  
     command 99, 107, 116, 128  
 SNODEID, CDRUNJOB command 106  
 STARTT, CDSUBPROC command 127  
 TDCB 87  
 TLABEL 89  
 TLINK, CDSNDSPL command 90  
 TMSVGP 90

TSQL 92  
 TSYSOPTS  
     CDRUNJOB command 108  
 TTYPE 92  
 TUNIT 92  
 TVOL 92  
 TVOLREF 92  
 TVSAMCAT, CDSNDSPL command 92  
 TYPE, WRKCDPROC command 132

PARM parameter 98

PCLASS parameter

    CDSUBMIT command 114

PERIOD parameter

    CDSELSTAT command 138

PERIOD, CDSELSTAT command 138

PGM parameter 97

PHOLD parameter

    CDSUBMIT command 114

PNAME

    parameter 64, 85, 139, 32, 98, 106, 114, 131

PNAME, CDSELSTAT command 139

PNODEID parameter 64, 86, 33, 98, 106, 114

PNOTIFY parameter, CDSUBMIT command 114

PNUM parameter

    CDSELSTAT command 139

    WRKCDPROC command 131

PPACCT, CDSUBMIT command 114

PPNODEID parameter, CDSUBMIT command 115

PPRTY parameter, CDSUBMIT command 115

PRECMPR parameter, CDSND command 25

PRECMPR, CDSND command 25

PREQUEUE parameter, CDSUBMIT command 115

PRETAIN parameter, CDSUBMIT command 115

PRIORITY parameter, CDSUBPROC command 127

Process Management Queue

    description 125

Process Manager (PMGR) 8

Processes, executing 8

PROCFILE parameter, CDSUBMIT command 113

prompt option

    command level 18

parameter 18  
 PRTY parameter 86  
 PSACCT parameter, CDSUBMIT command 115  
 PSNODE parameter, CDSUBMIT command 115  
 PSNODEID parameter, CDSUBMIT command 115  
 PSTARTT parameter, CDSUBMIT command 115  
 PSYMBOLS, CDSUBMIT command 115  
 PUSER parameter  
   CDSELSTAT command 139  
   WRKCDPROC command 131

## Q

QDLS 20, 48  
 QLANSrv 20, 48  
 QNAME parameter, WRKCDPROC command 132  
 QOpenSys 20, 48  
 QSYS.LIB 20  
 Queue Manager (QMGR) 8  
 queues  
   ACTIVQ 125  
   HOLDQ 125  
   STARTQ 125  
   TIMERQ 125  
   WAITQ 125

## R

RCDTYPE parameter  
   CDSELSTAT command 139  
 RCDTYPE parameter, CDSELSTAT command 139  
 receiving files  
   CDRCV screen 70  
   described 47  
   using the CDRCV command 48  
 REPLACE parameter 33, 64, 121, 146  
   CDDECOMP command 146  
 RETAIN parameter, CDSUBPROC command 127  
 RMTNODE parameter  
   CDSELSTAT command 139  
   WRKCDPROC command 132  
 RMTWORK parameter  
   CDRCV command 64  
   CDRUNJOB command 106

CDRUNTASK command 99  
 CDSND command 33  
 CDSNDSPL command 83, 86  
 CDSUBMIT command 115  
 root file system 20, 48  
 running tasks 95

## S

SACCT parameter 64, 86, 33, 99, 106, 115  
 scheduling processes 130  
 screens  
   Administration Menu 11  
   CDCOMP 122  
   CDDECOMP 146  
   CDRCV 69  
   CDRUNJOB 108  
   CDRUNTASK 100  
   CDSELSTAT 140  
   CDSND 43  
   CDSUBMIT 117  
   CDSUBPROC 129  
   description of 9  
   DSPCDMSGD 151  
   Main Menu 9  
   WRKCDPROC 132  
 SECOPTIONS parameter 139  
   CDSND  
     command 33, 38, 64, 66, 90, 99, 107,  
     116, 128  
   CDSNDSPL command 86  
   digital signature 33, 64, 86  
   encryption value 33, 64, 86  
 SELECT parameter 65, 33  
 selecting statistics 137  
 SELMBR parameter  
   CDCOMP command 121  
 sending files 19, 20, 75  
 Session Manager (SMGR) 8  
 simulating block sizes 41, 68  
 SNDDFFD parameter  
   CDRCV command 52  
   CDSND command 25  
 CDRUNJOB command  
   parameters  
     SNOE 104

CDRUNTASK command  
 parameters  
 SNODE 97

SNODE parameter 50, 77, 23, 97, 104, 113

SNODEID parameter 65, 86  
 CDRUNJOB command 106  
 CDSND command 34

CDRUNTASK command  
 parameters  
 SNODENVIRN 97

CDSND command  
 SNODENVIRN 23

SNODENVIRN  
 parameter 50, 77, 23, 97, 104, 113

SNODENVIRN parameters 50

Source Edit Utility (SEU)  
 editing CL program 16

SPLF parameter 77

SPLFNUM parameter 78

Spool Monitor (CDSPLMON) 8

starting  
 a receive 47  
 a send 19, 20  
 a send of a spool file 75  
 a task 95

STARTQ, definition 132

STARTT parameter  
 CDSUBPROC command 127

statistics  
 logging 8  
 selecting and printing 142  
 selecting, displaying, and printing 140

Statistics Manager (STATMGR) 8

Status messages  
 routing to specific user ID 149  
 specifying recipient 149

STRCDMNU 10

submitting a job on a remote node 103

submitting a Process on remote node 111

synchronous processes, definition 126

## T

TBUFND parameter 87, 34

TCP Listener (TCPLIST) 8

TCP/IP, request processing 8

TDBPARMS parameter 87, 35

TDCB parameter 87, 35

TDISP parameter 66, 89, 36

TDSN parameter 52, 78, 23

TDSN/FDSN format 51  
 MSP 51  
 OS/390 51  
 Stratus VOS 51  
 VM 51  
 VSE 51

TEXTFILE parameter  
 CDRCV command 53  
 CDSND command 25

Timer (TIMER) component 8

TIMERQ  
 definition 132

TIOEXIT parameter 89, 37

TIOPARM parameter 89, 37

TLABEL parameter 89, 37

TLINK parameter  
 CDSND command 37  
 CDSNDSPL command 90

TMSVGP parameter 90, 38

TOSYSOPTS parameter 26, 53, 79, 82, 26, 82

TOUSR parameter, receiving status messages 149

TSPACE parameter 91, 39

TSQL parameter 92, 39

TSYSOPTS parameter, CDRUNJOB command 108

TTYPE parameter 92, 39

TUNIT parameter 92, 40

TVOL parameter 92, 40

TVOLREF parameter 92, 40

TVSAMCAT parameter  
 CDSND command 40  
 CDSNDSPL command 92

TYPE parameter  
 CDSND command 25  
 WRKCDPROC command 132

**U**

UNIX file system 20, 48

## File Systems

Integrated File System

UNIXtype file system 20

**W**

WAITQ definition 132

working with processes 130

selecting specific Processes 130

user authority 130

viewing all Processes 130

WRKCDPROC command 130

change 134

delete 134

display 134

end 135

example 130

format 131

functions 125

hold 134

options 134

display 135

parameters 131

PNUM 131

PUSER 131

QNAME 132

RMTNODE 132

screens 132

suspend 135

TYPE 132

WRKCDPROC command, PNAME 131

WRKCDPROC screens

accessing 132

monitoring and controlling Processes 132

**X**

XTRAN parameter

CDRCV command 56

CDSND command 25

CDSNDSPL command 81

XTRANLDATA parameter

CDRCV command 56

CDSND command 25

CDSNDSPL command 82

XTRANLSI parameter

CDRCV command 56

CDSND command 25

CDSNDSPL command 82

XTRANLSO parameter

CDRCV command 56

CDSND command 25

CDSNDSPL command 82